

THE AICE LIBRARY

The Shortest Path

A History of the Art and Science of Problem-Solving

เส้นทางที่สั้นที่สุด

ประวัติศาสตร์ของศิลปะและศาสตร์แห่งการแก้ปัญหา

Great solvers do not search harder; they change the space until the path is short.

นักแก้ปัญหาที่ยิ่งใหญ่ไม่ค้นหาหนักขึ้น — เขาเปลี่ยนพื้นที่จนเส้นทางสั้นลง

2026-08-02

How to read this book · วิธีอ่านเล่มนี้

Great solvers do not search harder; they change the space until the path is short.

นักแก้ปัญหาที่ยิ่งใหญ่ไม่ค้นหาหนทางขึ้น — เขาเปลี่ยนพื้นที่จนเส้นทางสั้นลง

Every block of prose in this book appears twice in a row: the English, tagged **EN**, then its Thai twin, tagged **TH** and set against a rule in the margin. The two are never more than a paragraph apart, so a reader working in either language can check the other without turning a page. Read one tag and skip the other, or read both — the book is built for either habit.

ทุกย่อหน้าในเล่มนี้ปรากฏสองครั้งติดกัน — ภาษาอังกฤษที่ป้าย **EN** แล้วตามด้วยฉบับภาษาไทยที่ป้าย **TH** ซึ่งมีเส้นกำกับอยู่ริมซ้าย ทั้งสองห่างกันไม่เกินหนึ่งย่อหน้า ผู้อ่านที่ทำงานด้วยภาษาไหนก็ตามจึงเทียบอีกภาษาได้โดยไม่ต้องพลิกหน้า จะอ่านป้ายเดียวแล้วข้ามอีกป้าย หรืออ่านทั้งคู่ก็ได้ เล่มนี้สร้างมาเพื่อทั้งสองแบบ

Twelve chapters. Each closes by naming its **Move** — a transferable act, stated so it can be carried out of the story it came from. Chapter 99 assembles all eleven Moves into a kit, a standard opening, and a thirty-day route.

สิบสองบท แต่ละบทปิดด้วยการตั้งชื่อ **Move** ของมัน — ท่าที่ถอดออกมาใช้ที่อื่นได้ เขียนไว้ในรูปที่ยกออกจากเรื่องเล่าต้นทางได้จริง บทที่ 99 ประกอบทั้งสิบเอ็ดท่าเข้าเป็นชุดเครื่องมือหมากเปิดมาตรฐาน และเส้นทางฝึกสามสิบวัน

The English canon is the source of truth; the Thai is a rendition of it under a fixed contract (source/TH_SPEC.md). Every dated claim carries a checked source, and anecdotes the historical record disputes are flagged as disputed rather than smoothed over.

ต้นฉบับภาษาอังกฤษคือฉบับจริง ส่วนภาษาไทยคือการถ่ายทอดภายใต้สัญญาที่ตรงไว้ (source/TH_SPEC.md) ทุกข้ออ้างที่มีวันที่มีแหล่งที่ตรวจแล้ว และเรื่องเล่าที่หลักฐานทางประวัติศาสตร์ยังเถียงกันอยู่ถูกกำกับไว้ตรงนั้นว่ายังเถียงกันอยู่ ไม่กลบให้เรียบ

Contents • สารบัญ

00	The Knot and the Path ปมและเส้นทาง	5
01	The First Methods วิธีชุดแรก	23
02	The Reframe การเปลี่ยนกรอบ	48
03	The Heuristic Age ยุค heuristic	72
04	War and the Science of Search สงครามกับศาสตร์แห่งการค้นหา	95
05	The Algorithmic Path เส้นทางแบบอัลกอริทึม	119
06	The Wall กำแพง	142
07	Gallery I: The Single Move หอเรื่องเล่า I: ทำเดียวจบ	163
08	Gallery II: The Modern Solves หอเรื่องเล่า II: การแก้ครั้งใหญ่ยุคใหม่	185
09	The Aesthetics of Solution สุนทรียะของคำตอบ	208
10	The Future Solver นักแก้ปัญหาแห่งอนาคต	230
99	The Leverage Route เส้นทางคานงัด	256

CHAPTER 00

The Knot and the Path

ปมและเส้นทาง

Gordium, 333 BC

กอร์เดียม 333 ปีก่อนคริสตกาล

EN

In the old capital of Phrygia, in what is now central Turkey, an ox-cart stood in the citadel with its yoke lashed to a pole by a knot of cornel bark. The bark had hardened over generations until the knot was a single woody mass with no visible ends. An oracle had attached a promise to it: whoever undid the knot would rule Asia.

TH

เกวียนเทียมวัวคันหนึ่งตั้งอยู่ในป้อมปราการของเมืองหลวงเก่าแห่งฟรีเจีย ดินแดนที่ปัจจุบันคือตอนกลางของตุรกี แอ็กของมันเป็นมัดติดกับคันด้วยปมเปลือกไม้คอร์เนล เปลือกไม้แข็งตัวข้ามหลายชั่วอายุคน จนปมกลายเป็นก้อนไม้ก้อนเดียว มองไม่เห็นปลายเชือกสักปลาย คำพยากรณ์ผูกคำสัญญาไว้กับมัน: ใครแก้ปมนี้ออกได้ จะได้ปกครองเอเชีย

EN

Alexander of Macedon arrived that year, twenty-three years old and mid-conquest, and could not afford to walk away from a public prophecy. The ancient sources preserve two versions of what happened next, and the disagreement between them is worth more than either story alone.

TH

Alexander แห่งมาซิโดเนียมาถึงในปีนั้น อายุยี่สิบสามปี อยู่กลางศึกพิชิตดินแดน และเดินหนีคำพยากรณ์ที่คนทั้งเมืองรู้กันไม่ได้ แหล่งข้อมูลโบราณเก็บเรื่องที่เกิดขึ้นต่อจากนั้นไว้สองสำนวน ความไม่ลงรอยระหว่างสองสำนวนนั้นมีค่ามากกว่าเรื่องเล่าอันใดอันหนึ่งเพียงลำพัง

E N

In the version most people know — the one Plutarch says most writers reported — Alexander studied the knot, found no ends to work with, drew his sword, and cut it through (Plutarch, *Life of Alexander* 18). In the second version, which Arrian attributes to Aristobulus — an engineer who was actually on the campaign — Alexander did something quieter: he pulled out the wooden pin that ran through the pole and held the whole assembly together. The yoke slid free. The knot, still tied, no longer fastened anything (Arrian, *Anabasis* 2.3).

T H

Alexander ฟันปมขาดด้วยดาบ — นั่นคือสำนวนที่คนส่วนใหญ่รู้จัก และเป็นสำนวนที่ Plutarch บอกว่านักเขียนส่วนใหญ่รายงานตรงกัน: เขาพิเคราะห์ปม ไม่พบปลาย เชือกให้จับ จึงชักดาบออกมาฟัน (Plutarch, *Life of Alexander* 18) สำนวนที่สองเจียบกว่านั้น Arrian อ้างว่ามาจาก Aristobulus วิศวกรที่ร่วมทัพอยู่จริง: Alexander ดึงสลักไม้ที่เสียบทะลุคานและยึดทั้งชุดไว้ด้วยกันออกมา แยกหลุดเป็นอิสระ ปมยังผูกอยู่อย่างนั้น แต่ไม่ได้ยึดอะไรไว้อีกแล้ว (Arrian, *Anabasis* 2.3)

E N

Notice what neither version contains: Alexander untying the knot. Both accounts agree that he refused the problem as offered. The oracle said *undo the knot*; everyone before him had heard *untie it by hand* and failed. Alexander — by blade or by pin — heard only *make the knot irrelevant*, and succeeded in seconds at a task that had defeated petitioners for generations.

T H

ไม่มีสำนวนไหนเลยที่ Alexander แก่ปมออก ทั้งสองบันทึกตรงกันว่าเขาปฏิเสธปัญหาในรูปที่ถูกยื่นมาให้ คำพยากรณ์บอกว่า ทำให้ปมคลาย ทุกคนก่อนหน้าเขาได้ยื่นเป็น แก้วด้วยมือ แล้วล้มเหลว Alexander — ด้วยคมดาบหรือด้วยสลักไม้ก็ตาม — ได้ยื่นแค่ ทำให้ปมไม่สำคัญ และทำสำเร็จในไม่กี่วินาที กับังงานที่เอาชนะผู้มาลองมาแล้วหลายชั่วอายุคน

E N

The sword version celebrates force. The pin version celebrates insight. But the decisive move in both is the same, and it happened before anyone touched the rope: a decision about **what the problem actually was**. That move — made deliberately, named, studied, taught, and eventually mechanized — is what this book is about.

T H

ทำซ้ำขาดในทั้งสองสำนวนคือทำเดียวกัน และเกิดขึ้นก่อนใครจะแตะเชือก: การตัดสินใจว่า **ปัญหานี้คืออะไรกันแน่** สำนวนดาบยกย่องกำลัง สำนวนสลักไม้ยกย่องการหยั่งเห็น (insight) แต่ทั้งคู่เริ่มจากการตัดสินใจข้อนั้น ทำนั้น — ทำอย่างตั้งใจ ตั้งข้อศึกษา สอน และในที่สุดทำให้เป็นกลไก — คือเรื่องของหนังสือเล่มนี้

What a problem is

ปัญหาคืออะไร

E N

We use the word "problem" loosely, for everything from a flat tire to a failing marriage to Fermat's Last Theorem. The scientific study of solving begins by tightening it. The Gestalt psychologist Karl Duncker, writing in 1945, gave the definition the field still leans on: a problem exists when a living creature has a goal but does not know how the goal is to be reached (Duncker, *On Problem-Solving*, 1945). The key word is *know*. A flat tire is not a problem for a mechanic — it is a task, a known procedure executed with known tools. It becomes a problem the moment the jack is missing and nothing in memory supplies the next step.

T H

ปัญหาเกิดขึ้นเมื่อสิ่งมีชีวิตหนึ่งมีเป้าหมาย แต่ไม่รู้ว่าจะไปถึงเป้าหมายนั้นอย่างไร — นิยามนี้ Karl Duncker นักจิตวิทยาสำนักเกสตัลท์ (Gestalt) เขียนไว้ในปี 1945 และวงการยังพึงอยู่จนทุกวันนี้ (Duncker, *On Problem-Solving*, 1945) ก่อนถึงนิยามนั้น เราใช้คำว่า "ปัญหา" อย่างหลวมๆ ตั้งแต่ยางแบน ชีวิตคู่ที่กำลังพัง ไปจนถึงทฤษฎีบทสุดท้ายของ Fermat การศึกษาการแก้ปัญหาอย่างเป็นวิทยาศาสตร์เริ่มจากรัดค่านี้นี้ให้แน่น คำสำคัญคือ รู้ ยางแบนไม่ใช่ปัญหาสำหรับช่าง มันคืองาน คือขั้นตอนที่รู้อยู่แล้ว

ทำด้วยเครื่องมือที่รู้อยู่แล้ว มันกลายเป็นปัญหาในวินาทีที่แม่แรงหายไป และไม่มีอะไรในความจำบอกขั้นต่อไปได้

E N

A problem, then, is a **gap with no visible bridge**: a present state, a desired state, and no ready-made route between them. This definition does something useful — it makes "problem" a relationship between a situation and a solver, not a property of the situation alone. The same knot is a wall to one mind and a doorway to another. The entire discipline lives in that difference.

T H

ปัญหาคือ ช่องว่างที่ยังมองไม่เห็นสะพาน: สถานะ (state) ปัจจุบัน สถานะที่อยากไปถึง และไม่มีเส้นทางสำเร็จระหว่างสองสถานะนั้น นิยามนี้มีประโยชน์ตรงที่ทำให้ "ปัญหา" เป็นความสัมพันธ์ระหว่างสถานการณ์กับนักแก้ปัญหา ไม่ใช่คุณสมบัติของสถานการณ์เพียงลำพัง ปมเดียวกันเป็นกำแพงสำหรับสมองหนึ่ง เป็นประตูสำหรับอีกสมองหนึ่ง ศาสตร์ทั้งศาสตร์อยู่ในความต่างตรงนั้น

E N

In 1972, after fifteen years of watching people solve puzzles while thinking aloud, Allen Newell and Herbert Simon gave the definition its formal skeleton in *Human Problem Solving*, one of the foundational works of cognitive science (Newell & Simon, 1972). A problem, they proposed, can be described as a **problem space**: a set of *states* (every configuration the situation could be in), a set of *operators* (every legal move that transforms one state into another), an *initial state*, and a *goal state*. Solving is **search**: finding a sequence of operators — a path — from where you are to where you want to be.

T H

ปี 1972 Allen Newell กับ Herbert Simon ให้โครงกระดูกรูปนัยแก่นิยามนั้นใน *Human Problem Solving* หนึ่งในงานรากฐานของวิทยาศาสตร์การรู้คิด (cognitive science) หลังจากเฝ้าดูคนแก้ปริศนาพร้อมพูดความคิดออกมาดังๆ มาสิบห้าปี (Newell & Simon, 1972) ทั้งสองเสนอว่าปัญหาอธิบายได้ในรูป **พื้นที่ปัญหา (problem space)**: เซตของ สถานะ (ทุกรูปแบบที่สถานการณ์เป็นได้) เซตของ ตัว

ดำเนินการ (operator) (ทุกการเคลื่อนที่ถูกกติกาที่เปลี่ยนสถานะหนึ่งเป็นอีกสถานะหนึ่ง) สถานะตั้งต้น และ สถานะเป้าหมาย การแก้ปัญหาคือ **การค้นหา (search)**: หาลำดับของ operator — เส้นทางเส้นหนึ่ง — จากจุดที่ยืนอยู่ไปยังจุดที่อยากไปถึง

E N

This sounds like an analogy. It is not. It is a formalism, and it turned out to be astonishingly general. A chess position, a half-finished proof, a molecule to synthesize, a route through a city, a debugging session, a legal argument — each fits the frame: states, moves, start, goal. When Newell and Simon built the first artificial-intelligence programs on this idea in the late 1950s, the programs worked, which is the hardest kind of evidence that a definition has caught something real. The same skeleton now runs inside route planners, theorem provers, and the game engines that beat world champions. Chapter 05 tells that story in full.

T H

นี่ไม่ใช่การเทียบเคียง มันคือระบบรูปนัย และมันทั่วไปกว่าที่ใครคาด ตำแหน่งบนกระดานหมากรุก บทพิสูจน์ที่ค้างครึ่งทาง โมเลกุลที่ต้องสังเคราะห์ เส้นทางผ่านเมืองหนึ่ง การนั่งไล่บี้ข้อโต้แย้งทางกฎหมาย — ทุกอย่างเข้ากรอบนี้ได้: สถานะ การเคลื่อนที่ จุดเริ่ม จุดหมาย Newell กับ Simon สร้างโปรแกรมปัญญาประดิษฐ์ชุดแรกบนไอดีนี้ในปลายทศวรรษ 1950 และโปรแกรมทำงานได้จริง นั่นคือหลักฐานชนิดที่แข็งที่สุดว่านิยามหนึ่งจับของจริงได้ โครงกระดูกเดียวกันนี้วันนี้ทำงานอยู่ในตัววางแผนเส้นทาง ตัวพิสูจน์ทฤษฎีบท และเอนจินเกมที่เราชนะแชมป์โลก บทที่ 05 เล่าเรื่องนั้นเต็มๆ

E N

The problem-space view also explains, in one stroke, why problems are hard. The spaces are unimaginably large. Chess has more possible games than there are atoms in the observable universe. The space of possible proofs, or possible protein folds, or possible business strategies, is worse. No solver — human or machine — searches these spaces exhaustively. Anyone who solves anything nontrivial is doing something cleverer than looking everywhere.

T H

ปัญหายากเพราะพื้นที่ใหญ่เกินจินตนาการ — มุมมองแบบพื้นที่ที่ปัญหาอธิบายข้อนี้ได้ในที่เดียว หมายความว่าเกมที่เป็นไปได้มากกว่าจำนวนอะตอมในเอกภพที่สังเกตได้ พื้นที่ของบทพิสูจน์ที่เป็นไปได้ รูปการพับโปรตีนที่เป็นไปได้ หรือกลยุทธ์ธุรกิจที่เป็นไปได้ หนักกว่านั้นอีก ไม่นึกแก้ปัญหาหายไหน — คนหรือเครื่อง — ค้นพื้นที่พวกนี้ได้ครบทุกจุด ใครทำอะไรที่ไม่ธรรมดาได้ กำลังทำอะไรที่ฉลาดกว่าการมองให้ทั่วทุกที่

The thesis of this book

ข้อเสนอของหนังสือเล่มนี้

E N

What is the cleverer thing? A century and a half of mathematics, psychology, wartime engineering, and now machine intelligence converges on an answer that can be said in one sentence:

T H

คำถามคือสิ่งที่ฉลาดกว่านั้นคืออะไร คณิตศาสตร์ จิตวิทยา วิศวกรรมยามสงคราม และปัญญาของเครื่องจักรในวันนี้ ใช้เวลารวมกันหนึ่งศตวรรษครึ่งจึงบรรจบที่คำตอบเดียว และคำตอบนั้นพูดได้ในประโยคเดียว:

E N

Great solvers do not search harder; they change the space until the path is short.

T H

นักแก้ปัญหาที่ยิ่งใหญ่ไม่ค้นหาหนักขึ้น — เขาเปลี่ยนพื้นที่จนเส้นทางสั้นลง

E N

Difficulty is not a fixed property of a problem. Difficulty lives in the *representation* — in which states you bother to distinguish, which operators you allow yourself, and what you take the goal to be. Change any of those, and a maze can collapse into a corridor.

T H

ความยากไม่ใช่คุณสมบัติตายตัวของปัญหา มันอยู่ใน รูปแบบการแทนปัญหา (representation) — อยู่ที่ว่าเลือกแยกแยะสถานะไหน อนุญาตให้ตัวเองใช้ operator ไหน และถือว่าเป้าหมายคืออะไร เปลี่ยนอย่างใดอย่างหนึ่ง在那นั่น เขาจะกต ก็ยุบเหลือทางตรงเส้นเดียว

E N

The book in your hands is a history of that collapse, performed again and again by the best minds of every era:

T H

หนังสือเล่มนี้คือประวัติศาสตร์ของการยุบตัวนั้น หัวคิดดีที่สุดของทุกยุคทำมันซ้ำแล้วซ้ำอีก:

E N

- Leonhard Euler, asked whether a stroller could cross each of Königsberg's seven bridges exactly once, threw away the city — the streets, the water, the geography — until only dots and lines remained, and in that stripped space the answer became a two-line argument. Graph theory was the residue of his reframing (chapter 02).
- John Snow, facing a cholera epidemic that the best minds of 1854 attributed to bad air, changed what a *state* was: not a patient, but an address. Plotted on a map, the deaths encircled a single water pump on Broad Street (chapter 07).
- The physicists and economists of the 1940s discovered that thousands of apparently different questions — feeding an army cheaply, routing convoys, scheduling factories — were one question wearing different clothes, and built a single mathematical engine to answer all of them (chapter 04).
- A team at DeepMind, facing the fifty-year-old problem of predicting how a protein folds, stopped trying to simulate the physics and recast folding

as a pattern-completion problem over evolutionary history — and effectively closed the problem in a single year (chapter 08).

T H

- Leonhard Euler ถูกถามว่าคนเดินเล่นจะข้ามสะพานทั้งเจ็ดของ Königsberg ให้ครบสะพานละครั้งพอดีได้ไหม เขาโยนเมืองทิ้ง — ถนน สายน้ำ ภูมิศาสตร์ — จนเหลือแต่จุดกับเส้น และในพื้นที่ที่ถูกถลกจนเปลี่ยนนั้น คำตอบกลายเป็นข้อโต้แย้งสองบรรทัด ทฤษฎีกราฟคือตะกอนที่เหลืจากการเปลี่ยนกรอบ (reframe) ของเขา (บทที่ 02)
- John Snow เฝ้าดูการระบาดของอหิวาตกโรคที่หวัคคิดดีที่สุดของปี 1854 โยนความผิดให้อากาศเสีย เขาเปลี่ยนว่าอะไรคือ สถานะ: ไม่ใช่ผู้ป่วยหนึ่งคน แต่คือที่อยู่หนึ่งแห่ง พอลงจุดบนแผนที่ ผู้เสียชีวิตล้อมรอบเครื่องสูบน้ำเพียงตัวเดียวบนถนน Broad Street (บทที่ 07)
- นักฟิสิกส์และนักเศรษฐศาสตร์ในทศวรรษ 1940 ค้นพบว่าคำถามนับพันที่ดูต่างกัน — เลี้ยงกองทัพให้ถูกที่สุด จัดเส้นทางกองเรือคุ้มกัน จัดตารางโรงงาน — คือคำถามเดียวที่เปลี่ยนเสื้อผ้า แล้วสร้างเอนจินคณิตศาสตร์ตัวเดียวขึ้นมาตอบมันทั้งหมด (บทที่ 04)
- ทีมหนึ่งที่ DeepMind เฝ้าดูปัญหาอายุห้าสิบปีเรื่องการทำนายว่าโปรตีนพับตัวอย่างไร เลิกพยายามจำลองฟิสิกส์ แล้วหล่อการพับโปรตีนใหม่ให้เป็นปัญหาการเติมรูปแบบให้ครบบนประวัติศาสตร์วิวัฒนาการ — และปิดปัญหานั้นลงได้จริงภายในปีเดียว (บทที่ 08)

E N

Different centuries, different fields, one silhouette: the space changes, then the walk is short.

T H

ต่างศตวรรษ ต่างสาขา เกร่างเดียวกัน: พื้นที่เปลี่ยน แล้วทางเดินก็สั้น

E N

There is a hard version of this claim and a soft version. The soft version says reframing is *one useful trick among many*. The hard version — the one this book defends — says reframing is *the* central act, and that most of what we call genius in solving is skill at choosing spaces, with the searching itself almost an afterthought. The evidence for the hard version fills the chapters ahead; the honest limits of it (some spaces resist all compression — that is chapter 06, on the mathematics of genuine hardness) are part of the story too.

T H

หนังสือเล่มนี้ปกป้องเวอร์ชันแข็งของข้ออ้างนี้ ไม่ใช่เวอร์ชันอ่อน เวอร์ชันอ่อนบอกว่า การเปลี่ยนกรอบเป็น ลูกเล่นที่มีประโยชน์อันหนึ่งในหลายอัน เวอร์ชันแข็งบอกว่า การเปลี่ยนกรอบคือการกระทำหลัก อันเดียว และสิ่งที่เราเรียกว่าอัจฉริยภาพในการแก้ปัญหาส่วนใหญ่คือทักษะเลือกพื้นที่ ตัวการค้นหาเองแทบเป็นเรื่องรอง หลักฐานของเวอร์ชันแข็งอยู่ในบทข้างหน้า ชัดจำกัดของมันก็อยู่ในเล่มด้วย และต้องพูดกันตรงๆ: บางพื้นที่ด้านการบีบอัด (compression) ทุกรูปแบบ นั่นคือบทที่ 06 ว่าด้วยคณิตศาสตร์ของความยากที่แท้จริง

Why "shortest"

ทำไมต้อง "สั้นที่สุด"

E N

The title needs defending. Why prize the *shortest* path? Isn't any path to the goal a solution?

T H

ชื่อหนังสือต้องการการปกป้อง เพราะคำถามมีเหตุ: ทำไมต้องให้ค่ากับเส้นทาง ที่สั้นที่สุด ในเมื่อเส้นทางไหนที่ไปถึงเป้าหมายก็เป็นคำตอบ

E N

Three reasons, in ascending order of depth.

T H

เหตุผลมีสามข้อ เรียงจากตื้นไปลึก

E N

First, economy is survival. Every real solver — a person, a company, an algorithm on a deadline — pays for search in time, money, or attention. The history of operations research (chapter 04) is the history of shaving paths because the shavings were worth fortunes and, in wartime, lives.

T H

หนึ่ง ความประหยัดคือการอยู่รอด นักแก้ปัญหาจริงทุกราย — คน บริษัท อัลกอริทึม ที่มีเส้นตาย — จ่ายค่าการค้นหาค้นหาเป็นเวลา เงิน หรือความสนใจ ประวัติศาสตร์ของวิจัยดำเนินงาน (operational research, OR) ในบทที่ 04 คือประวัติศาสตร์ของการเหินเส้นทางให้สั้นลง เศษที่เหินออกมาเป็นเงินก้อนใหญ่ และในยามสงคราม เป็นชีวิตคน

E N

Second, the short path is a *certificate of understanding*. A thousand-step solution can be stumbled into; a three-step solution to the same problem means the solver saw structure the stumbler missed. When Gauss, as a schoolboy, summed the numbers 1 to 100 in seconds by pairing them into fifty 101s, the answer was worth nothing — any drudge could reach it — but the *route* proved he had seen what addition actually is. Teachers, examiners, and referees have always graded the path, not the destination, and they are right to.

T H

สอง เส้นทางสั้นคือ ใบรับรองว่าเข้าใจจริง คำตอบพ้นขั้นตอนอาจได้มาด้วยการคลำสะดุดไปเรื่อยๆ คำตอบสามขั้นตอนของปัญหาเดียวกันแปลว่านักแก้ปัญหาเห็นโครงสร้างที่คนคลำมองไม่เห็น Gauss ตอนเป็นเด็กนักเรียนบวกเลข 1 ถึง 100 ได้ในไม่กี่วินาที ด้วยการจับคู่เป็นเลข 101 ห้าสิบคู่ ตัวคำตอบไม่มีค่าอะไรเลย — คนขยันคนไหนก็ไปถึงได้ — แต่ เส้นทาง พิสูจน์ว่าเขาเห็นแล้วว่าการบวกคืออะไรกันแน่ ครูกรรมการสอบ และผู้ตรวจบทความให้คะแนนเส้นทางเสมอ ไม่ใช่ปลายทาง และเขาถูกแล้ว

E N

Third — and this is the strangest fact the book has to offer — the short path is how truth *feels*. Mathematicians speak of elegant proofs with the vocabulary of aesthetics, and Paul Erdős spoke of "The Book" in which God keeps the perfect proof of each theorem. This is not decoration. Chapter 09 argues, with support from information theory and from the neuroscience of insight, that the experience of beauty in a solution is the felt signal of *compression* — of a representation snapping into a form where the answer is nearly free. The shortest path is not just cheap. It is what understanding is, seen from inside.

T H

สาม — และนี่คือข้อเท็จจริงที่แปลกที่สุดที่หนังสือเล่มนี้มีให้ — เส้นทางสั้นคือ ความรู้สึก ของความจริง นักคณิตศาสตร์พูดถึงบทพิสูจน์ที่สง่างามด้วยคำศัพท์ของสุนทรียศาสตร์ Paul Erdős พูดถึง "คัมภีร์ (The Book)" ที่พระเจ้าเก็บบทพิสูจน์อันสมบูรณ์แบบของทุกทฤษฎีบทไว้ นี่ไม่ใช่เครื่องประดับ บทที่ 09 เสนอว่าประสบการณ์ของความงามในคำตอบหนึ่งคือสัญญาณที่รู้สึกได้ของ การบีบอัด — ของ representation ที่ติดเข้ารูปจนคำตอบแทบได้มาฟรี — โดยมีทฤษฎีสารสนเทศและประสาทวิทยาศาสตร์ของ insight หนุนหลัง เส้นทางที่สั้นที่สุดไม่ได้แค่ถูก มันคือตัวความเข้าใจ เอง เมื่อมองจากข้างใน

E N

One warning, before the warm feeling settles: the shortest path is almost never the first one found. It is usually cut *after* someone has hacked through the long way — Euler's two-liner came after the strollers of Königsberg had worn out their shoes. Elegance is typically a second draft. That, too, is a law of this science, and the galleries of chapters 07 and 08 will show it operating in every century.

T H

เส้นทางที่สั้นที่สุดแทบไม่เคยเป็นเส้นทางแรกที่เราเจอ — คำเตือนข้อนี้มาก่อนความรู้สึกอื่นๆ จะลงตัว ปกติมันถูกเขียนขึ้นมา หลังจาก มีคนบุกฝ่าทางยาวไปแล้ว ข้อโต้แย้งสองบรรทัดของ Euler มาหลังจากคนเดินเล่นแห่ง Königsberg เดินจนรองเท้าสึก ความสง่างามมักเป็นร่างที่สอง นั่นเป็นกฎข้อหนึ่งของศาสตร์นี้ด้วย และหอเรื่องเล่าในบทที่ 07 กับ 08 จะแสดงให้เห็นว่ามันทำงานอยู่ในทุกศตวรรษ

Four traditions, one science

สี่สาย หนึ่งศาสตร์

EN

Is there really a *science* here — or just a shelf of anecdotes about clever people? The skeptic's question deserves a straight answer, and the answer is the structure of this book.

TH

คนที่สงสัยถามว่าตรงนี้มี ศาสตร์ อยู่จริงหรือ หรือมีแค่ชั้นหนังสือที่เต็มไปด้วยเกร็ด
เรื่องคนฉลาด คำถามนี้ควรได้คำตอบตรงๆ และคำตอบคือโครงสร้างของหนังสือเล่ม
นี้

EN

The study of solving grew up four times, in four communities that barely read each other:

TH

การศึกษาการแก้ปัญหาเติบโตขึ้นสี่ครั้ง ในสี่ชุมชนที่แทบไม่อ่านงานของกันและกัน
เลย:

EN

1. **The mathematicians' tradition** asked: can the making of discoveries be taught? It runs from the compiled methods of Egyptian and Greek antiquity, through Descartes' dream of a universal method, to George Pólya's *How to Solve It* — the 1945 handbook that gave heuristics their modern names and became, decades later, required reading for the builders of artificial intelligence. Chapters 01–03. 2. **The psychologists' tradition** asked: what actually happens in a head when the answer arrives? From the Gestalt school's experiments on insight, through Newell and Simon's problem spaces, to the modern neuroscience that can watch the "aha" moment on an EEG trace. Chapters 03 and 09. 3. **The engineers' tradition** asked: can solving be made *industrial* — systematic, optimal, provable? Born in the operational research cells of the Second World War, it produced linear programming, dynamic programming, the shortest-path algorithms that run inside every navigation app, and, at its philosophical peak, the theory of computational complexity: a mathematics of *hardness* itself, which proved

that some spaces cannot be compressed by anyone, ever. Chapters 04–06. 4. **The machines' tradition** is the youngest: solvers that are not us. It began the moment Newell and Simon's programs first worked, and it has now produced systems that find paths no human has ever walked — move 37, novel proofs, folded proteins. What those systems still cannot do, and what that means for anyone who solves for a living, is the business of chapter 10.

TH

1. **สายนักคณิตศาสตร์** ถามว่า สอนให้คนค้นพบสิ่งใหม่ได้ไหม สายนี้ทอดจากวิธีที่ถูกรวบรวมไว้ในอียิปต์และกรีกโบราณ ผ่านความฝันของ Descartes เรื่องวิธีสากล มาจนถึง How to Solve It ของ George Pólya — คู่มือปี 1945 ที่ตั้งชื่อสมัยใหม่ให้วิธีเดาอย่างมีหลัก (heuristic) และกลายเป็นหนังสือที่ผู้สร้างปัญญาประดิษฐ์ต้องอ่านในอีกหลายทศวรรษต่อมา บทที่ 01–03 2. **สายนักจิตวิทยา** ถามว่า ในหัวคนเกิดอะไรขึ้นจริงๆ ตอนคำตอบมาถึง จากการทดลองเรื่อง insight ของสำนัก Gestalt ผ่านพื้นที่ปัญหาของ Newell กับ Simon มาถึงประสาทวิทยาศาสตร์สมัยใหม่ที่เฝ้าดูจังหวะ “อ้อ” บนกราฟ EEG ได้ บทที่ 03 และ 09 3. **สายวิศวกร** ถามว่า ทำให้การแก้ปัญหาเป็น อุตสาหกรรม ได้ไหม — เป็นระบบ เหมาะที่สุด พิสูจน์ได้ สายนี้เกิดในหน่วยวิจัยดำเนินงานของสงครามโลกครั้งที่สอง มันให้กำเนิดกำหนดการเชิงเส้น (linear programming) กำหนดการพลวัต (dynamic programming) อัลกอริทึมหาเส้นทางสั้นที่สุดที่ทำงานอยู่ในแอปนำทางทุกตัว และในจุดสูงสุดเชิงปรัชญาของมันคือทฤษฎีความซับซ้อนเชิงคำนวณ: คณิตศาสตร์ของ ตัวความยากเอง ซึ่งพิสูจน์ว่าบางพื้นที่ไม่มีใครบีบอัดได้ ไม่ว่าเมื่อไหร่ก็ตาม บทที่ 04–06 4. **สายเครื่องจักร** อายุ น้อยที่สุด: นักแก้ปัญหาที่ไม่ใช่เรา มันเริ่มในวินาทีที่โปรแกรมของ Newell กับ Simon ทำงานได้ครั้งแรก และตอนนี้มันผลิตระบบที่หาเส้นทางซึ่งไม่เคยมีมนุษย์คนไหนเดินมาก่อนได้ — Move 37 บทพิสูจน์ใหม่ โปรตีนที่พับแล้ว สิ่งที่ระบบเหล่านั้นยังทำไม่ได้ และความหมายของมันต่อคนที่แก้ปัญหาเป็นอาชีพ คือฐานะของบทที่ 10

E N

Four traditions, four vocabularies, one subject. The claim that they form a single science rests on how well their central ideas translate into one another — heuristic search, insight as restructuring, optimization, representation change — and the book is, among other things, a running demonstration of that translation.

T H

สี่สาย สี่ชุดคำศัพท์ หนึ่งหัวข้อ ข้ออ้างว่าทั้งสี่รวมเป็นศาสตร์เดียวกันอยู่กับว่าไฉน
หลักของแต่ละสายแปลข้ามกันได้ดีแค่ไหน — การค้นหาแบบ heuristic การยังเห็น
ในฐานะการจัดโครงสร้างใหม่ การหาค่าเหมาะสมที่สุด (optimisation) การเปลี่ยน
representation — และหนังสือเล่มนี้ ในบรรดาสิ่งอื่นที่มันเป็น คือการสาธิตการแปล
นั้นไปตลอดทั้งเล่ม

How to read this book

วิธีอ่านหนังสือเล่มนี้

E N

The chapters are arranged in four movements, and a coda:

T H

บททั้งหมดจัดเป็นสี่ท่อน กับท่อนปิดอีกหนึ่ง:

E N

- **I · Origins (01–03):** how humanity noticed that method itself could be an object — made, named, improved, and taught. From the scribes of the Rhind papyrus to Pólya's heuristics.
- **II · The Formal Sciences (04–06):** how solving became engineering — optimization, algorithms, and the wall of provable hardness where engineering ends.
- **III · The Gallery (07–08):** ten great solves across four centuries, each told around the single move that cracked it. The theory made flesh.
- **IV · Beauty and the Future (09–10):** why the short path feels the way it does, and what happens to human solvers now that machines search better than we do.

- **Coda (99):** the whole book folded into a practice — a vocabulary of moves and a thirty-day route for making them reflexes.

T H

- **I · จุดกำเนิด (01–03):** มนุษย์สังเกตได้อย่างไรว่าตัววิธีเองเป็นวัตถุชิ้นหนึ่งได้ — สร้างได้ ตั้งชื่อได้ ปรับปรุงได้ และสอนได้ จากอาลักษณ์ของกระดาศปาปรัส Rhind ถึง heuristic ของ Pólya
- **II · ศาสตร์รูปนัย (04–06):** การแก้ปัญหากลายเป็นวิศวกรรมได้อย่างไร — การหาค่าเหมาะที่สุด อัลกอริทึม และกำแพงของความยากที่พิสูจนได้ ซึ่งเป็นจุดที่วิศวกรรมสิ้นสุดลง
- **III · หอเรื่องเล่า (07–08):** การแก้ครั้งใหญ่สิบครั้งข้ามสี่ศตวรรษ แต่ละเรื่องเล่ารอบทำเดียวที่งัดมันแตก ทฤษฎีที่กลายเป็นเนื้อหนัง
- **IV · ความงามกับอนาคต (09–10):** ทำไมเส้นทางสั้นถึงให้ความรู้สึกแบบที่มันให้ และอะไรจะเกิดขึ้นกับนักแก้ปัญหาที่เป็นมนุษย์ เมื่อเครื่องจักรค้นหาเก่งกว่าเราแล้ว
- **ท่อนปิด (99):** หนังสือทั้งเล่มพับลงเป็นการฝึก — ชุดคำศัพท์ของท่า และเส้นทางฝึกสามสิบวันที่ทำให้ท่าเหล่านั้นกลายเป็นปฏิกิริยาอัตโนมัติ

E N

One structural device runs through everything, and you should know it now. **Every chapter ends by naming its Move** — the single transferable act that its history teaches, stated in a form you can actually use on Monday morning. The moves accumulate. By the coda you will hold the complete kit, and the closing chapter exists to turn the kit from vocabulary into reflex. Read for the stories if you like; the moves are the cargo.

T H

กลไกเชิงโครงสร้างหนึ่งอย่างวิ่งผ่านทั้งเล่ม และควรรู้ไว้ตั้งแต่ตอนนี้ **ทุกบทจบด้วยการตั้งชื่อท่าประจำบท** — การกระทำเดี่ยวที่ถ่ายทอดข้ามงานได้ ที่ประวัติศาสตร์ในบทนั้นสอนไว้ เขียนในรูปที่หยิบไปใช้เช้าวันจันทร์ได้จริง ทำสะสมกันไปเรื่อยๆ พอถึงท่อนปิด ในมือจะมีชุดเครื่องมือครบชุด และบทปิดมีอยู่เพื่อเปลี่ยนชุดเครื่องมือนั้น จากคำศัพท์ให้เป็นปฏิกิริยาอัตโนมัติ จะอ่านเอาเรื่องเล่าก็ได้ แต่ของที่บรรทุกมาจริงๆ คือท่า

EN

A note on evidence, because this library has standards: every dated claim, quotation, and case in this book carries a source, checked at the time of writing. Where the historical record is genuinely uncertain — as with the sword and the pin at Gordium — the uncertainty is stated rather than smoothed over. A history of clear thinking should not itself cheat.

TH

ทุกข้ออ้างที่มีวันที่ ทุกคำอ้างอิง และทุกกรณีในหนังสือเล่มนี้มีแหล่งที่มากำกับ ตรวจแล้ว ณ เวลาที่เขียน — หมายเหตุเรื่องหลักฐานข้อนี้จำเป็น เพราะห้องสมุดนี้มีมาตรฐาน ตรงไหนที่บันทึกทางประวัติศาสตร์ไม่แน่นอนจริง อย่างเรื่องดาบกับสลักไม้ที่กอร์เดียม ความไม่แน่นอนนั้นถูกพูดออกมาตรงๆ ไม่ใช่กลบให้เรียบ ประวัติศาสตร์ของการคิดให้กระจ่างไม่ควรโกงเสียเอง

The Move

ทำประจำบท

EN

Every chapter of this book closes the same way. Here is the first entry in the kit.

TH

นี่คือรายการแรกในชุดเครื่องมือ ทุกบทของหนังสือเล่มนี้ปิดท้ายแบบเดียวกัน

EN

Move 00 — Question the frame before touching the rope.

TH

Move 00 — ตั้งคำถามกับกรอบ ก่อนแตะเชือก

E N

When a problem arrives, it arrives *pre-framed*: someone — a customer, a boss, a textbook, your own habit — has already decided what the states are, what moves are legal, and what counts as solved. The frame reaches you disguised as reality. Generations failed at Gordium because they accepted a rule no oracle had actually spoken: *by hand, ends found, rope intact*.

T H

ปัญหามาถึงพร้อม กรอบที่ถูกตั้งไว้ล่วงหน้า เสมอ: ใครบางคน — ลูกค้า เจ้านาย ตำราเรียน หรือความเคยชินของตัวเอง — ตัดสินไปแล้วว่าสถานะมีอะไรบ้าง การเคลื่อนที่แบบไหนถูกกติกา และอะไรนับว่าแก้ได้แล้ว กรอบมาถึงในคราบของความจริง คนหลายชั่วอายุล้มเหลวที่กอร์เดียมเพราะยอมรับกฎที่ไม่มีคำพยากรณ์ข้อไหน พุดได้เลย: ใช้มือ หาดลายให้เจอ เชือกต้องไม่ขาด

E N

So, before searching — before any cleverness at all — interrogate the frame:

T H

ก่อนจะค้นหา — ก่อนจะฉลาดอะไรทั้งสิ้น — ให้ซักกรอบก่อน:

E N

- What would *count* as solved? Say it precisely. The oracle graded outcomes ("the knot is undone"), not procedures.
- Which constraints are real, and which are habit wearing a uniform? For each rule you are obeying, ask: who set this, and does it bind me?
- Am I solving the problem, or the problem's *packaging*?

T H

- อะไรจึง นับว่า แก้ได้แล้ว พุดออกมาให้ชัด คำพยากรณ์ให้คะแนนที่ผลลัพธ์ ("ปมคลายแล้ว") ไม่ใช่ที่ขั้นตอน
- ข้อจำกัดข้อไหนจริง และข้อไหนคือความเคยชินที่ใส่เครื่องแบบมา สำหรับกฎทุกข้อที่กำลังทำตามอยู่ ให้ถามว่า ใครตั้งกฎนี้ และมันผูกมัดเราจริงหรือ
- กำลังแก้ปัญหา หรือกำลังแก้ กล่องห่อ ของปัญหา

E N

Alexander's blade and Aristobulus' pin are the same move in different costumes: both attack the frame, not the knot. Everything else in this book — Euler's dots, Snow's map, Bellman's recursions, move 37 — is this move, deepened, varied, formalized, and finally taught to machines.

T H

คมดาบของ Alexander กับสลักไม้ของ Aristobulus คือทำเดียวกันในเครื่องแต่งกายคนละชุด: ทั้งคู่โจมตีกรอบ ไม่ใช่ปม ทุกอย่างที่เหลือในหนังสือเล่มนี้ — จุดของ Euler แผนที่ของ Snow การเรียกซ้ำของ Bellman, Move 37 — คือทำนี้ ทำให้ลึกลับขึ้น แดกแขนงออกไป ทำให้เป็นรูปนัย และในที่สุดสอนให้เครื่องจักร

E N

The knot is the problem as handed to you. The path is yours to choose.

T H

ปมคือปัญหาในรูปที่ถูกยื่นมาให้ ส่วนเส้นทาง เลือกเอง

E N

Next — Chapter 01 · The First Methods: four thousand years ago, someone first wrote a solution down not to record the answer, but to teach the route. Method became an object. Everything else followed.

T H

ต่อไป — บทที่ 01 · วิธีชุดแรก: สี่พันปีก่อน มีคนเขียนทางแกลงไปเป็นครั้งแรก ไม่ใช่เพื่อบันทึกคำตอบ แต่เพื่อสอนเส้นทาง วิธีกลายเป็นวัตถุ ทุกอย่างที่เหลือตามมาจากตรงนั้น

CHAPTER 01

The First Methods

วิธีชุดแรก

A Scribe's Desk, Around 1650 BC

โต๊ะอาลักษณ์ ราว 1650 ปีก่อนคริสตกาล

EN

A man named Ahmes sits with a papyrus scroll nearly five meters long unrolled across his lap, copying out mathematics that was already two centuries old when he began. He tells us so himself, in the opening lines of the document: he is writing in the thirty-third regnal year of the Hyksos king Apophis, transcribing a text that goes back to the reign of Amenemhat III of the twelfth dynasty (British Museum; Wikipedia). The scroll he is producing will end up, three and a half thousand years later, in the hands of a Scottish lawyer named Alexander Henry Rhind, who buys it in Thebes in 1858; today the two largest fragments sit in the British Museum, cut apart at some point by antiquities dealers hoping to sell the halves separately (British Museum). We know it as the Rhind Mathematical Papyrus. Ahmes calls it, in his own opening words, something closer to "accurate reckoning for inquiring into things, and the knowledge of all things, mysteries... all secrets."

TH

ม้วนปาปิรุสที่เรารู้จักในชื่อ Rhind Mathematical Papyrus เกิดจากมือของอาลักษณ์ชื่อ Ahmes ผู้นั่งคลี่ม้วนยาวเกือบห้าเมตรออกบนตัก แล้วคัดลอกคณิตศาสตร์ที่เก่าแก่มากสองศตวรรษแล้วในวันที่เขาเริ่มลงมือ เขาบอกเราเองในบรรทัดเปิดของเอกสาร: เขาเขียนในปีที่สามสิบสามแห่งรัชกาลกษัตริย์ Hyksos พระนาม Apophis คัดจากตัวบทที่ย้อนไปถึงรัชสมัยของ Amenemhat III แห่งราชวงศ์ที่สิบสอง (British Museum; Wikipedia) อีกสามพันห้าร้อยปีต่อมา ม้วนนี้ตกไปอยู่ในมือทนายความชาวสก๊อตชื่อ Alexander Henry Rhind ผู้ซื้อมันที่ Thebes ในปี 1858 ทุกวันนี้ชิ้นส่วนสองชิ้นที่ใหญ่ที่สุดอยู่ใน British Museum พ่อค้าของเก่าตัดมันเป็นสองชิ้นใน

ช่วงใดช่วงหนึ่ง หวังขายแยกกัน (British Museum) Ahmes เองเรียกมันด้วยถ้อยคำเปิดของเขา ว่าเป็นอะไรที่ใกล้เคียงกับ "การคำนวณอันแม่นยำเพื่อสืบค้นสรรพสิ่งและความรู้ในสรรพสิ่งทั้งปวง ในความลึกลับ... ในความลับทั้งหมด"

EN

That is a large claim for a document that spends most of its length on bread, beer, and grain. The papyrus contains 84 problems, worked out one after another: how to divide loaves among men so each gets a fair share, how to convert one measure of grain into another, how to find the slope of a pyramid face given its height and base (British Museum). None of this looks like the mathematics we associate with "theorem." There is no proof in it, in the sense Euclid will later demand. What there is, problem after problem, is something almost more valuable to this book's history: a written, repeatable *procedure*, worked start to finish, that a second scribe — one who never met Ahmes and never will meet the original problem-setter — can pick up and execute correctly.

TH

ขนมปัง เบียร์ และเมล็ดข้าว กินความยาวส่วนใหญ่ของเอกสารที่อ้างตัวใหญ่ขนาดนั้น ม้วนนี้มีโจทย์ 84 ข้อ แก่ให้ดูทีละข้อ: แบ่งขนมปังให้คนกลุ่มหนึ่งอย่างไรให้ทุกคนได้ส่วนที่เป็นธรรม แปลงหน่วยตวงเมล็ดข้าวจากหน่วยหนึ่งเป็นอีกหน่วยอย่างไร หาความชันของหน้าพีระมิดจากความสูงและฐานอย่างไร (British Museum) ไม่มีข้อใดดูเหมือนคณิตศาสตร์ที่เราผูกไว้กับคำว่า "ทฤษฎีบท" ไม่มีบทพิสูจน์ในความหมายที่ Euclid จะเรียกร้องในภายหลัง สิ่งที่มี ข้อแล้วข้อเล่า คือของที่มีค่าต่อประวัติศาสตร์ ในหนังสือเล่มนี้แทบยิ่งกว่า: ขั้นตอนวิธี ที่เขียนไว้ ทำซ้ำได้ เดินให้ดูตั้งแต่ต้นจนจบ อาลักษณ์คนที่สอง — คนที่ไม่เคยพบ Ahmes และจะไม่มีวันพบผู้ตั้งโจทย์คนแรก — หยิบขึ้นมาทำตามได้ถูกต้อง

EN

Look at Problem 24. A quantity, added to a seventh of itself, makes 19. Find the quantity. A modern student reaches for algebra: $x + x/7 = 19$, so $8x/7 = 19$, so $x = 133/8$. Ahmes has no algebra — no symbol for the unknown, no equation to rearrange. Instead the papyrus does something that looks, at first, like cheating: it *guesses*. Try 7, since a seventh of 7 is a clean whole

number. Seven plus a seventh of seven is eight. But the answer needs to be 19, not 8 — so the guess was off by a factor of $19/8$. Scale the guess by that same factor — 7 times $19/8$ — and the guess becomes correct: the true answer, expressed the way Egyptian scribes wrote fractions, comes out to 16 plus a half plus an eighth. This is the method later called *false position* — *regula falsi* in the Latin that would revive it more than two thousand years afterward — and its logic depends on nothing more than the fact that the equation is linear: whatever the initial guess produces, the final answer is off by the same *proportion* everywhere, so a single corrective multiplication fixes it in one stroke (*Regula falsi*, Wikipedia; MacTutor, Egyptian papyri).

TH

โจทย์ข้อ 24 แก่ด้วยการเดา จำนวนหนึ่ง บวกหนึ่งในเจ็ดของตัวเอง ได้ 19 จงหาจำนวนนั้น นักเรียนสมัยใหม่คว่าพีชคณิต: $x + x/7 = 19$ ฉะนั้น $8x/7 = 19$ ฉะนั้น $x = 133/8$ Ahmes ไม่มีพีชคณิต — ไม่มีสัญลักษณ์แทนค่าที่ไม่รู้ ไม่มีสมการให้ย้ายข้าง กระดาษปาปิรุสจึงทำสิ่งที่แรกเห็นเหมือนโง่: มันเดา ลอง 7 ดู เพราะหนึ่งในเจ็ดของ 7 เป็นจำนวนเต็มพอดี เจ็ดบวกหนึ่งในเจ็ดของเจ็ดได้แปด คำตอบต้องเป็น 19 ไม่ใช่ 8 — ค่าเดาจึงคลาดไปเป็นตัวประกอบ $19/8$ คูณค่าเดาด้วยตัวประกอบเดียวกัน — 7 คูณ $19/8$ — ค่าเดาก็กลายเป็นคำตอบที่ถูกต้อง: เขียนอย่างที่อาลักษณ์อียิปต์เขียนเศษส่วน คำตอบจริงคือ 16 บวกครึ่ง บวกหนึ่งในแปด ภายหลังวิธีนี้ได้ชื่อว่า วิธีวางค่าลอง (*false position*) — *regula falsi* ในภาษาละตินที่จะปลุกมันขึ้นใหม่อีกกว่าสองพันปีให้หลัง — ตรรกะของมันฟังข้อเท็จจริงเพียงข้อเดียว: สมการนี้เป็นเชิงเส้น ค่าเดาตั้งต้นให้ผลอะไรก็ตาม คำตอบสุดท้ายคลาดไปด้วยสัดส่วนเดียวกันทุกจุด การคูณแก้ครั้งเดียวจึงจบเรื่อง (*Regula falsi*, Wikipedia; MacTutor, Egyptian papyri)

EN

Notice what has actually happened here. Nobody has "solved for x " in our sense. What Ahmes has done is compress an entire family of problems — any quantity plus any fraction of itself equals any target — into a fixed sequence of moves: guess something convenient, measure the error as a ratio, rescale. The genius is not in problem 24 alone. It is in the fact that problem 24 is *written down as a method*, separable from its numbers, so that a scribe training a thousand miles and a hundred years away can apply the same three moves to a different heap, a different fraction, a different target, and trust the answer. The route has become reusable. That, and not the number 16%, is what survived four thousand years to reach us.

TH

สิ่งที่รูดข้ามสี่พันปีมาถึงเราคือวิธี ไม่ใช่เลข 16% ไม่มีใคร "แก้หา x " ในความหมายของเรา Ahmes ปีบอัดโจทย์ทั้งตระกูล — จำนวนใดก็ได้ บวกเศษส่วนใดก็ได้ของตัวมันเอง เท่ากับเป้าหมายใดก็ได้ — ให้เหลือลำดับทำลายตัวชุดหนึ่ง: เค้าค่าที่สะดวกวัดความคลาดเป็นอัตราส่วน แล้วปรับมาตราส่วน อัจฉริยภาพไม่ได้อยู่ที่โจทย์ข้อ 24 ลำพัง มันอยู่ที่โจทย์ข้อ 24 เขียนไว้ในฐานะวิธี แยกจากตัวเลขของมันได้ อาลักษณ์ที่ฝึกลอยห่างออกไปพันไมล์และร้อยปี เอาสามก้าวเดียวกันไปใช้กับกองอีกกอง เศษส่วนอีกตัว เป้าหมายอีกค่า แล้วเชื่อคำตอบได้ เส้นทางกลายเป็นของใช้ซ้ำได้

Two Rivers, Two Habits of Mind

สองแม่น้ำ สองนิสัยทางความคิด

EN

A few hundred kilometers to the east, and largely without contact with Egyptian scribal schools, Babylonian mathematicians were solving harder problems with an even more explicit sense that they were writing *procedures*, not just answers. The clay tablet now catalogued as YBC 7289 — a small disc, easily held in one hand, incised sometime between 1800 and 1600 BC — shows a square with its diagonals drawn in, and beside one diagonal a sexagesimal (base-60) numeral that works out to an approximation of the square root of two accurate to about six decimal places (YBC 7289, Wikipedia). Other tablets, grouped by historians under the heading "procedure texts," go further: they lay out, in prose instructions rather than

worked numerical examples, the general steps for solving whole classes of problems — what we would now call quadratic equations, systems of two equations, canal-digging volumes — with the numbers in each problem serving as illustration rather than the point (MacTutor, Babylonian mathematics).

TH

นักคณิตศาสตร์บาบิโลนเขียนขั้นตอนวิธี ไม่ใช่แค่คำตอบ และรู้ตัวว่ากำลังทำอย่างนั้นชัดกว่าอียิปต์ด้วยซ้ำ พวกเขาอยู่ห่างออกไปทางตะวันออกไม่กี่ร้อยกิโลเมตร แต่ไม่ติดต่อกับสำนักอาลักษณ์อียิปต์ และแก้โจทย์ที่ยากกว่า แผ่นดินเหนียวที่วันนี้ขึ้นทะเบียนไว้ในชื่อ YBC 7289 — แผ่นกลมเล็ก ถือมือเดียวได้ สลักขึ้นช่วงใดช่วงหนึ่งของระหว่าง 1800 ถึง 1600 ปีก่อนคริสตกาล — แสดงรูปสี่เหลี่ยมจัตุรัสที่ลากเส้นทแยงมุมไว้ ข้างเส้นทแยงมุมเส้นหนึ่งมีตัวเลขฐานหกสิบ (sexagesimal) ที่คิดออกมาเป็นค่าประมาณของรากที่สองของสอง แม่นถึงราวหกตำแหน่งทศนิยม (YBC 7289, Wikipedia) แผ่นอื่นที่นักประวัติศาสตร์จัดกลุ่มไว้ได้หัวข้อ "ตัวบทขั้นตอนวิธี" (procedure texts) ไปไกลกว่านั้น: มันวางขั้นตอนทั่วไปสำหรับแก้โจทย์ทั้งชั้นไว้เป็นคำสั่งร้อยแก้ว ไม่ใช่ตัวอย่างตัวเลขที่แก้ให้ดู — สิ่งที่เราวันนี้เรียกว่าสมการกำลังสองระบบสมการสองตัวแปร ปริมาตรการขุดคลอง — ตัวเลขในแต่ละโจทย์เป็นเพียงภาพประกอบ ไม่ใช่สาระ (MacTutor, Babylonian mathematics)

EN

The distinction between the Egyptian and Babylonian habits is subtle but important for this book's argument. Egyptian texts, the Rhind papyrus included, read overwhelmingly as worked examples: solve *this* bread problem, then *that* one, each new problem a fresh occasion to apply a family of techniques the student is expected to notice by pattern-matching across dozens of cases. Babylonian procedure texts more often state the *general* steps first — "take this, do that, then this" — with a single concrete numerical case appended to show the abstract instructions actually run. The difference is the difference between a cookbook that only ever shows finished dishes and one that occasionally pauses to write down the technique of braising, once, so it can be pointed to instead of re-demonstrated every time. Both traditions "wrote the route," but the Babylonian tablets took one step further toward writing the route in a form abstracted from the partic-

ular number that happened to prompt it — a step toward what a modern reader would recognize as an algorithm, a word that would not exist for another two and a half thousand years, when it was coined from the name of the Persian mathematician al-Khwarizmi (his story belongs to chapter 05).

TH

ความต่างระหว่างนิสัยอียิปต์กับบาบิโลนละเอียดอ่อน แต่สำคัญต่อข้อเสนอของหนังสือเล่มนี้ ตัวบทอียิปต์ รวมทั้ง Rhind papyrus อ่านแล้วเป็นตัวอย่างแก้ไขให้ดูแทบทั้งหมด: แก้โจทย์ขนมปังข้อนี้ แล้วก็ข้อนั้น โจทย์ใหม่แต่ละข้อคือโอกาสให้ใช้เทคนิคตระกูลหนึ่ง ผู้เรียนต้องจับแบบแผนเอาเองจากกรณีนับสิบ ตัวบทขั้นตอนวิธีของบาบิโลนมักบอกขั้นตอนทั่วไปก่อน — "เอาอันนี้ ทำอย่างนั้น แล้วอย่างนี้" — แล้วต่อท้ายด้วยกรณีตัวเลขรูปธรรมกรณีเดียว เพื่อแสดงว่าคำสั่งนามธรรมนั้นเดินได้จริง มันคือความต่างระหว่างตำราอาหารที่โชว์แต่งานที่เสร็จแล้ว กับตำราที่หยุดจดเทคนิคการเคี้ยวไว้ครั้งหนึ่ง เพื่อชี้กลับไปที่ยังไม่ได้แทนการสาธิตซ้ำทุกครั้ง ทั้งสองสาย "เขียนเส้นทางลงไว้" แต่แผ่นบาบิโลนก้าวไปอีกก้าว: เขียนเส้นทางในรูปที่ถอดออกจากตัวเลขเฉพาะที่บังเอิญเป็นต้นเหตุให้เขียน นั่นคือก้าวหนึ่งสู่สิ่งที่ผู้อ่านสมัยใหม่จำได้ว่าเป็นอัลกอริทึม คำนี้จะยังไม่มีอยู่ไปอีกสองพันห้าร้อยปี จนบัญญัติขึ้นจากชื่อนักคณิตศาสตร์เปอร์เซีย al-Khwarizmi (เรื่องของเขาอยู่ในบทที่ 05)

E N

This matters for the thesis of this book because it is the earliest visible instance of a pattern that recurs at every later stage: a problem's difficulty shrinks the moment its *solution procedure*, not merely its solution, becomes an object that can be inspected, criticized, and improved by someone other than the person who first found it. A single answer helps one person once. A written method helps everyone, forever, with every future instance of the same shape of problem — provided someone bothers to write it down as a method rather than an anecdote.

T H

ความยากของปัญหาตกลงทันทีที่ขั้นตอนการแก้ของมัน ไม่ใช่แค่คำตอบ กลายเป็นวัตถุที่คนอื่นนอกจากผู้ค้นพบคนแรกหยิบมาส่อง วิจัย และปรับปรุงได้ นี่คือการครั้งแรกสุดที่มองเห็นได้ของแบบแผนที่จะกลับมาซ้ำในทุกช่วงถัดจากนี้ และนั่นคือเหตุที่มันสำคัญต่อข้อเสนอของหนังสือเล่มนี้ คำตอบข้อเดียวช่วยคนคนเดียวได้ครั้งเดียว วิธีที่เขียนไว้ช่วยทุกคน ตลอดไป กับทุกกรณีในอนาคตที่มีรูปทรงเดียวกัน — ขอเพียงมีใครสักคนยอมเขียนมันไว้ในฐานะวิธี ไม่ใช่เกร็ดเล่า

Euclid's Machine

เครื่องจักรของ Euclid

E N

Around 300 BC, in the newly founded city of Alexandria, a mathematician about whom almost nothing personal is reliably known — not his birthplace, not his exact dates, barely even the certainty that "Euclid" refers to one person rather than a school — assembled thirteen books that would become the single most reprinted mathematics text in history: the *Elements* (MacTutor, Euclid; Euclid's Elements, Wikipedia). What makes the *Elements* the hinge of this chapter is not any single theorem inside it — most of the geometry was not new, gathered as it was from generations of earlier Greek mathematicians whose individual proofs are largely lost. What Euclid did that nobody had done at this scale before was turn the entire discipline of geometry into a single, explicit, checkable *machine* for producing new truths out of old ones.

TH

ราว 300 ปีก่อนคริสตกาล ในเมือง Alexandria ที่เพิ่งก่อตั้ง หนังสือสิบสามเล่มถูกประกอบขึ้น และมันจะกลายเป็นตำราคณิตศาสตร์ที่พิมพ์ซ้ำมากที่สุดในประวัติศาสตร์: Elements (MacTutor, Euclid; Euclid's Elements, Wikipedia) ผู้ประกอบคือนักคณิตศาสตร์ที่แทบไม่มีเรื่องส่วนตัวใครรู้ได้อย่างเชื่อถือได้ — ไม่รู้บ้านเกิด ไม่รู้ปีเกิดปีตายที่แน่ชัด แทบไม่มีแม้ความมั่นใจว่า "Euclid" หมายถึงคนคนเดียว หรือหมายถึงสำนัก บานพับของบทนี้ไม่ใช่ทฤษฎีบทข้อใดข้อหนึ่งใน Elements — เรขาคณิตส่วนใหญ่ไม่ใช่ของใหม่ รวบรวมมาจากนักคณิตศาสตร์กรีกหลายรุ่น ก่อนหน้า บทพิสูจน์ของแต่ละคนสูญหายไปเกือบหมด สิ่งที Euclid ทำ และไม่มีใครเคยทำในระดับนี้มาก่อน คือเปลี่ยนวิชาเรขาคณิตทั้งวิชาให้เป็นเครื่องจักรเครื่องเดียว ชัดแจ้ง ตรวจสอบได้ ผลิตความจริงใหม่ออกมาจากความจริงเก่า

EN

The machine has three moving parts, and Euclid states all of them, in order, at the start of Book I: 23 definitions of the basic terms (point, line, angle...); five postulates, which are unproved starting assumptions specific to geometry (a straight line can be drawn between any two points; all right angles are equal to each other; and so on); and five further "common notions," unproved assumptions general enough to apply outside geometry entirely (things equal to the same thing are equal to each other) (Euclid's Elements, Wikipedia). Everything that follows — all 465 propositions across all thirteen books — is required to be derivable from only these starting points, using only two permitted construction tools (an unmarked straightedge and a compass) and strict logical inference. A proposition is not allowed to smuggle in a fact that "looks obviously true" from a diagram; it must trace back, step by step, to the postulates. You can read the opening definitions today essentially as Euclid wrote them, in the Greek text still preserved and studied at digital archives such as the Perseus Project (Perseus, Euclid's Elements, Book I).

TH

เครื่องจักรนี้มีชิ้นส่วนขยับได้สามชิ้น และ Euclid แจกไว้ครบตามลำดับที่ต้นเล่ม I: บทนิยาม 23 ข้อของคำพื้นฐาน (จุด เส้น มุม...) ข้อตั้งห้าข้อ คือข้อสมมติตั้งต้นที่ไม่พิสูจน์และเฉพาะกับเรขาคณิต (ลากเส้นตรงระหว่างจุดสองจุดก็ได้ มุมฉากทุกมุม

เท่ากันหมด และอื่นๆ) และ "ความคิดสามัญ" อีกห้าข้อ ไม่พิสูจน์เช่นกัน แต่กว้างพอจะใช้นอกเรขาคณิตได้ทั้งหมด (สิ่งที่เท่ากับสิ่งเดียวกัน ย่อมเท่ากันเอง) (Euclid's Elements, Wikipedia) ทุกอย่างตามมา — ข้อเสนอทั้ง 465 ข้อตลอดสิบสามเล่ม — ต้องอนุมานจากจุดตั้งต้นเหล่านี้เท่านั้น ด้วยเครื่องมือการสร้าง (construction) ที่อนุญาตเพียงสองชิ้น (ไม้บรรทัดไร้ขีดกับวงเวียน) และการอนุมานเชิงตรรกะที่เคร่งครัด ข้อเสนอใดจะลักลอบเอาข้อเท็จจริงที่ "ดูจริงอยู่แล้ว" จากรูปวาดเข้ามาไม่ได้ มันต้องสาวกลับทีละขั้นไปถึงข้อตั้ง บทนิยามเปิดเรื่องยังอ่านได้ทุกวันนี้เกือบตรงตามที่ Euclid เขียน ในตัวบทภาษากรีกที่เก็บรักษาและศึกษากันอยู่ในคลังดิจิทัลอย่าง Perseus Project (Perseus, Euclid's Elements, Book I)

EN

This is the payoff of writing method down taken to its logical extreme. Ahmes wrote one procedure per problem. Euclid wrote the *rules for generating and checking any procedure whatsoever* within an entire field. Once the axioms and permitted moves are fixed and public, a claim is no longer "true because a respected geometer says so" — it is true if and only if a specific, traceable, reproducible chain of construction and inference connects it back to five postulates and five common notions. Any sufficiently patient reader, in any century, with no access to Euclid himself, can verify any proposition in the *Elements* by re-walking that chain. This is why the book survived, essentially unchanged in method, as the standard geometry textbook in the West for well over two thousand years, and why "Euclidean" still means, to a mathematician, not merely "about flat-plane geometry" but "built by the axiomatic machine."

TH

Ahmes เขียนขั้นตอนวิธีหนึ่งชุดต่อโจทย์หนึ่งข้อ Euclid เขียนกฎสำหรับสร้างและตรวจขั้นตอนวิธีใดก็ตามภายในทั้งวิชา นี่คือผลตอบแทนของการเขียนวิธีลงไว้ ผลักไปจนสุดทางตรรกะของมัน พอสัจพจน์และทำที่อนุญาตตั้งแน่และเปิดเผยต่อสาธารณะแล้ว ข้ออ้างหนึ่งก็ไม่ได้ "จริงเพราะนักเรขาคณิตที่คนนับถือบอกว่าจริง" อีกต่อไป มันจริงก็ต่อเมื่อมีโซ่ของการสร้างและการอนุมานเส้นหนึ่งทีเงาะจง สาวตามได้ ทำซ้ำได้ เชื่อมมันกลับไปหาข้อตั้งห้าข้อกับความคิดสามัญห้าข้อ ผู้อ่านที่อดทนพอ ในศตวรรษใดก็ตาม ไม่ต้องเข้าถึงตัว Euclid เลย ตรวจข้อเสนอใดใน

Elements ก็ได้ด้วยการเดินโซ่เส้นนั้นซ้ำ หนังสือเล่มนี้จึงอยู่รอดเป็นตำราเรขาคณิตมาตรฐานของตะวันตกมากกว่าสองพันปีโดยวิธีแทบไม่เปลี่ยน และคำว่า "Euclidean" ในหูของนักคณิตศาสตร์จึงยังหมายถึง ไม่ใช่แค่ "ว่าด้วยเรขาคณิตบนระนาบแบน" แต่ "สร้างขึ้นด้วยเครื่องจักรเชิงสัจพจน์"

EN

The cost of building such a machine is a certain narrowing. Euclid's tools are fixed — straightedge and compass, nothing else — which is precisely why three problems the ancient world cared about (doubling the cube, trisecting an arbitrary angle, squaring the circle) resisted solution inside his system for two thousand years; they are not solvable with those tools, a fact only proved with nineteenth-century algebra. That narrowing is not a flaw. It is the design: a method earns the power to *guarantee* correctness by being explicit about exactly what it will and will not allow itself to do. Chapter 06 returns to this theme in its harshest form, when the same impulse — state precisely what is and is not permitted — produces a mathematics of provable impossibility.

TH

เครื่องจักรแบบนี้มีต้นทุน: มันแคบลง เครื่องมือของ Euclid ตรึงตายตัว — ไม้บรรทัด ไรซ์ติดกับวงเวียน ไม่มีอย่างอื่น — โจทย์สามข้อที่โลกโบราณใส่ใจ (เพิ่มปริมาตรลูกบาศก์เป็นสองเท่า แบ่งมุมใดๆ ออกเป็นสามส่วน สร้างสี่เหลี่ยมจัตุรัสที่มีพื้นที่เท่าวงกลม) จึงด้านคำตอบภายในระบบของเขาอยู่สองพันปี มันแก้ไม่ได้ด้วยเครื่องมือชุดนั้น และข้อเท็จจริงนี้พิสูจน์ได้ก็ต่อเมื่อมีพีชคณิตของศตวรรษที่สิบเก้า การแคบลงไม่ใช่ข้อบกพร่อง มันคือการออกแบบ: วิธีหนึ่งได้อ่านาจารรับประกันความถูกต้องมา ด้วยการพูดให้ชัดว่าตัวเองจะยอมทำอะไรและไม่ทำอะไร บทที่ 06 กลับมาที่ประเด็นนี้ในรูปแบบที่แรงที่สุด ตอนที่แรงผลึกเดียวกัน — ระบุให้แม่นยำว่าอะไรอนุญาตและอะไรไม่อนุญาต — ผลิตคณิตศาสตร์ของความเป็นไปไม่ได้ที่พิสูจน์ได้ออกมา

The Two Archimedeses

Archimedes สองคน

EN

A century after Euclid, on the island of Sicily, Archimedes of Syracuse left behind two utterly different accounts of how a mathematician actually *finds* things — and the gap between them is one of the most instructive in this book.

TH

Archimedes แห่ง Syracuse ที่บันทึกไว้สองชุดว่านักคณิตศาสตร์หาของเจอได้อย่างไรจริงๆ และสองชุดนั้นต่างกันคนละขั้ว เรื่องเกิดบนเกาะ Sicily หนึ่งศตวรรษหลัง Euclid ช่องว่างระหว่างสองบันทึกคือหนึ่งในช่องว่างที่สอนอะไรได้มากที่สุดในห้องเรียนเล่มนี้

EN

The first account is not by Archimedes at all. It comes from the Roman architect and engineer Vitruvius, writing in the preface to Book IX of his treatise *On Architecture* sometime in the first century BC — roughly two hundred years after Archimedes' death (Vitruvius, *De Architectura* IX, Perseus). King Hiero of Syracuse, the story goes, commissioned a golden crown and, suspecting the goldsmith of adulterating it with silver, asked Archimedes to determine the truth without damaging the crown. Stepping into a bath and noticing the water rise in proportion to the volume of his own submerged body, Archimedes supposedly realized he could compare the crown's volume (via displaced water) against its weight, catching the fraud — and, in his excitement, ran naked through the streets of Syracuse shouting "Eureka," "I have found it." The story is vivid, quotable, and almost certainly embellished: Vitruvius is our earliest surviving source, writing long after the fact, and modern commentators have pointed out that the water-displacement method as described would have been far too imprecise, given ancient measuring technology, to reliably detect the small amount of silver a goldsmith might realistically substitute (*Scientific American*, "Fact or Fiction?"). Whether or not it happened as told, the story survives because it captures something people badly want a discovery to feel like: sudden, embodied, and involuntary — a bath, not a proof.

TH

บันทึกชุดแรกไม่ได้มาจาก Archimedes เลย มันมาจาก Vitruvius สถาปนิกและวิศวกรชาวโรมัน เขียนไว้ในคำนำเล่ม IX ของตำรา On Architecture ช่วงใดช่วงหนึ่งในศตวรรษแรกก่อนคริสตกาล — ราวสองร้อยปีหลังการตายของ Archimedes (Vitruvius, De Architectura IX, Perseus) เรื่องเล่าว่ากษัตริย์ Hiero แห่ง Syracuse สั่งทำมงกุฎทองคำ สงสัยว่าช่างทองเจือเงินลงไป จึงขอให้ Archimedes หาความจริงโดยไม่ทำมงกุฎเสียหาย พอก้าวลงอ่างอาบน้ำแล้วเห็นน้ำเอ่อขึ้นตามปริมาตรของร่างกายที่จมลงไป Archimedes ก็ฉุกคิดได้ — ตามเรื่องเล่า — ว่าเทียบปริมาตรของมงกุฎ (จากน้ำที่ล้น) กับน้ำหนักของมันได้ จับการโก่งได้ แล้วด้วยความตื่นเต้นก็วิ่งแก้ผ้าไปตามถนนใน Syracuse ตะโกน "Eureka" — "ข้าพเจ้าเจอแล้ว" เรื่องนี้เห็นภาพชัด ยกไปเล่าต่อง่าย และแทบแน่ว่าแต่งเติมกันมา Vitruvius คือแหล่งเก่าสุดที่ยังเหลืออยู่ เขียนหลังเหตุการณ์นานมาก นักวิจารณ์สมัยใหม่ชี้ว่าวิธีวัดน้ำล้นอย่างที่ยังบรรยายไว้หายากเกินไปสำหรับเทคโนโลยีการวัดของยุคนั้น จับปริมาณเงินเพียงเล็กน้อยที่ช่างทองจะเจือลงไปได้จริงไม่ไหว (Scientific American, "Fact or Fiction?") จะเกิดขึ้นตามที่เล่าหรือไม่ก็ตาม เรื่องนี้อยู่รอดเพราะมันจับสิ่งที่คนอยากให้การค้นพบเป็นไว้ได้: ฉับพลัน เกิดกับเนื้อตัว ห้ามไม่อยู่ — อ่างอาบน้ำ ไม่ใช่บทพิสูจน์

EN

The second account is by Archimedes himself, and it describes something almost the opposite of a flash of bathtub insight: a deliberate, two-stage discipline for doing mathematics. In a treatise called *The Method of Mechanical Theorems*, addressed as a letter to Eratosthenes — the polymath running the Library of Alexandria, whose own solution appears in chapter 07 — Archimedes reveals a technique he had been keeping semi-private. To find the area or volume of a shape (a parabolic segment, a sphere, a cylinder), he did not start with rigorous geometric proof. He started by imagining the shape sliced into infinitely many thin strips, and mentally *weighing* those strips against strips of a shape he already understood, balanced on an imaginary lever, using the law of the lever from his own work in mechanics (*The Method of Mechanical Theorems*, Wikipedia). This procedure is not rigorous by Greek geometric standards — it treats an area as if it were literally made of an infinite stack of one-dimensional lines, a move that would trouble mathematicians until the invention of calculus two thousand years later. But

it works: it tells Archimedes, quickly and reliably, what the *answer* is. Only afterward did he go back and construct a fully rigorous proof of the same result using the accepted method of exhaustion, the laborious double-contradiction argument Euclid's tradition required. In his own words to Eratosthenes, he explains that he is sending the results first, without their proofs, precisely because finding the proof of a result you already know to be true is far easier than finding a true result and its proof simultaneously from nothing.

T H

บันทึกชุดที่สองเป็นของ Archimedes เอง และมันบรรยายวินัยสองจังหวะที่จิตใจแทบตรงข้ามกับการหยั่งเห็นวาทเดียวในอ่างน้ำ ดำราชื่อ The Method of Mechanical Theorems เขียนเป็นจดหมายถึง Eratosthenes — ปราชญ์รอบด้านผู้ดูแลหอสมุดแห่ง Alexandria คำตอบของเขาเองจะปรากฏในบทที่ 07 — ในจดหมายนั้น Archimedes เปิดเผยเทคนิคที่เก็บไว้กึ่งส่วนตัวมาตลอด จะหาพื้นที่หรือปริมาตรของรูปทรงหนึ่ง (ส่วนของพาราโบลา ทรงกลม ทรงกระบอก) เขาไม่เริ่มจากบทพิสูจน์เชิงเรขาคณิตที่เคร่งครัด เขาเริ่มจากจินตนาการว่าชอยรูปทรงนั้นเป็นแผ่นบางจำนวนอนันต์ แล้วชั่งน้ำหนักแผ่นเหล่านั้นในใจ เทียบกับแผ่นของรูปทรงที่เขาเข้าใจอยู่แล้ว บนคานงัดในจินตนาการ ด้วยกฎคานงัดจากงานกลศาสตร์ของเขาเอง (The Method of Mechanical Theorems, Wikipedia) ขั้นตอนนี้ไม่เคร่งครัดตามมาตรฐานเรขาคณิตกรีก — มันปฏิบัติต่อพื้นที่ราวกับประกอบขึ้นจากกองเส้นมิติเดียวจำนวนอนันต์จริงๆ ทำนี้จะรบกวนใจนักคณิตศาสตร์ไปจนกว่าจะมีการคิดค้นแคลคูลัสอีกสองพันปีให้หลัง แต่มันใช้ได้ มันบอก Archimedes ได้เร็วและเชื่อถือได้ว่าคำตอบคืออะไร จากนั้นเท่านั้นเขาจึงย้อนกลับไปสร้างบทพิสูจน์ที่เคร่งครัดเต็มรูปของผลลัพธ์เดียวกัน ด้วยวิธีไล่ให้หมด (method of exhaustion) ที่ยอมรับกัน คือข้อโต้แย้งแบบขัดแย้งสองชั้นอันหนักหน่วงที่ธรรมเนียมของ Euclid เรียกร้อง เขาเขียนถึง Eratosthenes ด้วยถ้อยคำของตัวเองว่าส่งผลลัพธ์มาก่อนโดยไม่มีบทพิสูจน์ ก็เพราะการหาบทพิสูจน์ของผลลัพธ์ที่รู้อยู่แล้วว่าจริง ง่ายกว่าการหาผลลัพธ์ที่จริงพร้อมบทพิสูจน์ของมันจากศูนย์ไปพร้อมกันมาก

E N

That is a completely different confession about method than the bath story offers, and it is arguably the more radical one: Archimedes is telling us, in

his own hand, that mathematical discovery and mathematical rigor are *two separate activities*, performable in either order, and that doing the ugly, un-rigorous, mechanically-assisted one first is not a shortcut to be ashamed of — it is a legitimate research strategy. *The Method* was itself lost for most of history and known only by title until 1906, when the Danish scholar Johan Ludvig Heiberg identified it underneath a thirteenth-century Byzantine prayer book — a palimpsest, a manuscript scraped and reused — now known as the Archimedes Palimpsest. Large portions remained illegible until a further imaging campaign between 1998 and 2008, using ultraviolet, infrared, and X-ray techniques at what is now the Walters Art Museum in Baltimore, recovered text hidden under mold damage and even forged medieval paintings added by a twentieth-century forger trying to increase the manuscript's resale value (Archimedes Palimpsest project). We are, in other words, still recovering Archimedes' record of his own method eighteen centuries after Vitruvius wrote down the more entertaining but less trustworthy story about his bath.

T H

Archimedes กำลังบอกเราด้วยลายมือของเขาเองว่า การค้นพบทางคณิตศาสตร์กับความเคร่งครัดทางคณิตศาสตร์เป็นกิจกรรมสองอย่างที่แยกจากกัน ทำสลับลำดับกันก็ได้ และการทำอย่างที่น่าเกลียด ไม่เคร่งครัด อาศัยกลไกช่วย ก่อน ไม่ใช่ทางลัดที่ต้องละเลย — มันคือยุทธศาสตร์การวิจัยที่ชอบธรรม นี่คือคำสารภาพเรื่องวิธีที่ต่างจากเรื่องอ่างอาบน้ำโดยสิ้นเชิง และเถียงได้ว่าถอนรากกว่า ตัว *The Method* เอง สูญหายไปเกือบตลอดประวัติศาสตร์ รู้จักกันแค่ชื่อ จนปี 1906 นักวิชาการชาวเดนมาร์ก Johan Ludvig Heiberg ระบุมันได้ใต้หนังสือสวดมนต์ไบแซนไทน์ของศตวรรษที่สิบสาม — พาลิมป์เซสต์ (palimpsest) คือต้นฉบับที่ขูดออกแล้วเอากลับมาใช้ใหม่ — วันนี้รู้จักกันในชื่อ Archimedes Palimpsest เนื้อความส่วนใหญ่ยังอ่านไม่ออกจนมีการถ่ายภาพอีกรอบระหว่างปี 1998 ถึง 2008 ด้วยเทคนิคอัลตราไวโอเลต อินฟราเรด และเอกซเรย์ ที่สถานที่ซึ่งวันนี้คือ Walters Art Museum ในเมือง Baltimore จึงกู้ตัวบทที่ซ่อนอยู่ได้รอยราและได้ภาพวาดยุคกลางปลอม ที่นักปลอมแปลงในศตวรรษที่ยี่สิบเติมลงไปเพื่อดันราคาขายต่อของต้นฉบับกลับคืนมาได้ (Archimedes Palimpsest project) พุดอีกอย่าง เรายังกู้บันทึกที่

Archimedes เขียนถึงวิธีของตัวเองอยู่ สิบแปดศตวรรษหลังจากที่ Vitruvius จดเรื่องอ่างอาบน้ำที่สนุกกว่าแต่เชื่อถือได้น้อยกว่าเอาไว้

Ibn al-Haytham's Room

ห้องของ Ibn al-Haytham

EN

A thousand years after Archimedes, in Cairo, a scholar named Abu Ali al-Hasan ibn al-Haytham — known in Latin translation as Alhazen — was, by most accounts, confined to his own house for roughly a decade, from around 1011 to 1021. The reason given by several historians is almost too neat: he had rashly told the volatile Fatimid caliph al-Hakim that he could regulate the annual flooding of the Nile with a dam, was sent to survey the site near modern Aswan, and concluded on arrival that the project was impossible with the era's engineering. Fearing execution for the failure — al-Hakim's court had a documented record of brutal punishment for far smaller offenses — Ibn al-Haytham is said to have feigned madness to be spared, and was instead placed under house arrest until the caliph's death in 1021, at which point he was released as no longer legally responsible for a promise made while "insane" (Physics World, "The Scholar and the Caliph" — note that this source itself flags the account as reconstructed from incomplete records, calling it merely "likely"). What is much better attested is what he did with those ten years: he wrote a seven-volume treatise, the *Kitab al-Manazir*, known in the West as the *Book of Optics* (UNESCO Courier; Book of Optics, Wikipedia).

TH

หนึ่งพันปีหลัง Archimedes ที่ Cairo นักปราชญ์ชื่อ Abu Ali al-Hasan ibn al-Haytham — ฉบับแปลละตินเรียกว่า Alhazen — ถูกกักอยู่แต่ในบ้านของตัวเองราวหนึ่งทศวรรษ ตั้งแต่ราวปี 1011 ถึง 1021 ตามบันทึกส่วนใหญ่ เหตุผลที่นักประวัติศาสตร์หลายคนให้ไว้ลงตัวเกินไปจนน่าสงสัย: เขาเสนอทูลกาหลิบฟาติมิดผู้อารมณ์ร้าย al-Hakim ว่าคุมน้ำท่วมประจำปีของแม่น้ำไนล์ได้ด้วยเขื่อน กาหลิบส่งเขาไปสำรวจพื้นที่ใกล้เมือง Aswan ปัจจุบัน พอไปถึงเขาสรุปว่าโครงการเป็นไปได้ด้วยวิศวกรรมของยุคนั้น เขากลับถูกประหารเพราะความล้มเหลว — ราชสำนักของ

al-Hakim มีบันทึกการลงโทษโหดร้ายต่อความผิดที่เล็กกว่านั้นมาก — ว่ากันว่า Ibn al-Haytham จึงแกล้งเป็นบ้าเพื่อเอาตัวรอด และลงเอยด้วยการกักบริเวณในบ้านแทน จนกาหลิบสิ้นพระชนม์ในปี 1021 เขาจึงได้รับการปล่อยตัว เพราะไม่ต้องรับผิดตามกฎหมายต่อคำสัญญาที่ให้ไว้ตอน "วิกิลจริต" อีกต่อไป (Physics World, "The Scholar and the Caliph" — พึงสังเกตว่าแหล่งนี้เองก็ระบุว่าเรื่องนี้ประกอบขึ้นใหม่จากบันทึกที่ไม่ครบ และเรียกมันเพียงว่า "น่าจะเป็นเช่นนั้น") สิ่งที่มีหลักฐานหนักแน่นกว่ามากคือสิ่งที่เขาทำกับสิบปีนั้น: ตำราเจ็ดเล่มชื่อ Kitab al-Manazir ที่ตะวันตกรู้จักในชื่อ Book of Optics (UNESCO Courier; Book of Optics, Wikipedia)

E N

The Book of Optics addressed a question the ancient Greeks had never settled: does the eye see by sending rays out toward objects (the "extramission" theory associated with Euclid's own optical writings), or by receiving light that travels from objects into the eye ("intromission")? Ibn al-Haytham did not simply argue for intromission — he built physical devices to test it, describing his apparatus and procedures in enough concrete, reproducible detail that a reader with no access to him could set up the same experiment. His most famous device, a forerunner of the pinhole camera, used a small aperture in a screen between a set of candles and a viewing wall: light from each candle, he showed, cast a single, distinct, invertible spot on the far wall, corresponding to a straight line from that candle through the hole — evidence that light was moving from the object toward the observer along definite straight paths, not the reverse (UNESCO Courier; Book of Optics, Wikipedia).

T H

Book of Optics จัดการกับคำถามที่ชาวกรีกโบราณไม่เคยยุติได้: ตามองเห็นด้วยการยิงรังสีออกไปหาวัตถุ (ทฤษฎี "extramission" ที่ผูกอยู่กับงานด้านทัศนศาสตร์ของ Euclid เอง) หรือด้วยการรับแสงที่เดินทางจากวัตถุเข้ามาในตา ("intromission") Ibn al-Haytham ไม่ได้เพียงเถียงเข้าข้าง intromission — เขาสร้างอุปกรณ์จริงขึ้นมาทดสอบ และบรรยายเครื่องมือกับขั้นตอนไว้เป็นรูปธรรม ทำซ้ำได้ มากพอที่ผู้อ่านซึ่งไม่มีทางเข้าถึงตัวเขาจะตั้งการทดลองเดียวกันขึ้นเองได้ อุปกรณ์ที่ดังที่สุดของเขา

เป็นบรรพบุรุษของกล้องรูเข็ม: ช่องเล็กบนฉากกั้นระหว่างเทียนชุดหนึ่งกับผนังที่ใช้ดู เขาแสดงว่าแสงจากเทียนแต่ละเล่มทอดจุดเดียว คมชัด กลับหัว ลงบนผนังฝั่งตรงข้าม ตรงกับเส้นตรงที่ลากจากเทียนเล่มนั้นผ่านรู — หลักฐานว่าแสงเคลื่อนจากวัตถุมาหาผู้สังเกตตามเส้นทางตรงที่แน่นอน ไม่ใช่ทางกลับกัน (UNESCO Courier; Book of Optics, Wikipedia)

EN

This deserves the same caution this book applies elsewhere: calling Ibn al-Haytham the sole "father of the scientific method," a label popular accounts attach to him often, overstates a more gradual, distributed history that includes earlier Greek and later European experimentalists, and some historians of science push back hard on the tidier version (The Renaissance Mathematicus, "Ibn al-Haytham did not invent the Scientific Method"). What is not in dispute is narrower and, for this book's purposes, more important: the *Book of Optics* is a written record of an experimental procedure — apparatus, setup, prediction, result — precise enough that others who had never met its author could pick it up and re-run it. Translated into Latin around 1200 as *De Aspectibus*, it shaped the optics of Roger Bacon, Witelo, and eventually Kepler, centuries after its author's death — outliving him by exactly the mechanism this chapter keeps returning to: he wrote the route down clearly enough that it needed no further help from him to keep working.

TH

การเรียก Ibn al-Haytham ว่าเป็น "บิดาแห่งวิธีการทางวิทยาศาสตร์" แต่ผู้เดียว — ฉลากที่บทความทั่วไปติดให้เขาบ่อยๆ — พุดเกินจริง ประวัติศาสตร์จริงค่อยเป็นค่อยไปและกระจายตัวกว่านั้น รวมนักทดลองกรีกรุ่นก่อนและนักทดลองยุโรปรุ่นหลังไว้ด้วย และนักประวัติศาสตร์วิทยาศาสตร์บางคนค้านฉบับที่เรียบร้อยเกินจริงน้อยอย่างแรง (The Renaissance Mathematicus, "Ibn al-Haytham did not invent the Scientific Method") หนังสือเล่มนี้ใช้ความระมัดระวังแบบเดียวกันในที่อื่น และใช้ที่นี่ด้วย สิ่งที่ไม่มิข้อโต้แย้งแคบกว่า และสำคัญกว่าสำหรับจุดประสงค์ของหนังสือเล่มนี้: *Book of Optics* คือบันทึกลายลักษณ์ของขั้นตอนวิธีเชิงการทดลอง — อุปกรณ์ การตั้งค่า คำทำนาย ผลลัพธ์ — แม่นยำพอที่คนซึ่งไม่เคยพบผู้เขียนจะหยิบไปรันซ้ำได้ แปลเป็นละตินราวปี 1200 ในชื่อ *De Aspectibus* มันหล่อหลอมทัศนศาสตร์ของ

Roger Bacon, Witelo และในที่สุดคือ Kepler หลายศตวรรษหลังผู้เขียนตายไปแล้ว มั่นอยู่มากว่าเขาด้วยกลไกที่บนทึบนั้นกลับมาพูดถึงตลอด: เขาเขียนเส้นทางลงไว้ชัด พอจนมันไม่ต้องพึ่งเขาอีกเลยเพื่อจะทำงานต่อไป

Descartes' Unfinished Machine

เครื่องจักรที่ยังไม่เสร็จของ Descartes

EN

By the early seventeenth century, a French soldier-turned-philosopher named René Descartes had noticed the same thing this chapter has been arguing and decided to attack it head-on: if geometry could be turned into a machine (Euclid) and if a scholar under house arrest could turn optics into a reproducible experimental procedure (Ibn al-Haytham), why should there not be *one single method*, stated once, that would work for discovering truth in any field at all — physics, medicine, ethics, mathematics alike?

TH

René Descartes ทหารฝรั่งเศสที่ผันตัวเป็นนักปรัชญา เห็นสิ่งเดียวกับที่บทนี้เถียงมาตลอด และเข้าชนมันตรงๆ ในต้นศตวรรษที่สิบเจ็ด: ถ้าเรหาคณิตกลายเป็นเครื่องจักรได้ (Euclid) และถ้านักปราชญ์ที่ถูกกักอยู่ในบ้านตัวเองเปลี่ยนทัศนศาสตร์ให้เป็นขั้นตอนการทดลองที่ทำซ้ำได้ (Ibn al-Haytham) ทำไมจะไม่มียุวิธีเดียว แจงไว้ครั้งเดียว แล้วใช้ค้นความจริงได้ในทุกสาขา — ฟิสิกส์ การแพทย์ จริยศาสตร์ คณิตศาสตร์ เหมือนกันหมด

EN

Descartes made two attempts at writing this universal method down. The first, composed around 1628 and left unfinished, was the *Regulae ad Directionem Ingenii* — *Rules for the Direction of the Mind*. It was not published in his lifetime; the earliest edition appeared only in 1701, more than fifty years after his death (Rules for the Direction of the Mind, Wikipedia). In its fourth rule, Descartes names explicitly what he is chasing: a *mathesis universalis*, a "universal mathematics," which he describes as "a general science which explains all that can be sought concerning order and measure, restricted to no special subject matter" (Mathesis universalis, Wikipedia; MacTutor, Descartes). That is an extraordinary ambition stated plainly: not a better

way to do geometry, but the master method underneath geometry, algebra, and every other exact science — the pattern Euclid had built for one subject, generalized to all subjects at once.

TH

Descartes ลงมือเขียนวิธีสากลนี้สองครั้ง และครั้งแรกไม่จบ ราวปี 1628 เขาเขียน *Regulae ad Directionem Ingenii* — กฎสำหรับกำกับปัญญา — แล้วค้างไว้ มันไม่ได้ตีพิมพ์ตอนเขายังอยู่ ฉบับพิมพ์เก่าสุดออกมาในปี 1701 กว่าห้าสิบปีหลังเขาตาย (*Rules for the Direction of the Mind*, Wikipedia) กฎข้อที่สี่บอกตรงๆ ว่า Descartes ได้ทำอะไร: *mathesis universalis* หรือ "คณิตศาสตร์สากล" ที่เขาบรรยายว่าเป็น "ศาสตร์ทั่วไปที่อธิบายทุกสิ่งที่สืบค้นได้เกี่ยวกับลำดับและการวัด โดยไม่จำกัดอยู่กับเนื้อหาเฉพาะเรื่องใด" (*Mathesis universalis*, Wikipedia; MacTutor, Descartes) นั่นคือความทะเยอทะยานขนาดใหญ่ที่พูดออกมาได้ดังๆ: ไม่ใช่วิธีทำเรขาคณิตที่ดีกว่าเดิม แต่คือวิธีแม่บทที่รองรับได้เรขาคณิต พีชคณิต และวิทยาศาสตร์แม่นยำอื่นๆทุกแขนง — แบบแผนที่ Euclid สร้างไว้สำหรับวิชาเดียว ขยายไปคลุมทุกวิชาพร้อมกัน

EN

The *Regulae* stops abruptly, unfinished, partway through its planned three parts. Descartes' second and more famous attempt, nine years later, is more modest in scope but far more influential in form: the 1637 *Discourse on the Method*, written in French rather than the scholarly Latin, framed as something closer to autobiography than treatise. Descartes narrates his own dissatisfaction with everything he had been taught and distills his working method into four rules: accept nothing as true unless it presents itself so clearly and distinctly that there is no occasion to doubt it; divide every difficulty into as many parts as necessary to resolve it; conduct one's thoughts in order, beginning with the simplest objects and ascending by degrees to the more complex; and, throughout, make enumerations so complete, and reviews so general, that nothing is omitted (*Discourse on the Method*, Wikipedia).

TH

Regulae หยุดกลางคัน ไม่จบ เดินไปได้แค่บางส่วนของสามภาคที่วางไว้ ความพยายามครั้งที่สองของ Descartes เก็บไว้ให้หลังดังกว่า ขอบเขตถ่อมกว่า แต่ทรง

อิทธิพลอย่างมากในเชิงรูปแบบ: Discourse on the Method ปี 1637 เขียนเป็นภาษาฝรั่งเศสแทนภาษาละตินของวงวิชาการ วางกรอบให้ใกล้ชิดชีวิตประจำวันมากกว่าตำรา Descartes เล่าความไม่พอใจของตัวเองต่อทุกอย่างที่เคยเรียนมา แล้วค้นวิธีทำงานของเขาเองเลือกกฎสี่ข้อ: ยอมรับสิ่งใดว่าจริง เว้นแต่มันปรากฏชัดและแจ่มแจ้งจนไม่มีช่องให้สงสัย แบ่งความยากทุกอันเป็นส่วนย่อยให้มากเท่าที่จำเป็นจะคลี่มันได้ นำความคิดไปตามลำดับ เริ่มจากวัตถุที่ง่ายที่สุดแล้วไต่ทีละขั้นสู่สิ่งที่ซับซ้อนกว่า และตลอดทางนั้น แจงรายการให้ครบถ้วนและทบทวนให้กว้างขวางจนไม่มีอะไรตกหล่น (Discourse on the Method, Wikipedia)

EN

Appended to the *Discourse* as one of three scientific essays illustrating the method in action was *La Géométrie*, in which Descartes showed how to attach a pair of numbers to every point in a plane and rewrite every geometric curve as an algebraic equation (*La Géométrie*, Britannica). That single technical result — coordinates — turns out to be so consequential for the argument of this book that it earns its own extended treatment in chapter 02, as the master case of changing a problem's representation rather than merely following a fixed procedure within one. For now, notice only the shape of the ambition that produced it: Descartes wanted a method that would make discovery itself mechanical, the way Euclid had made geometric proof mechanical. He never finished building it — the *mathesis universalis* remains, by his own account, a program rather than a completed system, and philosophers still argue about how much of it he actually believed was achievable. But the dream he left behind — that solving, in general, might be systematized rather than merely practiced — is the ambition every later chapter of this book, in its own century's vocabulary, keeps trying to make good on.

TH

พิกัดคือผลลัพธ์เชิงเทคนิคขั้นเดียวที่สำคัญต่อข้อเสนอของหนังสือเล่มนี้มากจนได้พื้นที่ขยายเป็นของตัวเองในบทที่ 02 มันมาจาก *La Géométrie* หนึ่งในสามบทความวิทยาศาสตร์ที่ต่อท้าย Discourse เพื่อแสดงวิธีนี้ตอนทำงานจริง Descartes แสดงวิธีติดคู่ตัวเลขให้ทุกจุดบนระนาบ แล้วเขียนเส้นโค้งเรขาคณิตทุกเส้นใหม่ให้เป็นสมการพีชคณิต (*La Géométrie*, Britannica) บทนั้นจะใช้มันเป็นกรณีแม่บทของการ

เปลี่ยนรูปแบบการแทนปัญหา (representation) แทนที่จะแค่เดินตามขั้นตอนวิธีตายตัวอยู่ภายในกรอบเดิม ตอนนี้ให้สังเกตแค่รูปทรงของความทะเยอทะยานที่ผลิตมันขึ้นมา: Descartes อยากได้วิธีที่ทำให้การค้นพบเองเป็นเรื่องกลไก แบบเดียวกับที่ Euclid ทำให้บทพิสูจน์เรขาคณิตเป็นเรื่องกลไก เขาสร้างมันไม่เสร็จ — *mathesis universalis* ยังคงเป็นแผนงาน ไม่ใช่ระบบที่สมบูรณ์ ตามคำของเขาเอง และนักปรัชญายังเถียงกันอยู่ว่าเขาเชื่อจริงๆ แค่ไหนว่ามันทำสำเร็จได้ แต่ความฝันที่เขาทิ้งไว้ — ว่าการแก้ปัญหโดยทั่วไปอาจทำให้เป็นระบบได้ ไม่ใช่แค่ฝึกทำเอา — คือความทะเยอทะยานที่ทุกบทถัดจากนี้ของหนังสือเล่มนี้พยายามทำให้เป็นจริง ด้วยศัพท์ของศตวรรษของตัวเอง

Method Becomes an Object

วิธีกลายเป็นวัตถุ

EN

Step back from the four thousand years this chapter has covered and one pattern holds across Egypt, Babylon, Alexandria, Cairo, and Descartes' stove-heated room in Germany where he claims to have had his founding insight: in every case, someone took a way of solving problems that had previously lived only inside a single skilled person's head — a scribe's trick, a geometer's construction, a natural philosopher's experimental setup, a philosopher's private rules of thought — and wrote it down as an object separate from any particular problem it had been used to solve.

TH

แบบแผนหนึ่งยืนอยู่ตลอดสี่พันปีที่บั้นนี้ครอบคลุม ทั้งในอียิปต์ บาบิโลน Alexandria, Cairo และห้องที่มีเตาผิงในเยอรมนีที่ Descartes อ้างว่าเขาเกิดการหยั่งเห็น (insight) ตั้งต้นขึ้นที่นั่น: ทุกกรณี มีใครสักคนหยิบวิธีแก้ปัญหาก็่อนหน้านั้นอยู่แต่ในหัวของคนเก่งคนเดียว — เล่ห์ของอาลักษณ์ การสร้างของนักเรขาคณิต การตั้งการทดลองของนักปรัชญาธรรมชาติ กฎความคิดส่วนตัวของนักปรัชญา — แล้วเขียนมันลงไว้ในฐานะวัตถุที่แยกจากปัญหาข้อใดข้อหนึ่งที่มันเคยใช้แก้

EN

That act of writing is not mere record-keeping. It is the first and most fundamental kind of compression this book will describe. A method that lives

only in one mind can only be applied by that mind, to problems that mind happens to encounter, using whatever the mind happens to remember correctly on a given day. A method that has been written down — named, with its steps made explicit, its assumptions stated, its scope of application at least implicitly bounded — becomes something that other people can inspect, test against new cases, criticize, and improve, without needing to reconstruct it from scratch or consult its inventor. Ahmes' false position, Euclid's postulates, Archimedes' lever-balancing heuristic, Ibn al-Haytham's pinhole apparatus, and Descartes' four rules are five different answers to the same underlying design question — *how do we solve this class of problem, in general?* — and every one of them survives, in some form, into the mathematics and science practiced today.

TH

การเขียนนั้นคือการบีบอัด (compression) ชนิดแรกสุดและพื้นฐานที่สุดที่หนังสือเล่มนี้จะบรรยายถึง ไม่ใช่แค่การเก็บบันทึก วิธีที่อยู่แต่ในหัวคนเดียวใช้ได้แต่กับหัวนั้น กับปัญหาที่หัวนั้นบังเอิญเจอ ด้วยสิ่งที่หัวนั้นบังเอิญจำได้ถูกต้องในวันนั้น วิธีที่เขียนลงไว้แล้ว — มีชื่อ ขั้นตอนพูดออกมาชัด ข้อสมมติแจ้งไว้ ขอบเขตการใช้งานอย่างน้อยก็กำหนดไว้โดยนัย — กลายเป็นของที่คนอื่นส่องดู ทดสอบกับกรณีใหม่ วิจารณ์ และปรับปรุงได้ ไม่ต้องประกอบมันขึ้นใหม่จากศูนย์ ไม่ต้องไปถามผู้คิดค้น false position ของ Ahmes, ข้อตั้งของ Euclid, วิธีเดาอย่างมีหลัก (heuristic) แบบซึ่งน้ำหนักบนคานงัดของ Archimedes, อุปกรณ์รูเข็มของ Ibn al-Haytham และกฎสี่ข้อของ Descartes คือคำตอบห้าแบบต่อคำถามการออกแบบข้อเดียวกันที่อยู่ข้างใต้ — เราแก้ปัญหานี้โดยทั่วไปอย่างไร — และทุกคำตอบอยู่รอดมาในรูปใดรูปหนึ่ง จนถึงคณิตศาสตร์และวิทยาศาสตร์ที่ทำกันอยู่ทุกวันนี้

EN

This groundwork makes the rest of this book possible. You cannot deliberately change the space a problem lives in — the master move this book spends its remaining chapters tracking — until you have noticed that the space has a *shape* worth examining, distinct from any one instance sitting inside it. Writing the route down, rather than just reaching the answer, is what first makes that shape visible.

TH

การเขียนเส้นทางลงไว้ แทนที่จะแค่ไปถึงคำตอบ คือสิ่งแรกที่ทำให้พื้นที่ปัญหา (problem space) มีรูปทรงที่มองเห็นได้ แยกออกจากกรณีใดกรณีหนึ่งที่นั่งอยู่ข้างใน จะสนใจเปลี่ยนพื้นที่ที่ปัญหาหนึ่งอาศัยอยู่ — ท่าแม่บทที่หนังสือเล่มนี้ใช้บทที่เหลือทั้งหมดไล่ตาม — ต้องสังเกตเห็นเสียก่อนว่าพื้นที่นั้นมีรูปทรงที่ควรค่าแก่การตรวจสอบ งานพื้นฐานชุดนี้จึงทำให้ส่วนที่เหลือของหนังสือเล่มนี้เป็นไปได้

The Move

ทำประจำบท

EN

Move 01 — Turn the method into an object.

TH

Move 01 — เปลี่ยนวิธีให้เป็นวัตถุ

EN

- **Write the route, not just the answer.** When you solve something, the number or outcome is often the least valuable thing you produced. The sequence of moves that got you there — stated plainly enough that someone else could follow it without you in the room — is the reusable asset.
- **Name it so it can be improved by someone who isn't you.** A technique without a name lives and dies with its inventor's memory. A named, written procedure — "false position," "the axiomatic method," "the four rules" — becomes a target other people can criticize, test, and sharpen.
- **Separate finding from proving.** Archimedes' letter to Eratosthenes is explicit permission, twenty-two centuries old, to solve a problem by

whatever ugly, mechanical, half-rigorous means gets you the right answer fastest — and only afterward go back and build the clean argument. Chasing rigor and certainty simultaneously, from a blank page, is usually the slower path to both.

- **State your starting assumptions before you build on them.** Euclid's postulates and Descartes' four rules both do this: they say, explicitly and in advance, what is being taken for granted, so that everything constructed afterward can be checked against a fixed, visible foundation rather than an unspoken intuition.

TH

- **เขียนเส้นทาง ไม่ใช่แค่คำตอบ** เวลาทำอะไรได้ ตัวเลขหรือผลลัพธ์มักเป็นของที่มีค่าน้อยที่สุดในบรรดาสิ่งที่ผลิตออกมา ลำดับทำที่พาไปถึงตรงนั้นต่างหาก — แง่ไว้ชัดพอที่คนอื่นจะเดินตามได้โดยไม่ต้องมีเจ้าตัวอยู่ในห้อง — คือสินทรัพย์ที่ใช้ซ้ำได้
- **ตั้งชื่อให้มัน เพื่อให้คนที่ไม่ใช่เจ้าตัวปรับปรุงมันได้** เทคนิคที่ไม่มีชื่อเกิดและตายไปพร้อมความทรงจำของผู้คิดค้น ขั้นตอนวิธีที่มีชื่อและเขียนไว้ — "false position" "วิธีเชิงสัจพจน์" "กฎสี่ข้อ" — กลายเป็นเป้าที่คนอื่นวิจารณ์ ทดสอบ และลับให้คมได้
- **แยกการหาออกจากการพิสูจน์** จดหมายของ Archimedes ถึง Eratosthenes คือไบอัญญาตชัดๆ อายุยี่สิบสองศตวรรษ ให้แก้ปัญหาด้วยวิธีน่าเกลียด กลไกจำ กิ่งเครื่องครัด อะไรก็ได้ที่พาไปถึงคำตอบที่ถูกเร็วที่สุด — แล้วค่อยย้อนกลับไปสร้างข้อโต้แย้งที่สะอาดทีหลัง การไล่ล่าความเคร่งครัดกับความแน่ใจไปพร้อมกันจากหน้ากระดาษเปล่า มักเป็นเส้นทางที่ช้ากว่าสำหรับทั้งสองอย่าง
- **แฉข้อสมมติตั้งต้นก่อนจะสร้างอะไรทับลงไป** ข้อตั้งของ Euclid กับกฎสี่ข้อของ Descartes ทำแบบนี้ทั้งคู่: มันบอกไว้ชัดและบอกไว้ล่วงหน้าว่ากำลังถืออะไรเป็นที่ยึด ทุกอย่าง que สร้างต่อจากนั้นจึงตรวจกลับไปพื้นฐานรากที่ตรึงแน่นและมองเห็นได้ ไม่ใช่ที่สัญชาตญาณที่ไม่เคยพูดออกมา

E N

Next — Chapter 02 · The Reframe: a bridge-crossing puzzle so trivial Euler almost didn't bother, a schoolboy's shortcut whose true story is stranger than its legend, and a high jumper who solved "jump higher" by refusing to face the bar. Once method is an object, the next move is to change the space that object operates in.

T H

ต่อไป — บทที่ 02 · การเปลี่ยนกรอบ: ปริศนาข้ามสะพานที่กระจอกจน Euler เกือบไม่สนใจ ทางลัดของเด็กนักเรียนที่เรื่องจริงแปลกกว่าตำนานของมัน และนักกระโดดสูงที่แก้โจทย์ "กระโดดให้สูงขึ้น" ด้วยการไม่ยอมหันหน้าเข้าหาคาน พอวธีเป็นวัตถุแล้ว ทำถัดไปคือเปลี่ยนพื้นที่ที่วัตถุนั้นทำงานอยู่

CHAPTER 02

The Reframe

การเปลี่ยนกรอบ

Königsberg, 1735

Königsberg ปี 1735

EN

The East Prussian city of Königsberg sat astride the river Pregel, split by the water into four pieces of land — two riverbanks and two islands — connected to one another by seven bridges. Sunday strollers there had, for some time, amused themselves with a standing challenge: could a person walk a route through the city that crossed every one of the seven bridges exactly once, and no bridge twice? Nobody had managed it. Nobody had proved it was impossible either. It was, by every reasonable measure, a trivial puzzle — the kind of thing argued over on an afternoon walk, not brought before an Academy of Sciences.

TH

ปริศนาสะพานของ Königsberg เป็นของเล่นวันอาทิตย์ ไม่ใช่โจทย์วิชาการ เมือง Königsberg แห่งปรัสเซียตะวันออกตั้งคร่อมแม่น้ำ Pregel สายน้ำแยกเมืองเป็นผืนดินสี่ผืน — ตลิ่งสองฝั่งกับเกาะสองเกาะ — สะพานเจ็ดสะพานเชื่อมทั้งหมดเข้าด้วยกัน คนเดินเล่นวันอาทิตย์ที่นั่นเล่นโจทย์ค้างคาอยู่ข้อหนึ่งมาพักใหญ่แล้ว: เดินทะลุเมืองโดยข้ามสะพานครบทั้งเจ็ด สะพานละครั้งพอดี ไม่ซ้ำสักสะพาน ทำได้ไหม ไม่มีใครทำได้ ไม่มีใครพิสูจน์ได้ว่าเป็นไปไม่ได้เช่นกัน วัตด้วยเกณฑ์สมเหตุสมผลข้อไหนก็ตาม มันคือปริศนากระจอก — ของที่เถียงกันระหว่างเดินเล่นตอนบ่าย ไม่ใช่ของที่จะเอาไปวางต่อหน้า Academy of Sciences

EN

It reached Leonhard Euler anyway, relayed through a chain of acquaintances: Heinrich Kühn, a mathematics professor in Danzig, mentioned it to Carl Leonhard Gottlieb Ehler, the city's mayor and an amateur enthusiast,

who wrote to Euler in St. Petersburg asking for a solution "worthy of your great genius." Euler's first reaction, preserved in a letter he wrote on March 13, 1736 to Giovanni Marinoni, an Italian astronomer at the Viennese court, is almost dismissive: "This question is so banal, but seemed to me worthy of attention in that neither geometry, nor algebra, nor even the art of counting was sufficient to solve it." He goes on to wonder, half to himself, whether the puzzle belongs to a discipline the philosopher Gottfried Leibniz had once called for but never built — a *geometria situs*, a "geometry of position," concerned with arrangement rather than measurement (Sachs, Stiebitz & Wilson, "An Historical Note: Euler's Königsberg Letters," *Journal of Graph Theory*, 1988).

TH

มันไปถึง Leonhard Euler อยู่ดี ผ่านสายคนรู้จักเป็นทอดๆ: Heinrich Kühn ศาสตราจารย์คณิตศาสตร์ที่ Danzig เอ่ยเรื่องนี้กับ Carl Leonhard Gottlieb Ehler นายกเทศมนตรีของเมืองและนักคณิตศาสตร์สมัครเล่น Ehler เขียนจดหมายถึง Euler ที่ St. Petersburg ขอคำตอบที่ "คู่ควรกับอัจฉริยภาพอันยิ่งใหญ่ของท่าน" ปฏิกริยาแรกของ Euler แทบเป็นการปิดทั้ง จดหมายลงวันที่ 13 มีนาคม 1736 ถึง Giovanni Marinoni นักดาราศาสตร์อิตาลีในราชสำนักเวียนนา เก็บมันไว้: "คำถามนี้ ดาษด้นเหลือเกิน แต่ดูจะคู่ควรแก่ความสนใจอยู่ ตรงที่ทั้งเรขาคณิต ทั้งพีชคณิต หรือ แม้แต่ศิลปะแห่งการนับ ก็ไม่พอจะแก้มันได้" แล้วเขาก็ครุ่นคิดต่อครั่งๆ กับตัวเองว่า ปริศนานี้อาจอยู่ในวิชาที่นักปรัชญา Gottfried Leibniz เคยเรียกร้องให้มีแต่ไม่เคยสร้าง — *geometria situs* หรือ "เรขาคณิตของตำแหน่ง" วิชาที่สนใจการจัดวาง ไม่ใช่การวัด (Sachs, Stiebitz & Wilson, "An Historical Note: Euler's Königsberg Letters," *Journal of Graph Theory*, 1988)

EN

Three weeks later, replying directly to Ehler, Euler is blunter still — and reveals, in the same breath, exactly how little he thinks of the kind of solution he is about to hand over: "Thus you see, most noble Sir, how this type of solution bears little relationship to mathematics, and I do not understand why you expect a mathematician to produce it, rather than anyone else, for the solution is based on reason alone, and its discovery does not depend on any mathematical principle" (Sachs, Stiebitz & Wilson, 1988). Read that care-

fully. The man who is, at this exact moment, inventing what would become graph theory is telling a mayor, in writing, that what he has just done barely counts as mathematics. He is wrong — spectacularly, historically wrong — and the way he is wrong is the entire subject of this chapter.

TH

สามสัปดาห์ต่อมา Euler ตอบ Ehler โดยตรง และพูดตรงกว่าเดิม — ในลมหายใจเดียวกันเขาเผยว่าให้คำตอบที่กำลังจะยื่นให้น้อยเพียงใด: "ท่านยอมเห็นแล้ว ท่านผู้สูงศักดิ์ ว่าคำตอบชนิดนี้แทบไม่เกี่ยวข้องกับคณิตศาสตร์เลย และข้าพเจ้าไม่เข้าใจว่าเหตุใดท่านจึงคาดหวังให้นักคณิตศาสตร์เป็นผู้ผลิตมันออกมา แทนที่จะเป็นใครก็ได้ เพราะคำตอบนี้ตั้งอยู่บนเหตุผลล้วนๆ และการค้นพบมันไม่ได้ขึ้นอยู่กับหลักการทางคณิตศาสตร์ข้อใดเลย" (Sachs, Stiebitz & Wilson, 1988) อ่านตรงนี้ให้ดี ชายที่กำลังคิดค้นสิ่งที่จะกลายเป็นทฤษฎีกราฟอยู่ ณ วินาทีนั่นเอง บอกนายกเทศมนตรีเป็นลายลักษณ์อักษรว่า สิ่งที่เขาเพิ่งทำแทบนับเป็นคณิตศาสตร์ไม่ได้ เขาคิดผิด — ผิดครั้งใหญ่ ผิดในระดับที่ประวัติศาสตร์จดไว้ — และวิธีที่เขาผิดคือเนื้อหาทั้งหมดของบทนี้

EN

Here is what Euler actually did, presented to the St. Petersburg Academy on August 26, 1735 and published the following year as *Solutio problematis ad geometriam situs pertinentis* ("The solution of a problem relating to the geometry of position") (MAA Convergence; Seven Bridges of Königsberg, Wikipedia). He noticed that the exact streets a walker takes within a landmass, the exact spot on a bridge where a foot lands, the width of the river, the layout of the buildings — all of it is irrelevant to the question being asked. The *only* fact that matters is which landmasses each bridge connects. So he threw the city away. In its place he drew four points, one for each landmass, and seven lines connecting them, one for each bridge — the first drawing in history of what we would now instantly recognize as a graph. Every possible walking route through Königsberg's actual, cobblestoned, physically detailed streets became, in this new picture, nothing more than a sequence of dots visited by following lines.

T H

Euler โยนเมืองทิ้ง นั่นคือสิ่งที่เขาทำจริงๆ ในงานที่เสนอต่อ St. Petersburg Academy วันที่ 26 สิงหาคม 1735 และตีพิมพ์ปีถัดมาในชื่อ *Solutio problematis ad geometriam situs pertinentis* ("คำตอบของปัญหาที่เกี่ยวกับเรขาคณิตของตำแหน่ง") (MAA Convergence; Seven Bridges of Königsberg, Wikipedia) เขาสังเกตว่าถนนสายไหนที่คนเดินเลือกภายในผืนดินหนึ่ง จุดไหนบนสะพานที่ท้าทาย ความกว้างของแม่น้ำ ผังอาคาร — ทั้งหมดไม่เกี่ยวกับคำถามที่ถามอยู่เลย ข้อเท็จจริงเดียวที่มีน้ำหนักคือ สะพานแต่ละสะพานเชื่อมผืนดินไหนกับผืนดินไหน เขาจึงวาดจุดสี่จุดแทนเมือง จุดละหนึ่งผืนดิน แล้วลากเส้นเจ็ดเส้นเชื่อมจุดเหล่านั้น เส้นละหนึ่งสะพาน — ภาพวาดแรกในประวัติศาสตร์ของสิ่งทีวันนี้ใครก็จำได้ทันทีว่าเป็นกราฟ ในภาพใหม่นี้ ทุกเส้นทางเดินที่เป็นไปได้บนถนนปูหินจริงของ Königsberg พร้อมรายละเอียดทางกายภาพครบถ้วน เหลือเพียงลำดับของจุดที่แวะถึงโดยเดินตามเส้น

E N

Once the city has been stripped down to four dots and seven lines, the puzzle nearly solves itself. Consider any landmass that is not the start or end of the walk. Every time the walker arrives at it over one bridge, they must leave over a different one — arrivals and departures pair up, bridge by bridge. That means any landmass visited only in passing must have an *even* number of bridges touching it: each visit uses up exactly two, one in and one out. Count the bridges touching each of Königsberg's four landmasses in the real map, and every single one comes out odd — three, three, three, and five. A walk that starts and ends at different points can tolerate exactly two landmasses with an odd count (the start and the end); Königsberg has four. No route exists, full stop, for any walker, using any order of streets whatsoever — and Euler's argument proves it in roughly the space of two sentences, with no need to try a single one of the geographically enormous number of possible walking routes through the actual city.

T H

พอเมืองเหลือสี่จุดกับเจ็ดเส้น ปริศนาก็แทบแก้ตัวเอง พิจารณาผืนดินใดก็ได้ที่ไม่ใช่จุดเริ่มหรือจุดจบของการเดิน ทุกครั้งที่คนเดินมาถึงด้วยสะพานหนึ่ง เขาต้องออกไปด้วยอีกสะพานหนึ่ง — เขาเข้ากับขาออกจับคู่กันสะพานต่อสะพาน ผืนดินที่แค่แวะ

ผ่านจึงต้องมีสะพานแต่ละอยู่เป็นจำนวนคู่ เพราะการแวะแต่ละครั้งใช้สะพานไปสอง สะพานพอดี เข้าหนึ่ง ออกหนึ่ง นับสะพานที่แต่ละผืนดินทั้งสี่ผืนของ Königsberg บนแผนที่จริง ทุกผืนออกมาเป็นจำนวนคี่หมด — สาม สาม สาม และห้า การเดินที่เริ่มและจบคนละจุดยอมให้มีผืนดินจำนวนคี่ได้แค่สองผืนพอดี (จุดเริ่มกับจุดจบ) Königsberg มีสี่ เส้นทางนั้นไม่มีอยู่ จบ ไม่ว่าคนเดินเป็นใคร ไม่ว่าจะเรียงถนนแบบไหน — และข้อโต้แย้งของ Euler พิสูจน์ได้ในเนื้อที่ราวสองประโยค โดยไม่ต้องลองเดินสักเส้นทางเดียว จากเส้นทางที่เป็นไปได้ในเมืองจริงซึ่งมีจำนวนมากเกินจะไล่ครบ

EN

Notice what happened to the size of the problem. Before the reframe, "does a walk exist?" is a question about an intractably large space of possible paths through real streets, checkable only by trying route after route and hoping to either find one or run out of patience. After the reframe, it is a question about four small integers and whether they are even or odd — checkable, quite literally, on one hand. Nothing about the city changed. What changed was which of the city's features Euler agreed to see as part of the problem, and which he agreed to throw away as noise. That act of selective forgetting is not sloppiness. It is a proof, embedded in the argument itself, that the discarded detail could not possibly have affected the answer — which is exactly why the resulting two-line argument is not a cheat but a genuine solution, arguably a more complete one than exhaustively checking every route ever could be, since checking finitely many routes never rules out one you didn't think to try, while Euler's parity argument rules out all of them at once, forever, by construction.

TH

ขนาดของปัญหาคือสิ่งที่เปลี่ยน ก่อนการเปลี่ยนกรอบ (reframe) "มีเส้นทางเดินอยู่ไหม" เป็นคำถามเกี่ยวกับพื้นที่ของเส้นทางผ่านถนนจริง ซึ่งใหญ่เกินจะจัดการ ตรวจสอบได้ทางเดียวคือลองทีละเส้นทาง แล้วหวังว่าจะเจอสักเส้น หรือไม่ก็หมดความอดทนไปก่อน หลังเปลี่ยนกรอบ มันเป็นคำถามเกี่ยวกับจำนวนเต็มเล็กๆ สี่จำนวน ว่าป็นคู่หรือคี่ — นับได้บนมือข้างเดียวตามตัวอักษร เมื่อไม่ได้เปลี่ยนไปเลย สิ่งที่เปลี่ยนคือ Euler ยอมเห็นลักษณะไหนของเมืองเป็นส่วนของปัญหา และยอมทิ้งลักษณะไหนไป ในฐานะสัญญาณรรบวงวน การเลือกليمแบบนี้ไม่ใช่ความม้ง่าย มันคือบทพิสูจน์ที่ฝัง

อยู่ในข้อโต้แย้งเอง ว่ารายละเอียดที่ทิ้งไปไม่มีทางกระทบคำตอบ นั่นคือเหตุผลที่ข้อโต้แย้งสองบรรทัดนี้ไม่ใช่การโกง แต่เป็นคำตอบจริง และเถียงได้ด้วยว่าสมมุติว่าการไล่ตรวจทุกเส้นทางเสียอีก การตรวจเส้นทางจำนวนจำกัดไม่เคยตัดเส้นทางที่ยังนึกไม่ถึงออกไปได้ ข้อโต้แย้งเรื่องคู่ของ Euler ตัดทุกเส้นทางออกพร้อมกัน ตลอดไป โดยโครงสร้างของมันเอง

What a Representation Actually Is

รูปแบบการแทนปัญหาคืออะไรกันแน่

EN

Chapter 00 introduced the idea, borrowed from Allen Newell and Herbert Simon's 1972 formalization of problem spaces, that any problem can be described by a set of states, a set of legal moves between states, a starting state, and a goal. What Euler's bridges make vivid is that *this description is never handed to you by the world*. Someone — often you, often unconsciously — has to decide what counts as a "state" and what counts as a "move," and that decision is where nearly all of a problem's real difficulty actually lives.

TH

ความยากจริงเกือบทั้งหมดของปัญหาอยู่ที่การตัดสินใจอะไรนับเป็นสถานะและอะไรนับเป็นท่า บทที่ 00 แนะนำแนวคิดที่ยืมมาจากงานของ Allen Newell กับ Herbert Simon ปี 1972 ซึ่งทำให้พื้นที่ปัญหา (problem space) เป็นรูปนัยไว้แล้วว่า ปัญหาใดก็อธิบายได้ด้วยเซตของสถานะ (state) เซตของท่าที่ถูกดิดกาสำหรับย้ายจากสถานะหนึ่งไปอีกสถานะ สถานะตั้งต้น และเป้าหมาย สะพานของ Euler ทำให้เห็นชัดว่าโลกไม่เคยยื่นคำอธิบายนี้ให้เรา ต้องมีใครสักคน — บ่อยครั้งคือตัวเราเอง บ่อยครั้งโดยไม่รู้ตัว — ตัดสินใจว่าอะไรนับเป็น "สถานะ" และอะไรนับเป็น "ท่า" และความยากอาศัยอยู่ตรงการตัดสินใจนั่นเอง

EN

A representation, in this technical sense, is a choice about three things: which differences between situations you will bother to track as different *states*; which transformations between states you will permit yourself to call legal moves; and what you will count as the *goal* — the target configuration

that ends the search. Before Euler, the implicit representation of the bridge puzzle distinguished states by *exact physical location*: which specific street, which specific bridge underfoot, facing which direction. Under that representation the problem space is enormous, because two routes that differ only in some irrelevant meandering through a courtyard count as different states requiring separate consideration. Euler's representation instead distinguished states only by *which bridges remain uncrossed and which landmass the walker currently occupies* — a description that treats every meandering path within a single landmass as the same state, because for the question being asked, they are the same. He did not simplify the problem by ignoring difficulty. He simplified it by proving, correctly, that a huge amount of what looked like difficulty was never actually relevant to the answer.

TH

รูปแบบการแทนปัญหา (representation) ในความหมายทางเทคนิค คือการเลือกสามอย่าง: ความต่างระหว่างสถานการณ์อันไหนที่จะติดตามในฐานะสถานะคนละสถานะ การแปลงระหว่างสถานะอันไหนที่จะเรียกว่าท่าที่ถูกกติกา และอะไรที่จะนับเป็นเป้าหมาย — รูปแบบปลายทางที่ปิดการค้นหา ก่อน Euler representation ที่แฝงอยู่ในปริศนาสะพานแยกสถานะด้วยตำแหน่งทางกายภาพที่แน่นอน: ถนนเส้นไหน สะพานไหนได้ฝ่าเท้า หันหน้าไปทางไหน ภายใต้ representation นั้น พื้นที่ปัญหาใหญ่โต เพราะเส้นทางสองเส้นที่ต่างกันแค่การเดินอ้อมลานบ้านโดยไม่มีนัยสำคัญ ก็นับเป็นคนละสถานะที่ต้องพิจารณาแยกกัน representation ของ Euler แยกสถานะด้วยสะพานที่ยังไม่ได้ข้ามเหลืออยู่ที่สะพาน และคนเดินยืนอยู่บนผืนดินไหนตอนนี้เท่านั้น คำอธิบายแบบนี้ถือว่าทุกเส้นทางอ้อมภายในผืนดินเดียวกันคือสถานะเดียวกัน เพราะสำหรับคำถามที่ถามอยู่ มันคือสถานะเดียวกันจริง เขาไม่ได้ทำให้ปัญหาย่อยลงด้วยการเมินความยาก เขาพิสูจน์อย่างถูกต้องว่าสิ่งที่ดูเหมือนความยากก้อนใหญ่นั้นไม่เคยเกี่ยวกับคำตอบเลยตั้งแต่แรก

EN

This is the discipline that separates a good reframe from mere hand-waving, and it is worth stating plainly because it will recur, in different clothes, in every chapter that follows. Throwing away information you *shouldn't* throw away does not shrink a problem — it breaks it, producing a confident

wrong answer or no answer at all. Keeping information you *don't need* does not make a solution more careful — it just keeps the search space needlessly large. The right representation is the one that keeps exactly the distinctions the goal actually depends on, and collapses every distinction that doesn't change the answer into a single state. Finding that boundary — what to keep, what to discard — is not a mechanical step you can outsource to a procedure the way Ahmes' scribes outsourced arithmetic to false position. It is closer to a judgment call, made correctly by Euler here and, as chapters to come will show, made incorrectly by very smart people a great many times before someone else found the right cut.

TH

การเปลี่ยนกรอบที่ดีต่างจากการโบกมือลอยๆ ตรงวินัยข้อนี้ และควรพูดออกมาตรงๆ เพราะมันจะกลับมาในเสื้อผ้าชุดอื่นทุกบทหลังจากนี้ โยนข้อมูลที่ไม่ควรโยนทิ้ง ปัญหาไม่ได้เล็กลง — มันพัง ให้คำตอบผิดอย่างมั่นใจ หรือไม่ให้คำตอบเลย เก็บข้อมูลที่ไม่ต้องใช้ไว้ คำตอบไม่ได้รอบคอบขึ้น — พื้นที่ค้นหาแคใหญ่เกินจำเป็น representation ที่ถูกต้องคืออันที่เก็บความต่างที่เป้าหมายขึ้นอยู่กับมันจริงไว้พอดี และยุบทุกความต่างที่ไม่เปลี่ยนคำตอบให้เหลือสถานะเดียว การหาเส้นแบ่งนั้น — เก็บอะไร ทิ้งอะไร — ไม่ใช่ขั้นตอนกลไกที่มอบให้ขั้นตอนวิธีทำแทนได้ อย่างที่ อาลักษณ์ของ Ahmes มอบเลขคณิตให้วิธีวางค่าลอง (false position) มันใกล้เคียงกับการตัดสินใจด้วยวิจารณญาณมากกว่า Euler ตัดสินถูกในกรณีนี้ และอย่างที่บทถัดๆ ไปจะแสดง คนฉลาดมากตัดสินใจผิดมาหลายต่อหลายครั้ง ก่อนจะมีคนอื่นหาเส้นตัดที่ถูกเจอ

The Schoolroom That Keeps Changing Shape

ห้องเรียนที่เปลี่ยนรูปทรงอยู่เรื่อย

EN

Carl Friedrich Gauss is supposed to have performed the same kind of move, at a vastly smaller scale, as a schoolboy in Braunschweig sometime in the 1780s. A teacher — bored, or wanting quiet, depending on which telling you read — set the class the tedious busywork of adding up every whole number from 1 to 100. Young Gauss, the story goes, produced the correct answer, 5,050, almost instantly, while his classmates were still laboriously carrying sums. His method, as usually told: pair the numbers from opposite ends of

the list — 1 with 100, 2 with 99, 3 with 98 — and notice that every pair sums to the same value, 101. There are fifty such pairs. Fifty times 101 is 5,050. What began as one hundred sequential additions collapses into a single multiplication.

TH

Carl Friedrich Gauss เล่ากันว่าทำทำชนิดเดียวกันในสเกลเล็กกว่ามาก ตอนเป็นเด็กนักเรียนที่ Braunschweig ช่วงใดช่วงหนึ่งในทศวรรษ 1780 ครูคนหนึ่ง — เปือ หรืออยากได้ความเจียบ แล้วแต่สำนวนที่อ่าน — สั่งงานจุกจิกให้ทั้งชั้นบวกจำนวนเต็มทุกจำนวนตั้งแต่ 1 ถึง 100 เรื่องเล่าว่า Gauss ตัวน้อยให้คำตอบที่ถูก 5,050 แทนจะทันที ขณะที่เพื่อนร่วมชั้นยังทดเลขกันอยู่ วิธีของเขาตามที่เล่ากันทั่วไป: จับคู่จำนวนจากปลายทั้งสองข้างของรายการ — 1 กับ 100, 2 กับ 99, 3 กับ 98 — ทุกคู่บวกกันได้ค่าเดียวกันคือ 101 คู่แบบนั้นมีห้าสิบคู่ ห้าสิบคูณ 101 เท่ากับ 5,050 การบวกเรียงกันหนึ่งร้อยครั้งยุบลงเหลือการคูณครั้งเดียว

EN

That is a genuine and useful reframe, worth walking through carefully because it is structurally identical to Euler's, at a much smaller scale. The naive representation of "sum the numbers 1 to 100" treats the problem as a *sequence*: a running total, updated by one small addition at a time, a hundred steps from start to finish — a long, thin, unavoidably serial problem space. Gauss's representation instead treats the same hundred numbers as fifty *unordered pairs*, exploiting the fact that addition does not care what order its inputs arrive in. Reorganizing the numbers around their pair-sums, rather than around their sequence, throws away information (which number came "first") that the goal — the total — never needed in the first place, and replaces a hundred-step walk with a single leap.

TH

นี่คือการเปลี่ยนกรอบจริงและใช้ได้ผล ควรเดินดูให้ละเอียด เพราะโครงสร้างของมันเหมือน Euler ทุกประการ ต่างกันแค่สเกล representation แบบชื่อของ "บวกจำนวน 1 ถึง 100" มองปัญหาเป็นลำดับ: ยอดสะสมที่เพิ่มทีละชนิด หนึ่งร้อยก้าวจากต้นจนจบ — พื้นที่ปัญหาที่ยาว ผอม และเรียงตามลำดับโดยเรียงไม่ได้ representation ของ Gauss มองจำนวนร้อยตัวเดียวกันเป็นคู่ที่ไม่เรียงลำดับห้าสิบคู่ โดยอาศัยข้อเท็จจริงว่าการบวกไม่สนใจว่าอินพุตมาถึงตามลำดับไหน จัดจำนวนใหม่

รอบผลบวกของคู่แทนที่จะรอบลำดับของมัน ก็คือโยนข้อมูล (ว่าจำนวนไหนมา "ก่อน") ที่เป้าหมาย — ยอดรวม — ไม่เคยต้องใช้ทิ้งไป แล้วแทนการเดินร้อยก้าวด้วยการกระโดดครั้งเดียว

E N

But — and this is where the chapter's stated house rule about disputed anecdotes has to be honored rather than smoothed over — there is real reason to doubt that this specific, vivid version of the story is an accurate record of an actual 1780s classroom. The science writer Brian Hayes went looking for the story's origin for a 2006 *American Scientist* column, "Gauss's Day of Reckoning," and traced its earliest known appearance to an 1856 memorial biography of Gauss written after his death by Wolfgang Sartorius von Waltershausen — more than seventy years after the event it describes, based on Sartorius's personal recollections of conversations with an elderly Gauss, not on any contemporary record from the schoolroom itself. Hayes then tracked the story through dozens of later retellings and found the details drifting from version to version: the exact numbers being summed change, the teacher's name and motive change, and the method Gauss is credited with using is not always the pairing trick — some versions have him using a different, though related, arithmetic shortcut entirely (Hayes, "Gauss's Day of Reckoning," *American Scientist*, 2006). The honest conclusion is not "the story is false" — something like it plausibly happened, and Gauss undeniably possessed exactly this kind of number sense from childhood — but that the crisp, cinematic version most people repeat has been shaped by a hundred and seventy years of retelling into something cleaner than the historical record can actually support. As chapter 00 already flagged in passing, the same caution that applies to Alexander at Gordium applies here: a story can be pedagogically true and biographically unverifiable at the same time, and a book about clear thinking should say so rather than pretend otherwise.

T H

มีเหตุผลหนักแน่นให้สงสัยว่าเรื่องฉบับเจาะจงเห็นภาพชัดนี้ไม่ใช่บันทึกที่ตรงกับห้องเรียนจริงในทศวรรษ 1780 — และตรงนี้ก็ก้ำกึ่งกับเรื่องเกร็ดที่ยังพิพาทกัน ต้องได้รับการเคารพ ไม่ใช่กลบให้เรียบ นักเขียนวิทยาศาสตร์ Brian Hayes ตามหาต้นตอของเรื่องนี้เพื่อคอลัมน์ใน *American Scientist* ปี 2006 ชื่อ "Gauss's Day of

Reckoning" และสาวการปรากฏครั้งแรกเท่าที่รู้จักไปถึงชีวประวัติที่ระลึกของ Gauss ปี 1856 ที่ Wolfgang Sartorius von Waltershausen เขียนหลัง Gauss ตาย — กว่าจะเจ็ดสิบปีหลังเหตุการณ์ที่มันบรรยาย และอาศัยความทรงจำส่วนตัวของ Sartorius จากบทสนทนากับ Gauss ในวัยชรา ไม่ใช่บันทึกร่วมสมัยจากห้องเรียนนั่นเอง จากนั้น Hayes ได้ตามเรื่องนี้ผ่านการเล่าซ้ำหลายสิบสำนวนในยุคหลัง และพบว่ารายละเอียดเลื่อนไหลจากฉบับหนึ่งสู่อีกฉบับ: จำนวนที่บวกกันเปลี่ยน ชื่อครูและแรงจูงใจของครูเปลี่ยน และวิธีที่ยกให้เป็นของ Gauss ก็ไม่ใช่เล่ห์จับคู่เสมอไป — บางฉบับให้เขาใช้ทางลัดเชิงเลขคณิตอีกแบบไปเลย แม้จะเกี่ยวข้องกัน (Hayes, "Gauss's Day of Reckoning," *American Scientist*, 2006) ข้อสรุปที่ชัดเจนไม่ใช่ "เรื่องนี้เท็จ" — อะไรทำนองนี้น่าจะเกิดขึ้นจริง และ Gauss มีสัมผัสเชิงจำนวนแบบนี้มาตั้งแต่เด็กโดยปฏิเสธไม่ได้ — แต่คือฉบับคมกริบราวภาพยนตร์ที่คนส่วนใหญ่เล่าต่อกันนั้น ผ่านการเล่าซ้ำมาหนึ่งร้อยเจ็ดสิบปีจนสะอาดเกินกว่าหลักฐานทางประวัติศาสตร์จะรองรับ อย่างที่บทที่ 00 เกริ่นผ่านไปแล้ว ความระมัดระวังชุดเดียวกับที่ใช้กับ Alexander ที่กอร์เดียมใช้ที่นี่ด้วย: เรื่องหนึ่งเป็นจริงในเชิงการสอนและตรวจสอบไม่ได้ในเชิงชีวประวัติพร้อมกันได้ และหนังสือที่วาดด้วยการคิดให้ชัดเจนพูดออกมาจริงๆ แทนที่จะแก้งทำเป็นอื่น

E N

What survives the uncertainty is the mathematics, and the mathematics does not need the schoolboy to be real to make its point. Whether or not a nine-year-old actually stunned a Braunschweig classroom in the 1780s, the pairing trick is a clean, small-scale demonstration of exactly the move this chapter is about: a problem that looks sequential and slow under one representation (running sum, one hundred additions) becomes a single fast operation under another (fifty constant-sum pairs, one multiplication), and nothing about the underlying arithmetic changed to make that possible — only the arrangement in which it was viewed.

T H

คณิตศาสตร์รอดพ้นความไม่แน่นอนนั้นมาได้ และมันไม่ต้องอาศัยเด็กนักเรียนคนนั้นให้มีอยู่จริงเพื่อบอกประเด็นของมัน ไม่ว่าเด็กเก๋ახวบจะทำให้ห้องเรียนที่

Braunschweig ตะลึงในทศวรรษ 1780 จริงหรือไม่ เล่ห์จับคู่ก็คือการสาธิตสเกลเล็กที่สะอาดของท่าที่บนนี้พูดถึงพอดี: ปัญหาที่ดูเรียงลำดับและซ้ำภายใต้

representation หนึ่ง (ยอดสะสม บวกเรื่อยๆ ครั้ง) กลายเป็นปฏิบัติการเร็วครั้งเดียวภายใต้อีก representation หนึ่ง (หาคู่ที่ผลบวกคงที่ คูณครั้งเดียว) เลขคณิตข้างใต้ไม่ได้เปลี่ยนเลยเพื่อให้เป็นแบบนั้นได้ เปลี่ยนแค่การจัดวางที่เรามอง

Descartes' Bridge

สะพานของ Descartes

EN

Chapter 01 already introduced René Descartes as a man chasing a universal method for all discovery. One specific technical result from that chase deserves a closer look here, because it may be the single most productive representation change in the history of mathematics: the coordinate system he laid out in the 1637 appendix *La Géométrie*.

TH

ระบบพิกัดที่ René Descartes วางไว้ในภาคผนวก *La Géométrie* ปี 1637 อาจเป็นการเปลี่ยน representation ที่ให้ผลผลิตสูงที่สุดขึ้นเดียวในประวัติศาสตร์คณิตศาสตร์ บทที่ 01 แนะนำเขาไปแล้วในฐานะชายที่ไล่ล่าวิธีสากลสำหรับการค้นพบทุกชนิด ผลลัพธ์เชิงเทคนิคชิ้นนี้จากการไล่ล่านั้นสมควรได้รับการดูใกล้ๆ ตรงนี้ด้วยเหตุผลนั่นเอง

EN

Before Descartes, geometry and algebra were, for most practical purposes, two separate toolkits addressing two separate kinds of question. Geometry, in the Euclidean tradition, worked with figures — circles, lines, triangles — manipulated by physical-feeling constructions with compass and straightedge, and proved things about them through chains of logical argument grounded in diagrams. Algebra worked with symbols and equations, manipulated by symbolic rules, largely detached from any picture. A circle, geometrically, was "the set of all points a fixed distance from a center" — a definition you could draw but not easily calculate with. Descartes' innovation was to fix a pair of perpendicular reference lines — what we now call axes — and describe every point in the plane by its distance along each (*La Géométrie*, Britannica). Once every point is a pair of numbers, every curve through those points becomes an equation relating the numbers, and vice versa: that same circle is now simply every pair (x, y) satisfying $x^2 + y^2 = r^2$. Finding where two curves intersect — a question that used to require an inventive geometric construction, different for every pair of shapes — becomes a single, mechanical act of solving two equations simultaneously, the

same technique regardless of whether the curves in question are circles, parabolas, or shapes nobody had ever successfully drawn together before.

TH

ก่อน Descartes เรขาคณิตกับพีชคณิตเป็นกล่องเครื่องมือสองกล่องแยกกัน ตอบคำถามสองชนิดแยกกัน ในทางปฏิบัติแทบทุกกรณี เรขาคณิตในธรรมเนียมของ Euclid ทำงานกับรูป — วงกลม เส้น สามเหลี่ยม — จัดการด้วยการสร้างที่ให้ความรู้สึกทางกายภาพด้วยวงเวียนกับไม้บรรทัด แล้วพิสูจน์เรื่องของรูปเหล่านั้นผ่านข้อโต้แย้งเชิงตรรกะที่ยึดกับรูปวาด พีชคณิตทำงานกับสัญลักษณ์และสมการ จัดการด้วยกฎเชิงสัญลักษณ์ แทบไม่ผูกกับภาพใด ในทางเรขาคณิต วงกลมคือ "เซตของจุดทั้งหมดที่อยู่ห่างจากจุดศูนย์กลางเป็นระยะคงที่" — นิยามที่วาดได้ แต่คำนวณด้วยยาก นวัตกรรมของ Descartes คือตรึงเส้นอ้างอิงตั้งฉากกันคู่หนึ่ง — ที่วันนี้เรียกว่าแกน — แล้วอธิบายทุกจุดบนระนาบด้วยระยะทางตามแกนแต่ละแกน (La Géométrie, Britannica) พอทุกจุดเป็นคู่ตัวเลข ทุกเส้นโค้งที่ผ่านจุดเหล่านั้นก็เป็นสมการที่โยงตัวเลขเข้าด้วยกัน และกลับกันด้วย: วงกลมวงเดิมตอนนี้คือทุกคู่ (x, y) ที่สอดคล้องกับ $x^2 + y^2 = r^2$ เท่านั้นเอง การหาว่าเส้นโค้งสองเส้นตัดกันตรงไหน — คำถามที่เคยต้องอาศัยการสร้างเชิงเรขาคณิตอย่างมีชั้นเชิง ต่างกันไปทุกคู่รูปทรง — กลายเป็นการแก้สมการสองสมการพร้อมกันครั้งเดียวอย่างกลไก เทคนิคเดียวกันไม่ว่าเส้นโค้งคู่นั้นจะเป็นวงกลม พาราโบลา หรือรูปทรงที่ไม่เคยมีใครวาดคู่กันสำเร็จมาก่อน

EN

This is a representation change in exactly Euler's sense, just aimed at an entire discipline rather than one puzzle. The "states" of a geometric problem — points, curves, intersections — get redescribed as algebraic objects, and the "operators" — construct this line, bisect that angle — get redescribed as algebraic manipulations: substitute, factor, solve. Problems that had resisted geometric construction for centuries — including some, like general angle trisection, that Greek geometers' tools genuinely could not solve, a fact only provable *using* this same algebraic machinery two centuries later — suddenly had a common, systematic language in which to be attacked. Two disciplines that had developed almost independently for two thousand years were, after one reframe, doing the same arithmetic underneath different vocabularies. Chapters 04 through 06 rely on exactly this fusion whenever they reach for calculus or linear algebra to solve problems

that started out looking purely geometric or purely combinatorial; none of that machinery has anywhere to stand without Descartes' coordinates underneath it.

TH

นี่คือการเปลี่ยน representation ในความหมายเดียวกับ Euler เพียงแต่เล็งไปที่ทั้งวิชา ไม่ใช่ปริศนาข้อเดียว "สถานะ" ของปัญหาเรขาคณิต — จุด เส้นโค้ง จุดตัด — ถูกอธิบายใหม่เป็นวัตถุเชิงพีชคณิต และ "ตัวดำเนินการ" (operator) — สร้างเส้นนี้ แบ่งครึ่งมุมนั้น — ถูกอธิบายใหม่เป็นการจัดการเชิงพีชคณิต: แทนค่า แยกตัวประกอบ แก้วสมการ ปัญหาที่ด้านการสร้างเชิงเรขาคณิตมาหลายศตวรรษ — รวมถึงบางข้ออย่างการแบ่งมุมใดๆ เป็นสามส่วน ที่เครื่องมือของนักเรขาคณิตกรีกแก้ไม่ได้จริง และพิสูจน์ได้ก็ต่อเมื่อใช้กลไกพีชคณิตชุดเดียวกันนี้อีกสองศตวรรษให้หลัง — จู่ๆ ก็มีภาษากลางที่เป็นระบบให้เข้าใจมิติ สองวิชาที่พัฒนาแทบเป็นอิสระต่อกันมาสองพันปี กลายเป็นวิชาที่ทำเลขชุดเดียวกันได้ศัพท์คนละชุด หลังการเปลี่ยนกรอบครั้งเดียว บทที่ 04 ถึง 06 พังการหลอมรวมนี้ทุกครั้งที่เอื้อมไปคว้าแคลคูลัสหรือพีชคณิตเชิงเส้นมาแก้ปัญหาที่ตั้งต้นในรูปเรขาคณิตล้วนหรือการจัดหมู่ล้วน กลไกเหล่านั้นไม่มีที่ยืนเลย ถ้าไม่มีพิกัดของ Descartes รองอยู่ข้างใต้

The Man Who Turned His Back on the Bar

ชายผู้หันหลังให้คาน

EN

Not every reframe in this book's history comes from a mathematician with a pen. On the evening of October 20, 1968, in Mexico City's Estadio Olímpico Universitario, a twenty-one-year-old American named Dick Fosbury won Olympic gold in the high jump by doing something no serious competitor had ever done at that level: he went over the bar backward, head and shoulders first, arching his spine so his back faced the ground below.

TH

ไม่ใช่ทุกการเปลี่ยนกรอบในประวัติศาสตร์ที่หนังสือเล่มนี้เล่า ที่มาจากนักคณิตศาสตร์กับปากกา คำวันที่ 20 ตุลาคม 1968 ที่ Estadio Olímpico Universitario ในเม็กซิโกซิตี ชาวอเมริกันวัยยี่สิบเอ็ดปีชื่อ Dick Fosbury คว้าเหรียญทองโอลิมปิกกระโดดสูงด้วยสิ่งที่ไม่มีความจริงจังก่อนเคยทำในระดับนั้น: เขาข้ามคานด้วยหลัง หัวและไหล่ขึ้นก่อน แอ่นกระดูกสันหลังจนแผ่นหลังหันลงหาพื้นเบื้องล่าง

EN

Fosbury had not started out trying to reinvent his sport. As a high schooler in Medford, Oregon, following the loss of his younger brother Greg to a drunk-driving accident, he took up track partly to give himself something to hold onto, and struggled badly at the high jump using the standard technique of the era, the "straddle," in which a jumper clears the bar face-down, rotating one leg over first. Using the straddle, Fosbury routinely failed to clear even five feet — a height that barely qualified him for competition (Smithsonian, "The Fosbury Flop"). In the spring of 1963, mid-meet and mid-failure, he changed his body position instinctively rather than deliberately: instead of rotating face-down over the bar, he arched backward over it, landing on his neck and shoulders. It looked, to at least one local newspaper photo caption the following year, like "a fish flopping in a boat" — and the name stuck (Smithsonian, "The Fosbury Flop").

TH

Fosbury ไม่ได้ตั้งใจจะปฏิวัติกีฬาของตัวเองตั้งแต่แรก ตอนเป็นนักเรียนมัธยมที่ Medford รัฐ Oregon หลังเสีย Greg น้องชายไปกับอุบัติเหตุจากคนเมาแล้วขับ เขาหันมาเล่นกรีฑาส่วนหนึ่งเพื่อให้ตัวเองมีอะไรยึดไว้ และกระโดดสูงได้ยอดเยี่ยมด้วยเทคนิคมาตรฐานของยุคนั้น คือท่า "straddle" ที่นักกระโดดข้ามคานโดยคว่ำหน้า หมุนขาข้างหนึ่งข้ามไปก่อน ด้วยท่า straddle Fosbury ข้ามระดับห้าฟุตไม่ผ่านเป็นประจำ — ความสูงที่แทบไม่พอให้ผ่านเกณฑ์ลงแข่ง (Smithsonian, "The Fosbury Flop") ฤดูใบไม้ผลิปี 1963 กลางการแข่งขันและกลางความล้มเหลว เขาเปลี่ยนท่าทางร่างกายด้วยสัญชาตญาณ ไม่ใช่ความจงใจ: แทนที่จะหมุนคว่ำหน้าข้ามคาน เขาแอ่นหลังข้ามไป แล้วลงพื้นด้วยคอและไหล่ คำบรรยายภาพในหนังสือพิมพ์ท้องถิ่นฉบับหนึ่งเมื่อปีถัดมาเห็นว่ามันเหมือน "ปลาดีนอยู่ในเรือ" — และชื่อนั้นก็ติดมาถึงวันนี้ (Smithsonian, "The Fosbury Flop")

EN

The reframe here is worth stating precisely, because it is easy to describe it too loosely as simply "trying something new." The straddle technique accepts a goal implicitly stated as *get your whole body, moving forward and rotating, cleanly past a raised bar* — and treats every improvement as a search within that fixed frame: more speed on the approach, more leg flexibility, a faster rotation. Fosbury's flop instead exploits a genuine fact of physics that the straddle leaves on the table: a jumper's center of mass does not have to clear the bar at the same height as the rest of the body. By arching deeply backward, with the head, arms, and legs all hanging below the arched spine at the peak of the jump, a flopping jumper's *center of mass* can pass *underneath* the bar even while every part of the body technically passes over it — meaning a jumper can, in principle, clear a given bar height with less total energy than a technique whose center of mass has to rise above the bar directly (Stanford, "The Physics of the Fosbury Flop"). This is not a change in effort or talent. It is a change in what "clearing the bar" was allowed to mean — a redefinition of the legal states of the problem that the straddle's own frame had never questioned.

TH

การเปลี่ยนกรอบตรงนี้ต้องพูดให้แน่น เพราะบรรยายหลวมเกินไปได้ง่ายว่าเป็นแค่ "ลองอะไรใหม่ๆ" เทคนิค straddle ยอมรับเป้าหมายที่ระบุไว้โดยนัยว่า เอาทั้งตัวที่เคลื่อนไปข้างหน้าและหมุนอยู่ ผ่านคานที่ยกสูงไปให้สะอาด — แล้วมองการพัฒนาทุกอย่างเป็นการค้นหาภายในกรอบตายตัวนั้น: วิ่งเข้าเร็วขึ้น ขายืดหยุ่นขึ้น หมุนไวขึ้น ทำ flop ของ Fosbury ใช้ข้อเท็จจริงทางฟิสิกส์ที่ straddle วางทิ้งไว้บนโต๊ะ: จุดศูนย์กลางมวลของนักกระโดดไม่จำเป็นต้องข้ามคานที่ความสูงเดียวกับส่วนอื่นของร่างกาย แอ่นหลังลึก ให้หัว แขน และขาห้อยต่ำกว่าแนวสันหลังที่แอ่นตอนถึงจุดสูงสุด จุดศูนย์กลางมวลของนักกระโดดทำ flop ก็ผ่านใต้คานได้ ทั้งที่ทุกส่วนของร่างกายผ่านเหนือคานตามตัวอักษร — ในทางหลักการ นักกระโดดจึงข้ามคานความสูงหนึ่งได้ด้วยพลังงานรวมน้อยกว่าเทคนิคที่จุดศูนย์กลางมวลต้องยกขึ้นเหนือคานตรงๆ (Stanford, "The Physics of the Fosbury Flop") นี่ไม่ใช่การเปลี่ยนความพยายามหรือพรสวรรค์ มันคือการเปลี่ยนว่า "การข้ามคาน" ได้รับอนุญาตให้แปลว่าอะไร — นิยามสถานะที่ถูกกีดกันของปัญหาเสียใหม่ ในจุดที่กรอบของ straddle เองไม่เคยตั้งคำถาม

EN

Crucially, this reframe was only survivable because something in the *environment* had changed first, independent of any high jumper's cleverness. Traditional high-jump landing pits, through most of the twentieth century, were shallow beds of sand or wood shavings — fine for a straddle jumper landing feet or side-first, lethal for someone landing on their head and neck at speed. It was only the spread of deep foam landing mats through American university and high-school track programs during the 1960s — precursors to the "Port-A-Pit" design later commercialized by the inventor Donald Wallace Gordon — that made a backward, head-first landing survivable at all; Fosbury's own high school was among the earlier adopters of such matting, after his coaches, alarmed at how he was landing, arranged for the school to receive foam retired from a local university's upgraded pits (Smithsonian, "The Fosbury Flop"). Change the representation of the jump without changing the landing surface, and the same idea is simply a fast way to break your neck. This is a detail worth sitting with: sometimes the operators available to you — the legal moves your environment permits —

are not fixed facts of nature but recent, contingent, and changeable, and part of finding a shorter path is asking whether an operator you have been treating as fixed is actually just a habit someone else could unfix.

TH

การเปลี่ยนกรอบครั้งนี้รอดมาได้เพราะบางอย่างในสภาพแวดล้อมเปลี่ยนไปก่อน โดยไม่เกี่ยวกับความฉลาดของนักกระโดดสูงคนไหนเลย นี่คือจุดสำคัญ หลุมลงพื้นของกระโดดสูงตามธรรมเนียม ตลอดเกือบทั้งศตวรรษที่ยี่สิบ คือแอ่งทรายหรือซึกบัต้นๆ — พอใช้สำหรับนักกระโดดทำ straddle ที่ลงด้วยเท้าหรือข้างลำตัว แต่ถึงตายสำหรับคนที่ลงด้วยหัวและคอด้วยความเร็ว เบาะโฟมหนาที่แพร่เข้าไปในโครงการกรีฑาของมหาวิทยาลัยและโรงเรียนมัธยมอเมริกันช่วงทศวรรษ 1960 — ต้นแบบของเบาะ "Port-A-Pit" ที่นักประดิษฐ์ Donald Wallace Gordon ทำขายในภายหลัง — คือสิ่งเดียวที่ทำให้การลงพื้นแบบหันหลังเอาหัวลงรอดตายได้ โรงเรียนมัธยมของ Fosbury เองเป็นหนึ่งในที่ที่รับเบาะแบบนี้มาใช้ค่อนข้างเร็ว โค้ชของเขาตกใจกับท่าที่เขาลงพื้น จึงจัดการให้โรงเรียนได้รับโฟมที่มหาวิทยาลัยแห่งหนึ่งในท้องถิ่นปลดจากหลุมที่อัปเกรดแล้ว (Smithsonian, "The Fosbury Flop") เปลี่ยน representation ของการกระโดดโดยไม่เปลี่ยนพื้นที่ลง ไอเดียเดียวกันก็เป็นแค่วิธีหักคอตัวเองอย่างรวดเร็ว รายละเอียดนี้ควรนั่งอยู่กับมันสักพัก: บางที operator ที่มีให้ใช้ — ท่าที่ถูกกติกาซึ่งสภาพแวดล้อมอนุญาต — ไม่ใช่ข้อเท็จจริงตายตัวของธรรมชาติ แต่เพิ่งเกิด ขึ้นกับเงื่อนไข และเปลี่ยนได้ ส่วนหนึ่งของการหาเส้นทางที่สั้นกว่าคือถามว่า operator ที่ถูกมองเป็นของตายตัวมาตลอดนั้น แท้จริงเป็นแค่ความเคยชินที่คนอื่นแก้ได้หรือเปล่า

EN

Fosbury struggled even after his 1963 breakthrough — his own college coach at Oregon State initially pushed him back toward the straddle, worried the flop could not be coached to consistency, and Fosbury nearly lost his scholarship over inconsistent results before a breakthrough 1968 season culminating in a Pac-8 title, an NCAA championship, and finally Mexico City, where he cleared 2.24 meters for Olympic gold and an Olympic record, using a technique he was still, at the time, the only competitor in the field employing (World Athletics, "50 years since the day Dick Fosbury revolutionised the high jump"). He never jumped higher than that night. He

didn't need to, for the argument this book is making: within four years, at the 1972 Munich Olympics, 28 of the 40 competitors in the men's high jump final were using some version of his technique; by 1980, the straddle had essentially vanished from elite competition (World Athletics; Smithsonian, "The Fosbury Flop"). A field of specialists spent decades searching harder within one representation of "how to jump." One outsider changed the representation, once, and the entire field's search space was permanently reshaped around his answer.

T H

Fosbury ยังลำบากต่อแม้หลังการทะลุทะลวงปี 1963 — โค้ชมหาวิทยาลัยของเขาเองที่ Oregon State ตอนแรกดันเขากลับไปหาทำ straddle เพราะกังวลว่า flop สอนให้สม่ำเสมอไม่ได้ Fosbury เกือบเสียทุนการศึกษาเพราะผลงานไม่คงเส้นคงวา ก่อนฤดูกาล 1968 ที่ทะลุทะลวงจะปิดด้วยแชมป์ Pac-8 แชมป์ NCAA และในที่สุดคือเม็กซีโกซิตี ที่เขาข้าม 2.24 เมตรคว้าเหรียญทองโอลิมปิกพร้อมสถิติโอลิมปิก ด้วยเทคนิคที่ในเวลานั้นเขายังเป็นคู่แข่งคนเดียวในสนามที่ใช้ (World Athletics, "50 years since the day Dick Fosbury revolutionised the high jump") เขาไม่เคยกระโดดสูงกว่าคืบนั้นอีก และไม่จำเป็น สำหรับข้อเสนอที่หนังสือเล่มนี้วางไว้: ภายในสี่ปี ที่โอลิมปิก Munich ปี 1972 นักกีฬา 28 คนจาก 40 คนในรอบชิงกระโดดสูงชายใช้เทคนิคของเขาในรูปแบบใดรูปแบบหนึ่ง พอถึงปี 1980 ทำ straddle ก็หายจากการแข่งขันระดับสูงแทบสิ้นเชิง (World Athletics; Smithsonian, "The Fosbury Flop") วงการผู้เชี่ยวชาญใช้เวลาหลายทศวรรษค้นให้หนักขึ้นภายใน representation เดียวของ "จะกระโดดอย่างไร" คนนอกคนหนึ่งเปลี่ยน representation ครั้งเดียว พื้นที่ค้นหาของทั้งวงการก็เปลี่ยนรูปถาวรรอบคำตอบของเขา

Why Reframes Feel Like Cheating

ทำไมการเปลี่ยนกรอบถึงรู้สึกเหมือนการโกง

E N

There is a common thread running underneath Euler's letters, the Gauss legend, Descartes' coordinates, and Fosbury's back-first leap, and it is worth naming directly because it will recur for the rest of this book: a genuine

reframe very often does not feel, to the people watching it or even to the person performing it, like "real" work in the domain being reframed. Euler said so explicitly, in writing, about his own solution — it "bears little relationship to mathematics." Fosbury's own world-record-holding rival, Valery Brumel, was quoted at the time calling the flop "an aberration," not a technical advance. A reframe often looks less like mastering a domain's existing difficulty and more like refusing to engage with it on its own terms — which is, in a sense, exactly what it is. The difficulty a domain's practitioners have spent years learning to fight is precisely the difficulty that a good reframe makes irrelevant, and there is no reason the practitioners who mastered fighting it should be the first to recognize that the fight was optional.

TH

การเปลี่ยนกรอบที่แท้จริงมักไม่รู้สึกละเอียดเหมือนงาน "จริง" ในสาขาที่ถูกเปลี่ยนกรอบ ทั้งในสายตาคณดูและแม้แต่คนที่ทำเอง นี่คือการเปลี่ยนกรอบที่ลึกลับของ Euler ดำเนิน Gauss พิกัดของ Descartes และการทะยานเออาหลังขึ้นก่อนของ Fosbury และควรเรียกชื่อมันจริงๆ เพราะมันจะกลับมาซ้ำตลอดหนังสือเล่มนี้ Euler พุดออกมาเองเป็นลายลักษณ์อักษรถึงคำตอบของตัวเอง — มัน "แทบไม่เกี่ยวข้องกับคณิตศาสตร์" Valery Brumel คู่แข่งเจ้าของสถิติโลกของ Fosbury เอง ถูกอ้างคำพูดในเวลานั้นว่า flop คือ "ความผิดปกติ" ไม่ใช่ความก้าวหน้าเชิงเทคนิค การเปลี่ยนกรอบมักไม่ดูเหมือนการเอาชนะความยากที่มีอยู่ของสาขา แต่ดูเหมือนการปฏิเสธจะสู้กับมันตามกติกาของมันเอง — ซึ่งในความหมายหนึ่งก็ใช่ ความยากที่ผู้ปฏิบัติในสาขาใช้เวลาหลายปีเรียนรู้ที่จะสู้ คือความยากที่การเปลี่ยนกรอบที่ดีทำให้ไม่เกี่ยวข้องอีกต่อไปพอดี และไม่มีเหตุผลที่ผู้ปฏิบัติซึ่งเชี่ยวชาญการสู้จะต้องเป็นคนแรกที่มองออกว่าการสู้นั้นไม่จำเป็นตั้งแต่แรก

EN

This is why the theoretical apparatus matters and is not mere decoration. A representation is a specific, nameable thing — a choice of states, a choice of operators, a choice of goal — and asking "what representation am I using right now, and did I choose it or did I inherit it?" is a concrete, repeatable question, not a vague call to "think outside the box." Euler's states were landmasses and bridge-crossings, not streets. Gauss's states were pairs, not a sequence. Descartes' states were coordinate pairs, not geometric figures.

Fosbury's states were centers of mass, not bodies treated as rigid wholes facing a fixed direction. In every case, the difficulty of the original problem turns out to have been at least partly an artifact of an unexamined representation — and in every case, once the new representation is stated, the remaining work is comparatively short. That gap — vast search before the reframe, short walk after it — is the shortest path this book's title promises, and chapter 03 turns to the deliberate, teachable techniques several traditions have since built for finding that reframe on purpose, rather than waiting for a Euler, a Gauss, or a Fosbury to stumble into it.

TH

ชุดเครื่องมือทางทฤษฎีสำคัญด้วยเหตุนี้ ไม่ใช่ของระดับ representation เป็นของที่เจาะจงและเรียกชื่อได้ — การเลือกสถานะ การเลือก operator การเลือกเป้าหมาย — และคำถามว่า "ตอนนี้ใช้ representation อะไรอยู่ และเลือกมันเองหรือรับมรดกมา" เป็นคำถามรูปธรรมที่ถามซ้ำได้ ไม่ใช่คำเรียกร้องลอยๆ ให้ "คิดนอกกรอบ" สถานะของ Euler คือผืนดินกับการข้ามสะพาน ไม่ใช่ถนน สถานะของ Gauss คือคู่มือไม่ใช่ลำดับ สถานะของ Descartes คือคู่มือพิถีพิถัน ไม่ใช่รูปเรขาคณิต สถานะของ Fosbury คือจุดศูนย์กลางมวล ไม่ใช่ร่างกายที่มองเป็นก้อนแข็งก้อนเดียวหันไปทางตายตัว ทุกกรณี ความยากของปัญหาตั้งต้นกลายเป็นผลพลอยของ representation ที่ไม่เคยตรวจสอบ อย่างน้อยก็บางส่วน — และทุกกรณี พอ representation ใหม่แจ่มออกมา งานที่เหลือก็สั้นเมื่อเทียบกับ ช่องว่างนั้น — ค้นหาหาศาลก่อนเปลี่ยนกรอบ เดินสั้นๆ หลังจากนั้น — คือเส้นทางที่สั้นที่สุดที่ซื้อหนังสือเล่มนี้สัญญาไว้ บทที่ 03 จะหันไปหาเทคนิคที่จูงใจและสอนกันได้ ซึ่งหลายธรรมเนียมสร้างขึ้นตั้งแต่นั้น เพื่อหาการเปลี่ยนกรอบอย่างตั้งใจ แทนที่จะรอให้ Euler, Gauss หรือ Fosbury สักคนเดินสะดุดเข้าไปเจอ

The Move

ทำประจำบท

EN

Move 02 — Change the representation until the difficulty dissolves.

TH

Move 02 — เปลี่ยน representation จนความยากละลาย

EN

- **Ask what you are allowed to throw away.** Euler discarded every street in Königsberg and kept only which bridge touched which landmass. Before searching harder inside a hard representation, ask which of its details the goal actually depends on — and prove, don't just assume, that the rest is safe to discard.
- **Ask what you are needlessly holding apart.** Gauss's pairing trick works because addition doesn't care about order; the "sequence" was a habit of representation, not a requirement of the arithmetic. Look for structure your current framing is hiding by insisting on an arrangement — order, direction, sequence — the goal never actually required.
- **Check whether the environment's rules are fixed or just unquestioned.** Fosbury's reframe of the jump was only viable once foam mats replaced sand pits — an external operator, not a mental one, had to change first. When a reframe seems physically or practically blocked, ask whether the blocking constraint is a law of nature or a fixable habit of your environment.
- **Expect it to feel illegitimate.** Euler called his own breakthrough barely mathematics. If your reframe doesn't provoke at least a flicker of "that's not really solving the problem" from someone who knows the domain well, it may not be cutting deep enough yet.

TH

- **ถามว่าอนุญาตให้โยนอะไรทิ้งได้บ้าง** Euler ทิ้งถนนทุกเส้นใน Königsberg และเก็บไว้แค่สะพานไหนและพื้นดินไหน ก่อนจะค้นหาให้หนักขึ้นภายใน representation ที่ยาก ให้ถามว่ารายละเอียดอันไหนของมันที่เป้าหมายขึ้นอยู่กับมันจริงๆ — แล้วพิสูจน์ อย่าแค่สมมติ ว่าที่เหลือทิ้งได้อย่างปลอดภัย
- **ถามว่ากำลังแยกอะไรออกจากกันโดยไม่จำเป็น** เลห์จับคู่ของ Gauss ใช้ได้เพราะการบวกไม่สนใจลำดับ "ลำดับ" เป็นความเคยชินของ representation ไม่ใช่ข้อเรียกร้องของเลขคณิต มองหาโครงสร้างที่กรอบปัจจุบันซ่อนไว้ด้วยการย่นกราน เรื่องการจัดวาง — ลำดับ ทิศทาง การเรียง — ที่เป้าหมายไม่เคยต้องการจริงๆ
- **ตรวจว่ากฎของสภาพแวดล้อมตายตัวจริง หรือแค่ไม่มีใครตั้งคำถาม** การเปลี่ยนกรอบการกระโดดของ Fosbury เป็นไปได้ก็ต่อเมื่อเบาะโฟมมาแทนแอ่งทรายแล้ว

— operator ภายนอก ไม่ใช่ในหัว ที่ต้องเปลี่ยนก่อน เมื่อการเปลี่ยนกรอบดูเหมือนติดขัดในทางกายภาพหรือทางปฏิบัติ ให้ถามว่าข้อจำกัดที่ขวางอยู่นั้นเป็นกฎธรรมชาติ หรือเป็นความเคยชินของสภาพแวดล้อมที่แก้ได้

- **เตรียมใจว่ามันจะรู้สึกไม่ชอบธรรม** Euler เรียกการทะลุทะลวงของตัวเองว่าแทบไม่ใช่คณิตศาสตร์ ถ้าการเปลี่ยนกรอบไม่ได้กระตุ้นความรู้สึก "นั่นไม่ใช่การแก้ปัญหาล่ะ" นี่" ออกมาจากคนที่รู้จักสาขานั้นดี แม้แต่ুবเดียว มันอาจยังตัดไม่ลึกพอ

EN

Next — Chapter 03 · The Heuristic Age: what happens once solvers stop waiting for a reframe to arrive by accident and start building deliberate, teachable methods for finding one on purpose.

TH

ต่อไป — บทที่ 03 · ยุค heuristic: จะเกิดอะไรขึ้นเมื่อนักแก้ปัญหาเลิกรอให้การเปลี่ยนกรอบมาถึงโดยบังเอิญ แล้วเริ่มสร้างวิธีที่ตั้งใจและสอนกันได้ขึ้นมาหามันอย่างตั้งใจ

CHAPTER 03

The Heuristic Age

ยุค heuristic

Coutances, June 1880

Coutances มิถุนายน 1880

EN

For a fortnight, Henri Poincaré had been failing on purpose. Twenty-six years old and already restless with talent, he sat at his desk in Caen for an hour or two each day trying to prove a negative: that a certain kind of function — later named, in his honor and against his protest, Fuchsian functions — could not exist. He tried combination after combination. Nothing held. Then one evening he broke his own routine and drank black coffee before bed. He could not sleep. "Ideas rose in crowds," he wrote; he could feel them colliding, pairing off, until they locked into stable combinations. By morning he had done the opposite of what he set out to do: he had established that a whole class of these functions *did* exist, built from the hypergeometric series. He needed only a few hours to write out the verification (Poincaré, "Mathematical Creation," 1910).

TH

Henri Poincaré ล้มเหลวโดยตั้งใจมาสองสัปดาห์เต็ม เขาอายุยี่สิบหก พรรสวรค์ล้น จนนั่งไม่ติด และนั่งที่โต๊ะทำงานใน Caen วันละหนึ่งถึงสองชั่วโมงเพื่อพิสูจน์ข้อปฏิเสธข้อหนึ่ง: ฟังก์ชันชนิดหนึ่ง — ภายหลังได้ชื่อว่า Fuchsian functions เพื่อเป็นเกียรติแก่เขาและทั้งที่เขาทักท้วง — มีอยู่ไม่ได้ เขาลองจับคู่แล้วจับคู่เล่า ไม่มีอะไรอยู่ตัว แล้วเย็นวันหนึ่งเขาฉีกกฎวัตรของตัวเอง ดื่มกาแฟดำก่อนนอน แล้วนอนไม่หลับ "ความคิดผุดขึ้นมาเป็นฝูง" ("Ideas rose in crowds") เขาเขียนไว้; เขารู้สึกได้ว่ามันชนกัน จับคู่กัน จนลือกเข้าเป็นชุดที่มั่นคง พอถึงเช้าเขาทำสิ่งตรงข้ามกับที่ตั้งใจสำเร็จ: เขาแสดงได้ว่าฟังก์ชันพวกนี้ทั้งชั้นมีอยู่จริง สร้างจากอนุกรมไฮเพอร์จีโอ

เมตริก (hypergeometric series) การเขียนบทตรวจสอบใช้เวลาแค่ไม่กี่ชั่วโมง (Poincaré, "Mathematical Creation," 1910)

EN

That should have been the strange part of the story. It was not. A few weeks later, Poincaré left Caen on a geology field trip for the École des Mines, glad to think about rock strata instead of differential equations. At Coutances the party changed to a horse-drawn omnibus. "At the moment when I put my foot on the step," he recalled, the idea arrived, unbidden and complete: the transformations he had used to define his new functions were *identical* to the ones he had encountered, in an entirely different investigation, in non-Euclidean geometry. He did not check it on the spot — he had a conversation to get back to — but he felt, in his own word, a "perfect certainty." He verified it at leisure once he was home (Poincaré, "Mathematical Creation," 1910; the fullest secondhand retelling is in Shaw, "Henri Poincaré as an Investigator," *Popular Science Monthly*, 1913).

TH

ส่วนที่แปลกของเรื่องน่าจะจบตรงนั้น แต่ไม่จบ ไม่ก็สัปดาห์ต่อมา Poincaré ออกจาก Caen ไปทัศนศึกษาธรณีวิทยาให้ École des Mines และดีใจที่จะได้คิดเรื่องชั้นหิน แทนสมการเชิงอนุพันธ์ ที่ Coutances ขณะเปลี่ยนไปขึ้นรถโดยสารเทียมม้า "ในวินาทีที่วางเท้าลงบนขั้นบันได" เขาเล่า ความคิดก็มาถึง ไม่มีใครเรียก และมาครบทั้งก่อน: การแปลงที่เขาใช้นิยามฟังก์ชันใหม่ของตัวเองเหมือนกันทุกประการกับการแปลงที่เขาเคยพบในการค้นคว้าคนละเรื่องโดยสิ้นเชิง — ในเรขาคณิตนอกแบบยุคลิด เขาไม่ได้ตรวจสอบตรงนั้น — เขามีบทสนทนาค้างอยู่ที่ต้องกลับไปคุยต่อ — แต่เขารู้สึกถึงสิ่งที่เขาเรียกเองว่า "ความแน่ใจอย่างสมบูรณ์" ("perfect certainty") กลับถึงบ้านแล้วเขาจึงตรวจสอบตามสบาย (Poincaré, "Mathematical Creation," 1910; การเล่าต่อมือสองที่ครบที่สุดอยู่ที่ Shaw, "Henri Poincaré as an Investigator," *Popular Science Monthly*, 1913)

EN

Nothing about mathematics required him to be riding a bus. He had not been thinking about Fuchsian functions in the previous hour, or so he believed. And yet the answer to a problem he had spent months on arrived in

the second it took to lift one foot. Poincaré is the first major scientist to take that fact seriously rather than treat it as a charming anecdote for after-dinner speeches. If the mind can solve a problem while, by its owner's own testimony, not working on it, then whatever "solving" is, it is not simply conscious search performed faster. Something else is going on beneath the desk, and it was time somebody studied it the way one would study any other natural process — by collecting cases, naming the stages, and looking for the pattern that repeats.

TH

คณิตศาสตร์ไม่ได้บังคับให้เขาต้องนั่งรถโดยสารอยู่ ชั่วโมงก่อนหน้านั้นเขาไม่ได้คิดถึง Fuchsian functions เลย — หรืออย่างน้อยเขาก็เชื่ออย่างนั้น กระนั้นคำตอบของ โจทย์ที่เขาทุ่มเวลาให้หลายเดือนก็มาถึงในเสี้ยววินาทีที่ไฉยยกเท้าข้างเดียว Poincaré คือนักวิทยาศาสตร์คนสำคัญคนแรกที่จับข้อเท็จจริงนี้มาคิดจริงจัง แทนที่จะปล่อยให้ เป็นเกร็ดน่ารู้ไฉไลหลังมื้อค่ำ ถ้าจิตแก้โจทย์ได้ทั้งที่เจ้าของจิตยืนยันเองว่าไม่ได้ ทำงานกับมัน "การแก้" ไม่ว่าจะเป็นอะไร ก็ไม่ใช่แค่การค้นหาแบบรู้ตัวที่ทำได้เร็วขึ้น มีอย่างอื่นเกิดขึ้นใต้โต๊ะทำงาน และถึงเวลาที่ใครสักคนจะศึกษามันแบบเดียวกับที่ ศึกษากระบวนการทางธรรมชาติอื่น — เก็บกรณี ตั้งชื่อขั้นตอน แล้วมองหารูปแบบที่ ซ้ำเดิม

Four Stages, Named

ตั้งชื่อให้สี่ขั้นตอน

EN

That naming job fell to Graham Wallas, a British social psychologist who had never proved a theorem in his life but had read Poincaré and Hermann von Helmholtz — another scientist who had left notes on the odd, sourceless arrival of his own good ideas — closely enough to see the shape common to both. In his 1926 book *The Art of Thought*, Wallas proposed that creative problem-solving runs through four distinguishable stages: **preparation**, in which the problem is consciously studied and the relevant facts assembled; **incubation**, a period — sometimes minutes, sometimes years — in which the problem is set aside and, apparently, worked on without supervision; **illumination**, the sudden arrival of the idea, often at a moment (a bus stop, a

half-sleep, a walk) that has nothing to do with the problem's content; and **verification**, the deliberate, effortful checking of what illumination handed over, because illumination is a hunch-generator, not a proof-generator (Wallas, *The Art of Thought*, 1926; overview in Wikipedia, "Graham Wallas").

TH

งานตั้งชื่อตกเป็นของ Graham Wallas นักจิตวิทยาสังคมชาวอังกฤษที่ไม่เคยพิสูจน์ทฤษฎีบทสักข้อในชีวิต แต่เขาอ่าน Poincaré และ Hermann von Helmholtz — นักวิทยาศาสตร์อีกคนที่ทั้งบันทึกไว้ว่าความคิดดีๆ ของตัวเองมาถึงอย่างประหลาดและไร้ที่มา — ละเอียดพอจะเห็นรูปทรงที่ทั้งสองคนมีร่วมกัน ในหนังสือปี 1926 ชื่อ *The Art of Thought* Wallas เสนอว่าการแก้ปัญหาเชิงสร้างสรรค์เดินผ่านสี่ขั้นที่แยกออกจากกันได้: **การเตรียม (preparation)** ขั้นที่ศึกษาโจทย์อย่างรู้ตัวและรวบรวมข้อเท็จจริงที่เกี่ยวข้อง; **การบ่มความคิด (incubation)** ช่วงเวลา — บางครั้งไม่กี่นาที บางครั้งหลายปี — ที่โจทย์ถูกวางลง แล้วดูเหมือนมีอะไรทำงานต่อโดยไม่มีใครควบคุม; **การสว่างวาบ (illumination)** ความคิดมาถึงฉับพลัน มักในจังหวะ (ขั้น) บันไดรถไฟโดยสาร ครั้งหลังครั้งต้น การเดินเล่น) ที่ไม่เกี่ยวอะไรกับเนื้อหาของโจทย์เลย; และ **การตรวจสอบ (verification)** ใส่ตรวจสอบสิ่งที่ illumination ยื่นมาให้อย่างตั้งใจและออกแรง เพราะ illumination เป็นเครื่องผลิตทางสังหรณ์ ไม่ใช่เครื่องผลิตบทพิสูจน์ (Wallas, *The Art of Thought*, 1926; ภาพรวมที่ Wikipedia, "Graham Wallas")

EN

Notice what this model does and does not claim. It does not say incubation is magic, and it does not say illumination can be summoned on demand. It says that the four stages are functionally distinct — that a mind engaged in incubation is doing something different from a mind engaged in preparation, even though the thinker experiences the first as "doing nothing." Wallas had, in effect, taken Poincaré's private confession and turned it into a public, testable claim about how *any* mind solves *any* hard problem, whether the problem is a theorem, a poem, or a stuck hinge. It was the first attempt at a science of the solver's own head — not of the problem, but of the person facing it.

TH

แบบจำลองนี้อ้างน้อยกว่าที่ฟังดู มันไม่ได้บอกว่า incubation คือเวทมนตร์ และไม่ได้บอกว่าเรียก illumination มาตามสั่งได้ มันบอกว่าสี่ขั้นนี้ต่างกันในเชิงหน้าที่ — จิตที่อยู่ใน incubation ทำอะไรบางอย่างที่ต่างจากจิตที่อยู่ในการเตรียม แม้คนคิดจะรู้สึกว่าจะอย่างแรกคือ "การไม่ทำอะไรเลย" Wallas เอาคำสารภาพส่วนตัวของ Poincaré มาแปลงเป็นข้อกล่าวอ้างสาธารณะที่ทดสอบได้ ว่าจิตใดก็ตามแก้โจทย์ยากข้อใดก็ตามอย่างไร ไม่ว่าโจทย์นั้นจะเป็นทฤษฎีบท บทกวี หรือบานพับที่ติดขัด นี่คือการพยายามครั้งแรกที่จะทำศาสตร์ว่าด้วยหัวของนักแก้ปัญหาเอง — ไม่ใช่ศาสตร์ว่าด้วยตัวโจทย์ แต่ว่าด้วยคนที่ยืนอยู่ตรงหน้าโจทย์

EN

The claim needed more than two data points, however brilliant those points were. It got them from a mathematician with a habit even more useful than genius: Jacques Hadamard kept a filing cabinet's worth of curiosity about his own mental habits, and when he could not satisfy it alone, he wrote letters. His 1945 book *The Psychology of Invention in the Mathematical Field* is built from decades of his own introspection cross-checked against reports he solicited from working mathematicians and physicists, George Pólya and Albert Einstein among them (Hadamard, *An Essay on the Psychology of Invention in the Mathematical Field*, 1945). Hadamard confirmed Wallas's four stages independently and added a detail that startled his correspondents when they compared notes: several of them, Einstein included, reported that their best mathematical thinking did not use words at all. It

arrived as a felt sense of relations between objects — visual, muscular, almost architectural — which then had to be translated into ordinary language and symbols afterward, sometimes with real effort and loss. Preparation and verification are the parts of discovery that happen in a language you could dictate to a stenographer. Incubation and illumination, on Hadamard's evidence, mostly do not.

TH

จุดข้อมูลสองจุดไม่พอคำข้อกล่าวอ้างนี้ ต่อให้สองจุดนั้นเปล่งประกายแค่ไหน จุดที่เหลือมาจากนักคณิตศาสตร์ที่มีนิสัยซึ่งมีประโยชน์ยิ่งกว่าอัจฉริยภาพ: Jacques Hadamard สะสมความสงสัยเรื่องนิสัยทางความคิดของตัวเองไว้ได้เต็มตู้เอกสาร และเมื่อสนองมันคนเดียวไม่ไหว เขาก็เขียนจดหมาย หนังสือปี 1945 ของเขา *The Psychology of Invention in the Mathematical Field* สร้างจากการสำรวจตัวเอง นานหลายทศวรรษ เทียบกับรายงานที่เขาขอมาจากนักคณิตศาสตร์และนักฟิสิกส์ที่ทำงานจริง ในนั้นมี George Pólya และ Albert Einstein ด้วย (Hadamard, *An Essay on the Psychology of Invention in the Mathematical Field*, 1945) Hadamard ยืนยันสี่ขั้นของ Wallas ได้อย่างเป็นอิสระ และเพิ่มรายละเอียดหนึ่งที่ทำให้ผู้ตอบจดหมายตกใจตอนเอามาเทียบกัน: หลายคน รวมทั้ง Einstein รายงานว่าความคิดทางคณิตศาสตร์ที่ดีที่สุดของตนไม่ได้ใช้ถ้อยคำเลย มันมาถึงเป็นความรู้สึกถึงความสัมพันธ์ระหว่างวัตถุ — เป็นภาพ เป็นกลัมนื้อ เกือบจะเป็นสถาปัตยกรรม — แล้วต้องแปลเป็นภาษาและสัญลักษณ์ธรรมดาที่หลัง บางครั้งต้องออกแรงจริงและสูญเสียบางอย่างไป การเตรียมกับการตรวจสอบคือส่วนของการค้นพบที่เกิดในภาษาที่บอกให้เสมือนจดตามได้ incubation กับ illumination ตามหลักฐานของ Hadamard ส่วนใหญ่ไม่ใช่

EN

This is not the whole story of what this book will call reframing — Wallas and Hadamard describe a process happening to the solver, not a deliberate choice the solver makes. But it is the first hard evidence that incubation is real, that a mind can restructure a problem's representation without being asked to and without the owner watching it happen. Chapters ahead will show reframing being done on purpose, as a discipline. This chapter is about what a mind does when reframing happens to it — and about the tools people built once they noticed the pattern and tried to make it happen more often.

TH

นี่ยังไม่ใช่เรื่องทั้งหมดของสิ่งที่หนังสือเล่มนี้จะเรียกว่าการเปลี่ยนกรอบ (reframe) — Wallas กับ Hadamard บรรยายกระบวนการที่เกิดขึ้นกับนักแก้ปัญหา ไม่ใช่ทางเลือกที่นักแก้ปัญหาดังใจเลือก แต่มันคือหลักฐานแข็งขันเร็วกว่า incubation มีจริง จิตปรับโครงสร้างรูปแบบการแทนปัญหา (representation) ของโจทย์ใหม่ได้เอง ไม่มีใครสั่งและเจ้าของไม่ได้เฝ้าดู บทข้างหน้าจะแสดงการเปลี่ยนกรอบที่ทำอย่างจงใจ ในฐานะวินัยอย่างหนึ่ง บทนี้ว่าด้วยสิ่งที่จิตทำเมื่อการเปลี่ยนกรอบเกิดขึ้นกับมัน — และว่าด้วยเครื่องมือที่คนสร้างขึ้นเมื่อเห็นรูปแบบนี้แล้วพยายามทำให้มันเกิดบ่อยขึ้น

What Apes and Candles Taught the Gestaltists

สิ่งที่ลึกลับที่เรียนสอนนักจิตวิทยาเกสตัลท์

EN

While Poincaré's confessions were circulating among mathematicians, an entirely separate community was reaching the same conclusion from the opposite direction: not by asking brilliant humans to introspect, but by watching animals fail and then, suddenly, not fail.

TH

ระหว่างที่คำสารภาพของ Poincaré หมุนเวียนอยู่ในหมู่นักคณิตศาสตร์ ชุมชนอีกกลุ่มที่แยกกันโดยสิ้นเชิงกำลังไปถึงข้อสรุปเดียวกันจากทิศตรงข้าม: ไม่ใช่ด้วยการขอให้มนุษย์หัดสำรวจใจตัวเอง แต่ด้วยการเฝ้าดูสัตว์ทำไม่สำเร็จ แล้วอยู่ๆ ก็สำเร็จ

EN

In 1913, the Prussian Academy of Sciences opened an anthropoid research station on Tenerife, in the Canary Islands, and installed as its director a young Gestalt psychologist named Wolfgang Köhler (Wikipedia, "Wolfgang Köhler"). Over the next several years Köhler set his chimpanzees problems that had a food reward just out of reach — bananas outside the bars, requiring a stick to pull them closer; bananas hanging from the ceiling, requiring two crates stacked to make a step. The prevailing theory of animal learning at the time, championed by Edward Thorndike, held that all such learning was trial-and-error: the animal flails, something works by accident, and the successful action is stamped in by reward. Köhler's chimpanzees did something that did not fit that story. They would sit — visibly, almost theatrically, to Köhler's eye — doing nothing, surveying the crates and the bananas, and then rise and execute the entire stacking sequence in one continuous, confident motion, with no visible trial before the error-free success. Köhler called this **insight**, and published his findings in 1917 as *Intelligenzprüfungen an Menschenaffen*, translated into English in 1925 as *The Mentality of Apes* (Wikipedia, "Wolfgang Köhler"). Whatever this was, it looked much less like accumulated luck and much more like a sudden reorganization of the animal's picture of the situation — the crate stopped being "a thing chimpanzees sit near" and became "a thing chimpanzees stand on."

TH

ปี 1913 Prussian Academy of Sciences เปิดสถานีวิจัยลิงเอบที่ Tenerife ในหมู่เกาะคานารี และตั้งนักจิตวิทยาเกสตัลท์ (Gestalt) หนุ่มชื่อ Wolfgang Köhler เป็นผู้อำนวยการ (Wikipedia, "Wolfgang Köhler") หลายปีถัดมา Köhler ตั้งโจทย์ให้ชิมแปนซีของเขาโดยวางรางวัลอาหารไว้พ้นมือเอื้อมนิดเดียว — กล้วยอยู่นอกซี่กรง ต้องใช้ไม้เขี่ยเข้ามา; กล้วยห้อยจากเพดาน ต้องซ้อนลังสองใบขึ้นไปเหยียบ ทฤษฎีการเรียนรู้ของสัตว์ที่ครองสนามอยู่ตอนนั้น มี Edward Thorndike เป็นหัวหอก ถือว่าการเรียนรู้แบบนี้ทั้งหมดคือลองผิดลองถูก: สัตว์เดินไปเรื่อย บางอย่างบังเอิญได้ผล แล้วรางวัลก็ตกท่าที่สำเร็จนั้นลงไป ชิมแปนซีของ Köhler ทำอะไรบางอย่างที่ไม่เข้ากับเรื่องเล่านั้น พวกมันนั่ง — เห็นได้ชัด เกือบจะเหมือนแสดงละคร ในสายตาของ Köhler — โดยไม่ทำอะไร กวาดตาดูลังกับกล้วย แล้วลุกขึ้นซ้อนลังทั้งชุดในการเคลื่อนไหวเดียวที่ต่อเนื่องและมั่นใจ ไม่มีการลองผิดที่มองเห็นได้ก่อนความสำเร็จที่

ไร้ข้อผิดพลาดนั้น Köhler เรียกสิ่งนี้ว่า **การหยั่งเห็น (insight)** และตีพิมพ์ผลการศึกษาในปี 1917 ในชื่อ *Intelligenzprüfungen an Menschenaffen* แปลเป็นภาษาอังกฤษปี 1925 ในชื่อ *The Mentality of Apes* (Wikipedia, "Wolfgang Köhler") ไม่ว่ามันจะเป็นอะไร มันดูไม่เหมือนโชคที่สะสมมา แต่เหมือนภาพของสถานการณ์ในหัวสัตว์ตัวนั้นถูกจัดระเบียบใหม่ในพริบตา — ลังเล็กเป็น "สิ่งที่чимแปนชิ้นๆ" แล้วกลายเป็น "สิ่งที่чимแปนชียีนบน"

EN

Köhler's student Karl Duncker pushed the same question into the human laboratory and gave it a name that survives every introductory psychology course: **functional fixedness**. In the task now known as the candle problem, a subject is handed a candle, a box of thumbtacks, and a book of matches, and asked to mount the lit candle on a wall so no wax drips on the table below. Most people try to tack the candle straight to the wall, or melt wax onto the wall as glue — both fail. Very few think to empty the tack box, tack the box to the wall, and stand the candle inside it, because the box has already been cast, before the subject even starts reasoning, in the role of "container for tacks." When Duncker instead handed subjects the tacks loose, next to an empty box, the solution rate rose sharply — the box, no longer costumed as a container, became visible as a shelf (Duncker, "On Problem Solving," *Psychological Monographs*, 1945; Wikipedia, "Candle problem"). Functional fixedness is the mind pre-loading a representation — assigning an object one role and refusing, without help, to try another — and it is the same failure that trapped every petitioner at Gordium: a state space silently narrowed by habit rather than by the actual rules of the problem.

TH

Karl Duncker ลูกศิษย์ของ Köhler ผลักคำถามเดียวกันเข้าห้องทดลองกับมนุษย์ และตั้งชื่อให้มันจนอยู่รอดในวิชาจิตวิทยาเบื้องต้นทุกคอร์ส: **ความยึดติดในหน้าที่ (functional fixedness)** ในภารกิจที่ตอนนี้รู้จักกันในชื่อโจทย์เทียนไข ผู้ถูกทดลองได้รับเทียนหนึ่งเล่ม กล่องหมุดกดหนึ่งกล่อง และไม้ขีดหนึ่งกล่อง แล้วถูกขอให้ติดเทียนที่จุดแล้วไว้บนผนังโดยไม่ให้ขี้ผึ้งหยดลงโต๊ะข้างล่าง คนส่วนใหญ่พยายามตอกหมุดยึดเทียนติดผนังตรงๆ หรือหลอมขี้ผึ้งลงบนผนังเป็นกาว — ล้มเหลวทั้งคู่ น้อยคนมากที่คิดจะเทหมุดออกจากกล่อง ตอกกล่องติดผนัง แล้วตั้งเทียนไว้ข้างใน เพราะ

กล่องถูกจับสวมบท "ภาษาไร้เหตุผล" ไปแล้วตั้งแต่ก่อนผู้ถูกทดลองจะเริ่มคิดด้วยซ้ำ พอ Duncker เปลี่ยนไปยื่นหมุดแบบกองรวมให้ วางไว้ข้างกล่องเปล่า อัตราการแก้สำเร็จก็พุ่งขึ้นชัด — กล่องที่เลิกแต่งตัวเป็นภาษา กลายเป็นชั้นวางที่มองเห็นได้

(Duncker, "On Problem Solving," *Psychological Monographs*, 1945;

Wikipedia, "Candle problem") functional fixedness คือจิตโหลด representation

ไว้ล่วงหน้า — จำยบทให้วัตถุหนึ่งบท แล้วไม่ยอมลองบทอื่นถ้าไม่มีใครมาช่วย —

และมันคือความล้มเหลวแบบเดียวกับที่ดักผู้มาลองทุกคนที่กอร์เดียม: พื้นที่สถานะที่ความเคยชินบีบให้แคบลงเงียบๆ ไม่ใช่กฎจริงของโจทยบีบ

EN

Norman Maier, working at the same moment on the opposite side of the Atlantic, went further and asked a harder question: does insight need to be conscious to work? His two-string problem hangs two cords from the ceiling, too far apart for a subject to hold one and reach the other; a pair of pliers lying nearby can be tied to one cord and swung as a pendulum, letting the subject catch it while holding the first. Subjects who stalled were given a hint disguised as an accident: Maier would walk past and "accidentally" brush one of the strings, setting it swinging. Most who then solved the problem could not say why the answer had occurred to them — in Maier's post-experiment interviews, only around a third of those helped this way recognized they had been cued at all (Maier, "Reasoning in Humans II," *Journal of Comparative Psychology*, 1931; replication and discussion in Landrum, "Maier's (1931) Two-String Problem Revisited," 1990). The unconscious, in other words, can accept a piece of evidence, use it to restructure a solution, and hide the receipt. Wallas's incubation was not a metaphor. It was a mechanism, and Maier had just shown it operating below the floor of self-report.

TH

Norman Maier ทำงานอยู่ในช่วงเวลาเดียวกันที่อีกฝั่งของแอตแลนติก และถามคำถามที่ยากกว่า: การหยั่งเห็นต้องรู้ตัวไหมถึงจะทำงาน โจทย์สองเชือกของเขาแขวนเชือกสองเส้นจากเพดาน ห่างกันเกินกว่าที่ผู้ถูกทดลองจะจับเส้นหนึ่งแล้วเอื้อมถึงอีกเส้น; คีมคู่หนึ่งที่วางอยู่ใกล้ๆ ผูกกับเชือกเส้นหนึ่งแล้วแกว่งเป็นลูกตุ้มได้ ผู้ถูกทดลองจึงรับมันได้ขณะถือเชือกเส้นแรกอยู่ คนที่ตันจะได้คำใบ้ที่อำพรางเป็นอุบัติเหตุ:

Maier เดินผ่านแล้ว "เฟล" ปิดเชือกเส้นหนึ่งให้แกว่ง คนส่วนใหญ่ที่แก้โจทย์ได้หลังจากนั้นบอกไม่ได้ว่าทำไมคำตอบถึงผุดขึ้นมา — ในการสัมภาษณ์หลังการทดลองของ Maier มีเพียงราวหนึ่งในสามของคนที่ได้รับความช่วยเหลือแบบนี้ที่รู้ว่าถูกใบ้ (Maier, "Reasoning in Humans II," *Journal of Comparative Psychology*, 1931; การทำซ้ำและการอภิปรายอยู่ที่ Landrum, "Maier's (1931) Two-String Problem Revisited," 1990) พุดอีกอย่าง จิตไร้สำนึกรับหลักฐานชิ้นหนึ่งเข้าไป ใช้มันปรับโครงสร้างคำตอบใหม่ แล้วซ่อนใบเสร็จไว้ได้ incubation ของ Wallas ไม่ใช่คำเปรียบเปรย มันคือกลไก และ Maier เพิ่งแสดงให้เห็นว่ามันทำงานอยู่ใต้พื้นของสิ่งที่คนรายงานเกี่ยวกับตัวเองได้

E N

Taken together, Köhler's chimpanzees, Duncker's candle, and Maier's swinging string describe the same discovery three ways: solving is not always, or even usually, a matter of trying harder inside a fixed picture of the problem. Often the picture itself is the obstacle, and the picture can change — sometimes all at once, sometimes with help the solver never notices — without the solver doing anything you could call "more search."

T H

ชิมแปนซีของ Köhler เทียนของ Duncker และเชือกแกว่งของ Maier บรรยายการค้นพบเดียวกันสามแบบ: การแก้ไม่ได้เป็นเรื่องของการพยายามหนักขึ้นภายในภาพของโจทย์ที่ตายตัวเสมอไป และส่วนใหญ่ก็ไม่ใช่ด้วยซ้ำ บ่อยครั้งตัวภาพนั่นเองคืออุปสรรค และภาพเปลี่ยนได้ — บางทีเปลี่ยนทีเดียวทั้งภาพ บางทีเปลี่ยนด้วยความช่วยเหลือที่นักแก้ปัญหาไม่เคยสังเกตเห็น — โดยที่นักแก้ปัญหาไม่ได้ทำอะไรที่เรียกว่า "ค้นหาเพิ่ม" เลย

Pólya's Toolbox

กล่องเครื่องมือของ Pólya

EN

All of this — Poincaré's testimony, Wallas's stages, Hadamard's survey, the Gestalt animal and human labs — described what happens to a solver. None of it told a stuck student what to *do* on a Tuesday afternoon with a geometry problem due Wednesday. That gap was closed, decisively and for the first time in a form ordinary people could use, by a Hungarian mathematician who had spent years watching himself and his students think.

TH

ทั้งหมดนี้ — คำให้การของ Poincaré ขั้นตอนของ Wallas การสำรวจของ Hadamard ห้องทดลองสัตว์และมนุษย์ของ Gestalt — บรรยายสิ่งที่เกิดขึ้นกับนักแก้ปัญหาก็ไม่มีอะไรในนั้นบอกนักเรียนที่ตันว่าให้ทำอะไรในบ่ายวันอังคารกับโจทย์เรขาคณิตที่ต้องส่งวันพุธ คนที่ปิดช่องว่างนั้น ปิดอย่างเด็ดขาดและเป็นครั้งแรกในรูปแบบที่คนธรรมดาใช้ได้ คือนักคณิตศาสตร์ชาวฮังการีที่ใช้เวลาหลายปีเฝ้าดูตัวเองและลูกศิษย์คิด

EN

George Pólya's *How to Solve It*, published in 1945, opens with a plan of embarrassing simplicity: **understand the problem, devise a plan, carry out the plan, look back**. Understanding sounds too obvious to need saying, and Pólya's own point is that this is exactly why students skip it and then wonder why they are stuck; his prompts — *what is the unknown, what are the data, can you restate this in your own words* — exist because "understand the problem" is a stage, not a formality (*How to Solve It* (1945), summarized at Wikipedia). But the four phases were only the skeleton. The body of the book is a dictionary of some sixty-odd named heuristics — practical, unguaranteed methods for finding the plan in step two — and it is these that made the book a fixture on desks far from any mathematics department.

TH

How to Solve It ของ George Pólya ตีพิมพ์ปี 1945 เปิดด้วยแผนที่เรียบง่ายจนน่าอาย: เข้าใจโจทย์, วางแผน, ทำตามแผน, มองย้อนกลับ การเข้าใจฟังดูชัดเจนกว่าจะต้องพูด และประเด็นของ Pólya เองก็คือ นี่แหละเหตุผลที่นักเรียนข้ามมันไปแล้ว

มางว่าทำไมถึงตัน; คำถามนำของเขา — อะไรคือสิ่งที่ไม่รู้ อะไรคือข้อมูล เล่าโจทย์นี้ใหม่ด้วยคำของตัวเองได้ไหม — มีอยู่เพราะ "เข้าใจโจทย์" คือขั้นตอนหนึ่ง ไม่ใช่วิธีการ (How to Solve It (1945), summarized at Wikipedia) แต่ที่ขั้นนั้นเป็นแค่โครงกระดูก เนื้อของหนังสือคือพจนานุกรมของวิธีเดาอย่างมีหลัก (heuristic) ที่มีชื่อกว่าหกสิบรายการ — วิธีที่ใช้ได้จริงแต่ไม่มีอะไรรับประกัน สำหรับหาแผนในขั้นที่สอง — และส่วนนี้เองที่พาหนังสือเล่มนี้ไปตั้งอยู่บนโต๊ะที่ห่างไกลจากภาควิชาคณิตศาสตร์

EN

A few of Pólya's heuristics deserve their names remembered, because they recur, disguised, throughout the rest of this book. **Analogy:** if you cannot solve the problem in front of you, have you solved one that resembles it? **The auxiliary problem:** can you invent an easier, related problem, solve that one, and use the solution as a ladder? **Working backward:** start from the goal and ask what state would have to precede it, then what would have to precede that — a heuristic Pólya credits with turning "find a path" into "find a path," read right to left, which is sometimes the shorter search. **Specialization and generalization:** if the problem is too abstract to get a grip on, try a small concrete case; if it is too narrow to see the structure, ask what larger family it belongs to. None of these procedures guarantees an answer. All of them narrow the region of the problem space worth searching, by betting — on the strength of experience with other problems — that the path is more likely to run through analogy, or backward from the goal, or through a simpler cousin, than through blind trial.

TH

heuristic ของ Pólya บางตัวสมควรจดจำชื่อไว้ เพราะมันกลับมาซ้ำในรูปปลอมตัวตลอดทั้งเล่มที่เหลือ **การเทียบเคียง (analogy):** ถ้าแก้โจทย์ตรงหน้าไม่ได้ เคยแก้โจทย์ที่คล้ายกันได้ไหม **โจทย์ช่วย (auxiliary problem):** คิดโจทย์ที่ง่ายกว่าและเกี่ยวข้องกันขึ้นมาเอง แก้อันนั้น แล้วใช้คำตอบเป็นบันไดได้ไหม **ทำย้อนกลับ:** เริ่มจากเป้าหมายแล้วถามว่าสถานะไหนต้องมาก่อนมัน แล้วอะไรต้องมาก่อนตัวนั้น — heuristic ที่ Pólya ยกความดีความชอบว่าเปลี่ยน "หาเส้นทาง" ให้เป็น "หาเส้นทาง" ที่อ่านจากขวาไปซ้าย และบางครั้งนั่นคือการค้นหาที่สั้นกว่า **การเจาะจงและการวางนัยทั่วไป (specialization and generalization):** ถ้าโจทย์นามธรรมเกินกว่าจะจับ

ได้ ลองกรณีเล็กที่เป็นรูปธรรม; ถ้ามันแคบเกินกว่าจะเห็นโครงสร้าง ถามว่ามันอยู่ในตระกูลใหญ่ตระกูลไหน ไม่มีขั้นตอนไหนในนี้รับประกันคำตอบ ทุกตัวแค่บิบบริเวณของพื้นที่ปัญหา (problem space) ที่คุ่มค่าจะค้นให้แคบลง ด้วยการเติมพื้นที่ — โดยอาศัยประสบการณ์จากโจทย์อื่น — ว่าเส้นทางน่าจะพาดผ่านการเทียบเคียง หรือ ย้อนจากเป้าหมาย หรือผ่านญาติที่ง่ายกว่า มากกว่าจะผ่านการลองแบบมีติดบอด

E N

The book's afterlife is its real argument. Marvin Minsky, writing in 1961 at the dawn of artificial intelligence research, told his readers flatly that "everyone should know the work of George Pólya on how to solve problems" (quoted in Wikipedia, "How to Solve It"), and Allen Newell — who with Herbert Simon would go on to build the problem-space formalism this book leans on throughout (chapter 00) — later wrote an entire paper examining exactly how much of Pólya's heuristic vocabulary a computer program could be made to use, and how much resisted translation into code (Newell, "The Heuristic of George Polya and Its Relation to Artificial Intelligence," 1981). Newell's conclusion was double-edged: Pólya's heuristics were recognized by everyone in the field as the right *kind* of idea, and yet decades of effort proved how hard it was to turn "notice an analogy" into an algorithm. A heuristic that a human applies with judgment is not automatically a heuristic a machine can apply at all — a tension this book will meet again when the search moves from human heads into silicon (chapters 04 through 06, and again in chapter 10).

T H

ชีวิตหลังตีพิมพ์ของหนังสือคือข้อโต้แย้งที่แท้จริงของมัน Marvin Minsky เขียนไว้ในปี 1961 ณ รุ่งอรุณของงานวิจัยปัญญาประดิษฐ์ บอกผู้อ่านตรงๆ ว่า "ทุกคนควรรู้จักงานของ George Pólya เรื่องการแก้ปัญหา" ("everyone should know the work of George Pólya on how to solve problems") (quoted in Wikipedia, "How to Solve It") และ Allen Newell — ผู้ที่ร่วมกับ Herbert Simon สร้างรูปนัยของพื้นที่ปัญหาที่หนังสือเล่มนี้พึ่งอยู่ตลอด (บทที่ 00) — ภายหลังเขียนบทความทั้งบทความตรวจสอบว่าคำศัพท์ heuristic ของ Pólya มีมากแค่ไหนที่ทำให้โปรแกรมคอมพิวเตอร์ใช้ได้ และมากแค่ไหนที่ต้านการแปลเป็นโค้ด (Newell, "The Heuristic of George Polya and Its Relation to Artificial Intelligence," 1981) ข้อสรุปของ

Newell มีสองคม: ทุกคนในวงการยอมรับว่า heuristic ของ Pólya เป็นไอเดียชนิดที่ถูกต้อง แต่ความพยายามหลายทศวรรษก็พิสูจน์ว่าการเปลี่ยน "สังเกตเห็นการเทียบเคียง" ให้เป็นอัลกอริทึมนั้นยากเพียงใด heuristic ที่มนุษย์ใช้ด้วยวิจารณญาณไม่ได้แปลว่าเครื่องจักรจะใช้มันได้เลย — ความตึงที่หนังสือเล่มนี้จะเจออีกครั้งเมื่อการค้นหาย้ายจากหัวคนไปสู่ซิลิคอน (บทที่ 04 ถึง 06 และอีกครั้งในบทที่ 10)

Forty Thousand Patents and One Idea

สิทธิบัตรสี่หมื่นฉบับ กับไอเดียเดียว

EN

Half a world away, and unaware for years that Pólya's book even existed, a Soviet inventor was solving the identical problem from the opposite trade. Genrich Altshuller worked reviewing patent applications for the Soviet Navy, and it needed him that engineers kept "inventing" solutions that had already been invented, in different industries, under different jargon. He began, starting in 1946, to read patents not for their content but for their *shape*: what kind of difficulty had the inventor faced, and what kind of move had resolved it. Building the classical contradiction matrix, Altshuller and his collaborators examined on the order of forty thousand patents drawn from a much larger pool — one contemporary account puts the source pool at roughly four hundred thousand — screening for the subset that represented genuine breakthroughs rather than routine refinements (Savransky, "A Few Words About the Altshuller's Contradiction Matrix," *The TRIZ Journal*, 1997). Exact patent counts vary across the movement's own retellings — later organizers cite larger cumulative totals as the research continued for decades after 1946 — but the order of magnitude, tens of thousands sifted by hand and eye, is not in dispute (Wikipedia, "TRIZ").

TH

อีกซีกโลกหนึ่ง นักประดิษฐ์โซเวียตคนหนึ่งกำลังแก้โจทย์เดียวกันจากอีกอาชีพ และไม่รู้มาหลายปีด้วยซ้ำว่าหนังสือของ Pólya มีอยู่ Genrich Altshuller ทำงานตรวจคำขอสิทธิบัตรให้กองทัพเรือโซเวียต และมันขัดใจเขาที่วิศวกรเอาแต่ "ประดิษฐ์" ทางแก้ที่ถูกประดิษฐ์ไปแล้วในอุตสาหกรรมอื่น ได้ศัพท์เฉพาะชุดอื่น ตั้งแต่ปี 1946 เขาเริ่มอ่านสิทธิบัตรไม่ใช่เพื่อเนื้อหา แต่เพื่อรูปทรงของมัน: นักประดิษฐ์เจอความยาก

ชนิดไหน และทำชนิดไหนที่คลี่คลายมัน ในการสร้างเมทริกซ์ความขัดแย้งฉบับคลาสสิก Altshuller และผู้ร่วมงานตรวจสอบสิทธิบัตรราวสี่หมื่นฉบับที่ดึงมาจากกองที่ใหญ่กว่านั้นมาก — บันทึกพร้อมสมัยฉบับหนึ่งระบุกองต้นทางไว้ที่ราวสี่แสนฉบับ — โดยคัดเฉพาะส่วนที่เป็นการทะเลาะลงของจริง ไม่ใช่การขัดเกลามาตามปกติ (Savransky, "A Few Words About the Altshuller's Contradiction Matrix," The TRIZ Journal, 1997) จำนวนสิทธิบัตรที่แน่นอนต่างกันไปในแต่ละฉบับที่ขบวนการนี้เล่าเอง — ผู้จัดตั้งรุ่นหลังอ้างยอดสะสมที่ใหญ่กว่า เพราะงานวิจัยเดินต่ออีกหลายทศวรรษหลังปี 1946 — แต่อันดับของขนาด คือหลักหมื่นฉบับที่ถูกร่อนด้วยมือและตา ไม่มีใครเถียง (Wikipedia, "TRIZ")

E N

What Altshuller extracted from that reading was TRIZ — *Teoriya Resheniya Izobretatelskikh Zadatch*, the Theory of Inventive Problem Solving — built on a single organizing unit he called the **technical contradiction**: improving one parameter of a system (make it lighter) degrades another (it becomes weaker). Altshuller's claim, radical for engineering in 1946, was that a genuine invention is not a compromise between the two — thinner and therefore weaker — but a move that escapes the trade-off altogether, and that the moves which escape trade-offs repeat across industries, so that the solution to a chemical-plant problem might be structurally identical to a solution already patented in bridge-building. He and his colleagues eventually distilled this into forty named inventive principles and a matrix for finding, given any pair of colliding parameters, which of the forty was likeliest to apply (Wikipedia, "TRIZ").

T H

สิ่งที่ Altshuller สกัดออกมาจากการอ่านนั้นคือ TRIZ — *Teoriya Resheniya Izobretatelskikh Zadatch* หรือทฤษฎีการแก้ปัญหาเชิงประดิษฐ์คิดค้น — สร้างขึ้นบนหน่วยจัดระเบียบเดียวที่เขาเรียกว่า **ความขัดแย้งเชิงเทคนิค (technical contradiction)**: ปรับพารามิเตอร์หนึ่งของระบบให้ดีขึ้น (ทำให้เบาลง) แล้วอีกพารามิเตอร์แย่ลง (มันอ่อนแอลง) ข้อกล่าวอ้างของ Altshuller สุดโต่งสำหรับวิศวกรในปี 1946: การประดิษฐ์ของจริงไม่ใช่การประนีประนอมระหว่างสองอย่างนั้น — บางลงและอ่อนแอลงตามไปด้วย — แต่คือทำที่หนึ่งการแลกได้แลกเสียไปได้

ทั้งหมด และทำที่หนึ่งได้นั้นซ้ำกันข้ามอุตสาหกรรม จนทางแก้ของโจทย์ในโรงงานเคมี อาจเหมือนกันเชิงโครงสร้างกับทางแก้ที่จดสิทธิบัตรไว้แล้วในงานสร้างสะพาน สุดท้ายเขากับเพื่อนร่วมงานกลั่นมันลงเหลือหลักการประดิษฐ์ที่มีข้อสี่สิบข้อ และเมทริกซ์สำหรับหาว่า เมื่อพารามิเตอร์คู่ใดชนกัน ข้อไหนในสี่สิบข้อน่าจะใช้ได้มากที่สุด (Wikipedia, "TRIZ")

EN

The convergence with Pólya is not coincidence dressed up as destiny — it is what you would expect if both men had, independently, put their hands on the same underlying fact: across wildly different domains, the *number of genuinely different move-types* is far smaller than the number of problems they solve. Pólya found perhaps sixty for mathematics. Altshuller found forty for engineering. Neither number is sacred, and this book will not pretend the two lists are secretly the same list. But both men had discovered that a stuck solver rarely needs a new idea invented from nothing; far more often what is needed is a shift into a known *kind* of move, correctly matched to the kind of obstacle in front of them.

TH

การบรรจบกันกับ Pólya ไม่ใช่ความบังเอิญที่แต่งตัวเป็นพรหมลิขิต — มันคือสิ่งที่ควรคาดได้ ถ้าทั้งสองคนต่างวางมือลงบนข้อเท็จจริงพื้นฐานอันเดียวกันอย่างเป็นอิสระ: ข้ามโดเมนที่ต่างกันสุดขีด จำนวนชนิดของท่าที่ต่างกันจริงๆ น้อยกว่าจำนวนโจทย์ที่มันแก้ได้มาก Pólya พบราวหกสิบสำหรับคณิตศาสตร์ Altshuller พบสี่สิบสำหรับวิศวกรรม ไม่มีตัวเลขไหนศักดิ์สิทธิ์ และหนังสือเล่มนี้จะไม่แสวงว่าสองรายการนั้น แอบเป็นรายการเดียวกัน แต่ทั้งคู่ค้นพบว่านักแก้ปัญหาก็ค้นพบไม่เคยต้องการไอเดียใหม่ที่สร้างจากความว่างเปล่า สิ่งที่ต้องการบ่อยกว่ามากคือขยับไปหาชนิดของท่าที่รู้จักอยู่แล้ว จับคู่ให้ถูกกับชนิดของอุปสรรคที่อยู่ตรงหน้า

Invert, Always Invert

กลับด้าน กลับด้านเสมอ

EN

One of those move-types is old enough to predate the whole psychological turn, and simple enough to state in five words: when stuck, try the opposite

direction. The German mathematician Carl Gustav Jacob Jacobi is widely credited with the maxim *man muss immer umkehren* — "one must always invert" — reportedly drilled into his students as a standing instruction whenever a differential equation resisted a direct attack: stop asking what the integral of this expression is, and ask instead what expression this could be the integral of. The attribution deserves the same caution this book gave the sword and the pin at Gordium: the phrase survives only as something Jacobi "is said to have" told his students, quoted at several removes and popularized outside mathematics only much later, so the wording — and possibly the entire anecdote — should be read as folklore with a mathematical core, not a verified transcript (Mathematical Association of America, "Quotations in Context: Jacobi"). What is not folklore is the technique's power: Jacobi and Niels Henrik Abel both made their names by inverting elliptic integrals — asking not "what value does this integral take" but "what limits produce this value" — and finding, on the far side of the inversion, a whole new family of well-behaved functions waiting to be named.

TH

หนึ่งในชนิดของท่าเหล่านั้นเก่าแก่กว่าการหันเข้าหาจิตวิทยาทั้งหมดนี้ และเรียบง่ายพอจะพูดได้ในห้าคำ: เมื่อต้น ลองทิสตรงข้าม นักคณิตศาสตร์ชาวเยอรมัน Carl Gustav Jacob Jacobi ได้รับเครดิตอย่างกว้างขวางสำหรับคติ *man muss immer umkehren* — "ต้องกลับด้านเสมอ" ("one must always invert") — มีรายงานว่าเขาปลูกฝังใส่ลูกศิษย์เป็นคำสั่งประจำทุกครั้งที่สมการเชิงอนุพันธ์ด้านการโจมตีตรงๆ: เลิกถามว่าปริพันธ์ของนิพจน์นี้คืออะไร แล้วถามแทนว่านิพจน์นี้เป็นปริพันธ์ของอะไรได้บ้าง การให้เครดิตนี้ต้องระวังเท่ากับที่หนังสือเล่มนี้ระวังดาบและหมุดที่เกอร์เดียม: วลีนี้ยู่รอดมาในฐานะสิ่งที่ Jacobi "ว่ากันว่า" บอกลูกศิษย์เท่านั้น ถูกอ้างต่อกันมาหลายทอด และเพ่งแพร่หลายนอกวงคณิตศาสตร์ในเวลาต่อมามาก ฉะนั้นถ้อยคำ — และอาจรวมถึงเกร็ดทั้งเกร็ด — ควรอ่านเป็นคติชนที่มีแกนทางคณิตศาสตร์ ไม่ใช่บันทึกคำต่อคำที่ตรวจสอบแล้ว (Mathematical Association of America, "Quotations in Context: Jacobi") สิ่งที่ไม่ใช่คติชนคือพลังของเทคนิคนี้: Jacobi และ Niels Henrik Abel ต่างสร้างชื่อจากการกลับด้านปริพันธ์เชิงวงรี — ไม่ถามว่า "ปริพันธ์นี้มีค่าเท่าไร" แต่ถามว่า "ขอบเขตแบบไหนให้ค่านี้" — แล้วพบ

ฟังก์ชันตระกูลใหม่ทั้งตระกูลที่มีพฤติกรรมดีรออยู่ให้ตั้งชื่อ อยู่อีกฟากของการกลับด้าน (inversion)

E N

Inversion belongs on Pólya's list in spirit if not by his exact label — it is a cousin of working backward — and it is the same move Alexander made at Gordium in chapter 00, whether by sword or by pin: refuse the direction the problem was handed to you in, and search the reverse instead. It will re-surface, formalized into an algorithm rather than a maxim, when this book reaches dynamic programming's habit of solving a problem from its last step backward (chapter 04).

T H

การกลับด้านอยู่ในรายการของ Pólya โดยจิตวิญญาณ ถึงจะไม่ตรงกับป้ายชื่อที่เขาใช้เป๊ะๆ — มันเป็นญาติของการทำย้อนกลับ — และมันคือทำเดียวกับที่ Alexander ใช้ที่กอร์เดียมในบทที่ 00 ไม่ว่าจะด้วยดาบหรือด้วยหมุด: ปฏิเสธทิศทางที่โจทย์ถูกยื่นมาให้ แล้วค้นย้อนกลับแทน มันจะโผล่มาอีก ในรูปอัลกอริทึมแทนที่จะเป็นคติ เมื่อหนังสือเล่มนี้ไปถึงนิสัยของการพลวัต (dynamic programming) ที่แก้โจทย์จากขั้นสุดท้ายย้อนกลับมา (บทที่ 04)

The Searchlight, Not the Guarantee

ไฟฉายส่องทาง ไม่ใช่คำรับประกัน

E N

Step back from the individual names and a single shape emerges from this whole chapter. Wallas's incubation, Hadamard's wordless thought, Köhler's insight, Duncker's fixedness, Maier's hidden hint, Pólya's dictionary, Altshuller's contradictions, Jacobi's inversion — none of them is a procedure that finds the shortest path by exhausting the space and comparing every route. Every one of them is a bet, made in advance of full information, about *where in the space the short paths are statistically likely to hide*. Analogy bets that a problem's structure has already been solved once, somewhere else, in different clothing. Working backward bets that the goal state constrains its predecessors more tightly than the start state constrains its successors.

Altshuller's contradiction matrix bets that whatever principle broke a trade-off in aircraft design will also break one in chocolate packaging.

TH

ถอยออกจากข้อแต่ละข้อ แล้วทั้งบทนี้เหลือรูปทรงเดียว incubation ของ Wallas ความคิดที่ไร้ถ้อยคำของ Hadamard insight ของ Köhler ความยึดติดของ Duncker คำใบ้ที่ซ่อนอยู่ของ Maier พจนานุกรมของ Pólya ความขัดแย้งของ Altshuller การกลับด้านของ Jacobi — ไม่มีอันไหนเลยเป็นขั้นตอนที่หาเส้นทางสั้นที่สุดด้วยการไล่พื้นที่จนหมดแล้วเทียบทุกเส้นทาง ทุกอันคือการเดิมพัน วางลงก่อนจะมีข้อมูลครบว่าในพื้นที่นั้น เส้นทางสั้นๆ มีแนวโน้มทางสถิติจะซ่อนอยู่ตรงไหน การเทียบเคียงเดิมพันว่าโครงสร้างของโจทย์เคยถูกแก้ไปแล้วครั้งหนึ่ง ที่ไหนสักแห่ง ในเสื้อผ้าอีกชุด การทำย้อนกลับเดิมพันว่าสถานะเป้าหมายบีบสิ่งที่มาก่อนมันแน่นกว่าที่สถานะตั้งต้นบีบสิ่งที่ตามมันมา เมทริกซ์ความขัดแย้งของ Altshuller เดิมพันว่าหลักการที่หักการแลกได้แลกเสียในงานออกแบบอากาศยาน จะหักการแลกได้แลกเสียในการบรรจุช็อกโกแลตได้ด้วย

EN

This is why the heuristic age is not the end of this book's argument but its opening move. A heuristic does not change the problem space — it does not throw away states, redefine the goal, or invent new operators, the way Euler's abstraction of Königsberg did in chapter 02's aftermath, or the way Snow's re-definition of a "case" will in chapter 07. A heuristic simply shines a searchlight into a fixed space and says: *look here first*. That is real, useful, and teachable — Pólya's book has stayed in print continuously since 1945 for exactly that reason (Wikipedia, "How to Solve It") — but it is a smaller claim than reframing, and honesty requires saying so before the next four chapters show what happens when solvers stop shining lights into the given space and start rebuilding the space itself.

TH

ยุค heuristic จึงไม่ใช่บทสรุปของข้อโต้แย้งในหนังสือเล่มนี้ แต่เป็นหมากเปิดของมัน heuristic ไม่เปลี่ยนพื้นที่ปัญหา — มันไม่ทิ้งสถานะ ไม่นิยามเป้าหมายใหม่ ไม่คิดตัวดำเนินการ (operator) ใหม่ขึ้นมา อย่างที่การทำ Königsberg ให้เป็นนามธรรมของ Euler ทำในผลพวงของบทที่ 02 หรืออย่าง que การนิยาม "เคส" ใหม่ของ Snow จะทำ

ในบทที่ 07 heuristic แค่อ้างไฟฉายเข้าไปในพื้นที่ที่ตายตัวแล้วบอกว่า ดูตรงนี้ก่อน นั่นเป็นของจริง มีประโยชน์ และสอนกันได้ — หนังสือของ Pólya พิมพ์ต่อเนื่องมาตั้งแต่ปี 1945 ด้วยเหตุผลนี้เป๊ะๆ (Wikipedia, "How to Solve It") — แต่มันเป็นข้อกล่าวอ้างที่เล็กกว่าการเปลี่ยนกรอบ และความเชื่อตรงเรียกร้องให้พุดออกมาก่อนที่จะบดบังไปจะแสดงว่าเกิดอะไรขึ้นเมื่อนักแก้ปัญหาเลิกส่องไฟเข้าไปในพื้นที่ที่ถูกยืมมาให้ แล้วเริ่มสร้างพื้นที่ขึ้นใหม่เอง

The Move

ทำประจำบท

E N

Move 03 — Carry named tools.

T H

Move 03 — พกเครื่องมือที่มีชื่อ

E N

A heuristic is not a guarantee. It is a bet, made by someone before you, about where in a space of possibilities the short paths tend to cluster — and every bet has a failure mode you should know before you place it.

T H

heuristic ไม่ใช่คำรับประกัน มันคือการเดิมพันที่คนก่อนหน้าคุณวางไว้ ว่าในพื้นที่ของความเป็นไปได้ เส้นทางสั้นๆ มักจับกลุ่มกันอยู่ตรงไหน — และการเดิมพันทุกครั้งมีโหมดความล้มเหลวที่ควรรู้ก่อนวางเงิน

E N

- **Keep a short deck, not a long one.** Pólya's dictionary runs past sixty entries and Altshuller's matrix has forty principles, but almost no solver holds more than a handful in working memory under pressure. Choose a working set — analogy, working backward, the auxiliary problem, inversion are a defensible starting four — and know them cold rather than knowing a hundred shallowly.
- **Name the failure mode with the tool.** Analogy fails when the surface resembles a solved problem but the structure does not. Working backward

fails when the goal state under-constrains its predecessors. Functional fixedness is not a flaw in the solver, Duncker and Maier showed — it is a predictable cost of any heuristic that pre-assigns roles to objects, including the useful ones.

- **Protect incubation on purpose.** Poincaré did not summon the omnibus insight; he set the problem down and let a fortnight of grinding preparation finish itself off-screen. If a plan has failed after honest effort, walking away is not surrender — it is deploying the one stage of discovery, per Wallas and Hadamard, that specifically requires you to stop looking.
- **Treat a heuristic's silence as information.** When your whole deck comes up empty against a problem, that is evidence the space itself is wrong, not that you need a sixty-first heuristic. That is the cue to stop searching within the given frame and start asking what would happen if you changed it — which is where this book goes next.

TH

- **ถือสารับสัน ไม่ใช่สำหรับยาว** พจนานุกรมของ Pólya ยาวเกินหกสิบรายการ และเมทริกซ์ของ Altshuller มีสี่สิบหลักการ แต่แทบไม่มีนักแก้ปัญหาคนไหนถือไว้ในความจำใช้งาน (working memory) ได้เกินหยิบมือเดียวตอนอยู่ใต้ความกดดันเลือกชุดทำงานมาชุดหนึ่ง — การเทียบเคียง ทำย้อนกลับ โจทย์ช่วย การกลับด้าน คือสิ่งที่ตัวตั้งต้นที่ป้องกันตัวเองได้ — แล้วรู้จักมันให้ทะลุ ดีกว่ารู้จักร้อยตัวแบบผิวๆ
- **ตั้งชื่อโหมดความล้มเหลวไว้คู่กับเครื่องมือ** การเทียบเคียงล้มเหลวเมื่อผิวหน้าคล้ายโจทย์ที่แก้แล้วแต่โครงสร้างไม่คล้าย การทำย้อนกลับล้มเหลวเมื่อสถานะเป้าหมายบิบบสิ่งที่มีมาก่อนมันไม่พอ functional fixedness ไม่ใช่ข้อบกพร่องของนักแก้ปัญหา — Duncker กับ Maier แสดงให้เห็นแล้วว่ามันคือต้นทุนที่คาดเดาได้ของ heuristic ทุกตัวที่จ่ายบทบาทให้วัตถุไว้วงหน้า รวมถึงตัวที่มีประโยชน์ด้วย
- **ปกป้อง incubation อย่างจงใจ** Poincaré ไม่ได้เรียก insight บนรถโดยสารนั้นมา เขาวางโจทย์ลงแล้วปล่อยให้การเตรียมที่บดขยี้มาสองสัปดาห์เดินจนจบนอกจอ ถ้าแผนหนึ่งล้มเหลวหลังจากลงแรงอย่างซื่อสัตย์แล้ว การเดินจากไปไม่ใช่การยอมแพ้ — มันคือการใช้ขั้นตอนหนึ่งเดียวของการค้นพบ ตามที่ Wallas และ Hadamard ว่าเป็น ซึ่งเรียกร้องให้หยุดมองโดยเฉพาะ

- **ถือความเจ็บบของ heuristic เป็นข้อมูล** เมื่อสำหรับทั้งสำหรับเปิดออกมาว่างเปล่าต่อโจทย์หนึ่ง นั่นคือหลักฐานว่าตัวพื้นที่นั้นแหละที่ผิด ไม่ใช่ว่าต้องการ heuristic ตัวที่หกสิบเอ็ด นั่นคือสัญญาณให้หยุดค้นหาภายในกรอบที่ถูกยื่นมาให้ แล้วเริ่มถามว่า จะเกิดอะไรขึ้นถ้าเปลี่ยนมัน — ซึ่งคือที่ที่หนังสือเล่มนี้จะไปต่อ

EN

Next — Chapter 04 · War and the Science of Search: heuristics compressed the search inside one mind. The Second World War would industrialize it — turning the shortest path into a discipline with a budget, a uniform, and a body count.

TH

ต่อไป — บทที่ 04 · สงครามกับศาสตร์แห่งการค้นหา: heuristic บีบการค้นหาให้สั้นลงภายในหัวคนคนเดียว สงครามโลกครั้งที่สองจะทำให้มันกลายเป็นอุตสาหกรรม — เปลี่ยนเส้นทางที่สั้นที่สุดให้เป็นวินัยที่มีงบประมาณ มีระเบียบ และมียอดผู้เสียชีวิต

CHAPTER 04

War and the Science of Search

สงครามกับศาสตร์แห่งการค้นหา

Portsmouth, 1941

Portsmouth 1941

EN

Patrick Blackett had won a Nobel-caliber reputation in physics — he would in fact collect the actual Nobel Prize in 1948, for work on cosmic rays — and in the spring of 1941 he was doing something no physicist had ever been assigned to do before: sitting in a Royal Air Force operations room, watching U-boats sink merchant ships faster than Britain could build them, and trying to compute his way out of it.

TH

ฤดูใบไม้ผลิปี 1941 Patrick Blackett นั่งอยู่ในห้องบัญชาการปฏิบัติการของ Royal Air Force ฝ้าดูเรืออูจมเรือสินค้าเร็วกว่าที่อังกฤษจะต่อขึ้นมาใหม่ได้ และพยายามคำนวณหาทางออกจากมัน ไม่เคยมีนักฟิสิกส์คนไหนถูกมอบหมายให้ทำงานแบบนี้มาก่อน เขาสร้างชื่อเสียงระดับโนเบลไว้ในวงการฟิสิกส์แล้ว — และได้รับรางวัลโนเบลจริงในปี 1948 จากงานเรื่องรังสีคอสมิก

E N

The team around him had no official name, so people called it Blackett's circus: a physiologist, a physicist, an astrophysicist, and eventually two future Nobel laureates, all seconded from their laboratories to Coastal Command's new Operational Research Section (INFORMS biographical profile of Blackett). None of them had ever fought a war. What they had was a habit, bred into every physicist, of refusing to trust an intuition that hadn't been checked against a number.

T H

ทีมของเขาไม่มีชื่อทางการ คนจึงเรียกว่า Blackett's circus: นักสรีรวิทยาหนึ่งคน นักฟิสิกส์หนึ่งคน นักฟิสิกส์ดาราศาสตร์หนึ่งคน และในที่สุดผู้ที่จะได้รางวัลโนเบลในอนาคตอีกสองคน ทุกคนถูกยืมตัวจากห้องทดลองมาประจำ Operational Research Section ที่เพิ่งตั้งขึ้นของ Coastal Command (INFORMS biographical profile of Blackett) ไม่มีใครในทีมเคยรบในสงครามมาก่อน สิ่งที่พวกเขามีคือนิสัยที่นักฟิสิกส์ทุกคนถูกปลูกฝังมา: ไม่เชื่อสัญชาตญาณที่ยังไม่ได้เทียบกับตัวเลข

E N

The intuition they checked first nearly ended a career. Britain's convoys — merchant ships sailing in escorted groups across the Atlantic — had for years followed an unspoken rule: keep them small, because a smaller cluster of ships is a smaller target, and fewer eggs are lost if the basket is dropped. Naval officers as senior as America's Admiral Ernest King believed this so firmly that it barely seemed worth testing. Blackett's group tested it anyway, and found the opposite. A U-boat has to *find* a convoy before it can attack it, and the ocean is enormous; the area available for a submarine to search does not care how many ships are inside a convoy's perimeter. Once a raider does find a convoy, meanwhile, its stock of torpedoes is limited, so a convoy of 80 ships loses only slightly more tonnage to a single attacker than a convoy of 40 — while *half as many* convoys have to cross the ocean and expose themselves to the risk of being found at all. The reasoning had a clean geometric core: pack merchant ships inside a circle and the number you can fit grows with the *area*, which scales as the square of the radius, while the escorts patrolling the *perimeter* only need to scale with the radius itself. Double the convoy's size and you do not need to double its escort — a

few more escort ships now guard proportionally far more cargo (Pacific War Encyclopedia, "Convoys"). The British Admiralty had clung to an "ideal" convoy size near 40 ships for no reason anyone could articulate; when it was finally persuaded to run convoys as large as shipping would allow, sinkings fell. Each escort vessel built, the analysts calculated, saved on the order of two merchant ships over the course of the war.

TH

สัญญาณข้อมแรกทีพวกเขาตรวจสอบเกือทำให้อาชีพใครคนหนึ่งจบลง กองเรือคุ้มกันของอังกฤษ — เรือสินค้าที่แล่นเป็นกลุ่มข้ามแอตแลนติกโดยมีเรือคุ้มกัน — ยึดกฎที่ไม่มีใครพูดออกมาดังๆ มาหลายปี: ให้กองเล็กไว้ กลุ่มเรือที่เล็กกว่าคือเป้าที่เล็กกว่า และถ้าตะกร้าหล่น ไขก็เสียน้อยกว่า นายทหารเรือระดับสูงถึงชั้นพลเรือเอก Ernest King ของอเมริกาเชื่อเรื่องนี้แน่นจนแทบไม่คิดว่าคุ้มจะทดสอบ กลุ่มของ Blackett ทดสอบอยู่ดี และพบสิ่งตรงข้าม เรือต้องหากองเรือให้เจอก่อนจึงโจมตีได้ และมหาสมุทรกว้าง พื้นที่ที่เรือดำน้ำต้องค้นไม่ขึ้นกับว่ามีเรือกี่ลำอยู่ในวงของกองเรือ พอเรือล่าเจอกองเรือแล้ว ตอร์ปิโดที่มันมีก็จำกัด กองเรือ 80 ลำจึงเสียระวางบรรทุกให้ผู้โจมตีรายเดียวมากกว่ากองเรือ 40 ลำเพียงเล็กน้อย — ในขณะที่จำนวนกองเรือที่ต้องข้ามมหาสมุทรและเปิดตัวเองให้เสี่ยงต่อการถูกพบตั้งแต่แรกนั้นลดลงครึ่งหนึ่ง แก่นของเหตุผลเป็นเรขาคณิตล้วน: อัลดเรือสินค้าเข้าไปในวงกลม จำนวนที่ใส่ได้โตตามพื้นที่ คือกำลังสองของรัศมี ส่วนเรือคุ้มกันที่ลาดตระเวนตามเส้นรอบวงเพิ่มตามรัศมีเท่านั้น เพิ่มขนาดกองเรือเป็นสองเท่า เรือคุ้มกันไม่ต้องเพิ่มเป็นสองเท่า — เรือคุ้มกันอีกไม่กี่ลำคุ้มสินค้าได้มากกว่าเดิมมากตามสัดส่วน (Pacific War Encyclopedia, "Convoys") กองทัพเรืออังกฤษยึดขนาดกองเรือ "ในอุดมคติ" ไว้ที่ราว 40 ลำ โดยไม่มีใครอธิบายเหตุผลได้ เมื่อถูกโน้มน้าวในที่สุดให้เดินกองเรือใหญ่เท่าที่การขนส่งเอื้อ ยอดเรือถูกจมก็ลดลง นักวิเคราะห์คำนวณว่าเรือคุ้มกันที่ต่อขึ้นแต่ละลำช่วยเรือสินค้าไว้ได้ราวสองลำตลอดช่วงสงคราม

EN

The same circus turned its calculators on the RAF's depth charges — the underwater bombs dropped on submerged U-boats — and found that crews had been fusing them to explode at roughly 100 feet, the depth a submarine commander might plausibly dive to escape. But data on actual attacks showed most U-boats were still near the surface, diving in a panic, when

they were caught. Reset the fuse to explode at about 35 feet instead, and a depth charge that used to miss its target because it detonated too deep would now catch boats where they actually were. The change alone is credited with roughly doubling the kill rate of attacks that scored a hit (INFORMS, Blackett profile). Nobody built a new weapon. Nobody found a new tactic. Somebody looked at the numbers already sitting in a filing cabinet and asked a sharper question of them.

TH

ชนวนตั้งผิดความลึก นั่นคือสิ่งที่คณะละครสัตว์คณะเดิมพบ เมื่อหันเครื่องคิดเลขไปที่ระเบิดลึกของ RAF — ระเบิดใต้น้ำที่ทิ้งใส่เรือดำน้ำอยู่ ลูกเรือตั้งชนวนให้ระเบิดที่ราว 100 ฟุต ความลึกที่ผู้บังคับการเรือดำน้ำน่าจะดำหนีลงไป แต่ข้อมูลการโจมตีจริงบอกว่าเรือส่วนใหญ่ยังอยู่ใกล้ผิวน้ำตอนถูกจับได้ กำลังดำลงอย่างตื่นตระหนก ตั้งชนวนใหม่ให้ระเบิดที่ราว 35 ฟุตแทน ระเบิดลึกที่เคยพลาดเป้าเพราะจุดระเบิดลึกเกินไป ก็จับเรือได้ตรงที่มันอยู่จริง การเปลี่ยนแค่นี้อย่างเดียวได้เครดิตว่าเพิ่มอัตราการจมของการโจมตีที่ยิงเข้าเป้าขึ้นราวเท่าตัว (INFORMS, Blackett profile) ไม่มีใครสร้างอาวุธใหม่ ไม่มีใครคิดยุทธวิธีใหม่ มีคนมองตัวเลขที่อยู่ในตู้เอกสารอยู่แล้ว และถามคำถามที่คมกว่าเดิมกับมัน

EN

This is the discovery this chapter is about, and it is worth stating plainly before the stories pile up: in the crucible of the Second World War, a scattered handful of physicists, statisticians, and economists realized that war — and soon after, business, agriculture, and engineering — was full of decisions that *looked* like a matter of experience and nerve but were actually optimization problems wearing a uniform. Once you saw them that way, you could solve them with arithmetic instead of instinct. The convoy problem and the depth-charge problem look nothing alike on the surface. Underneath, both are the same move: measure what actually happens, not what you assume happens, and let the measurement — not the veteran's gut — choose the number.

TH

บทนี้ว่าด้วยการค้นพบข้อเดียว และควรพูดตรงๆ ก่อนเรื่องเล่าจะกองสุมกัน: ในบ้ำหลอมของสงครามโลกครั้งที่สอง นักฟิสิกส์ นักสถิติ และนักเศรษฐศาสตร์กลุ่มเล็กๆ

ที่กระจัดกระจายตระหนักรู้ว่าสงคราม — และไม่นานหลังจากนั้น ธุรกิจ เศรษฐกรกรรม และวิศวกรรม — เต็มไปด้วยการตัดสินใจที่ดูเหมือนเป็นเรื่องของประสบการณ์และความกล้า แต่แท้จริงคือโจทย์การหาค่าเหมาะที่สุด (optimisation) ที่สวมเครื่องแบบมา มองเห็นแบบนั้นแล้ว ก็แก้ด้วยเลขคณิตแทนสัญชาตญาณได้ โจทย์กองเรือกับโจทย์ระเบิดลึกลงไม่เหมือนกันเลยที่ผิวหน้า ช่างได้คือทำเดียวกัน: วัดสิ่งที่เกิดขึ้นจริง ไม่ใช่สิ่งที่สมมติว่าเกิด แล้วให้การวัด — ไม่ใช่สัญชาตญาณของทหารผ่านศึก — เป็นผู้เลือกตัวเลข

The Bombers That Didn't Come Home

เครื่องบินทิ้งระเบิดที่ไม่ได้กลับบ้าน

EN

Across the ocean, a mathematician named Abraham Wald was solving a problem with the same shape and a very different flavor, and his name has since attached itself to one of the most retold anecdotes in applied statistics — a story worth telling *and* worth correcting, because the correction is itself a lesson in this book's thesis.

TH

อีกฟากมหาสมุทร Abraham Wald นักคณิตศาสตร์ กำลังแก้โจทย์รูปทรงเดียวกันแต่รสชาติต่างกันไปมาก ชื่อของเขาติดอยู่กับเกร็ดที่ถูกเล่าซ้ำมากที่สุดเรื่องหนึ่งในสถิติประยุกต์ตั้งแต่นั้นมา — เรื่องที่คุ้มจะเล่าและคุ้มจะแก้ไขให้ถูก เพราะตัวการแก้เองคือบทเรียนของแก่นความคิดในหนังสือเล่มนี้

EN

The version everyone has heard goes like this: the U.S. Army, wanting to armor its bombers without making them too heavy to fly, brought Wald data on where returning aircraft had taken damage — mostly on the fuselage and wings, rarely near the engines — and prepared to reinforce the fuselage. Wald, the story says, told them they had it backwards: armor the engines, because the planes that got hit *there* were the ones that never made it home to be counted. This is the "survivorship bias" story, and as an illustration of the idea it is close to perfect.

TH

เวอร์ชันที่ทุกคนเคยได้ยินเป็นแบบนี้: กองทัพบกสหรัฐฯ อยากติดเกราะเครื่องบินทั้งระเบิดโดยไม่ให้หนักจนบินไม่ขึ้น จึงนำข้อมูลมาให้ Wald ว่าเครื่องบินกลับมาเสียหายตรงไหน — ส่วนใหญ่ที่ลำตัวและปีก น้อยครั้งที่ใกล้เครื่องยนต์ — และเตรียมเสริมเกราะที่ลำตัว เรื่องเล่าว่า Wald บอกพวกเขาว่าคิดกลับหัวกลับหาง: ให้ติดเกราะที่เครื่องยนต์ เพราะเครื่องบินที่ถูกยิงโดนตรงนั้นคือลำที่ไม่ได้กลับมาให้นับ นี่คือการ "อคติจากผู้รอด (survivorship bias)" และในฐานะภาพประกอบของแนวคิดนี้ มันเกือบสมบูรณ์แบบ

EN

As history, it is largely invented. In 2016 the mathematician Bill Casselman went looking for the primary sources and found astonishingly little: no transcript of Wald delivering advice to a room of generals, no diagram of a bullet-riddled bomber with Wald's own hand on it, nothing resembling the vivid scene retold on a thousand websites. What survives are Wald's actual memoranda for the Statistical Research Group at Columbia, released decades later, and they contain something both less cinematic and more impressive: a technical apparatus for estimating, from the hit patterns on *surviving* planes alone, the probability that a plane hit n times would have gone down — without ever seeing the missing planes at all (Bill Casselman, "The Legend of Abraham Wald," AMS Feature Column, 2016; the original memoranda and the standard modern gloss are collected in Mangel & Samaniego, "Abraham Wald's Work on Aircraft Survivability," *Journal of the American Statistical Association* 79 (1984)). Wald assumed enemy fire struck aircraft roughly at random, then built a recursive equation linking the *observed* dis-

tribution of hits on survivors to the *unobserved* population of planes that never returned — solving, in effect, for the ghosts by their absence from the room. He never records a punchy one-liner about where to bolt the armor; the SRG's stated policy was to answer the technical question it was asked, not to issue tactical advice. The closest primary account, from SRG director W. Allen Wallis, states only that Wald "assumed, on good evidence, that hits in combat were uniformly distributed over the planes" and derived vulnerability estimates from that assumption — the seed, probably, from which the vivid modern retelling eventually grew.

TH

ในฐานะประวัติศาสตร์ เรื่องนี้ถูกแต่งขึ้นเสียเป็นส่วนใหญ่ ปี 2016 Bill Casselman นักคณิตศาสตร์ ไปตามหาแหล่งข้อมูลปฐมภูมิ และพบน้อยมาก: ไม่มีบันทึกคำต่อคำที่ Wald ให้คำแนะนำต่อหน้าห้องที่เต็มไปด้วยนายพล ไม่มีแผนภาพเครื่องบินที่ระเบิดพุนกระสุนที่มีลายมือของ Wald เองอยู่บนนั้น ไม่มีอะไรใกล้เคียงกับฉากมีสีสันที่เล่าซ้ำบนเว็บไซต์นับพัน สิ่งที่รอดมาคือบันทึกช่วยจำจริงของ Wald สำหรับ Statistical Research Group ที่ Columbia เปิดเผยหลายทศวรรษต่อมา ในนั้นมีบางอย่างที่ทำให้ดูหนังน้อยกว่าและน่าประทับใจกว่า: กลไกทางเทคนิคสำหรับประมาณจากรูปแบบรอยกระสุนบนเครื่องบินที่รอดกลับมาเท่านั้น ว่าเครื่องบินที่ถูกยิงโดน n ครั้งมีความน่าจะเป็นที่จะตกเท่าไร — โดยไม่เคยเห็นเครื่องบินที่หายไปเลยสักลำ (Bill Casselman, "The Legend of Abraham Wald," AMS Feature Column, 2016; บันทึกช่วยจำต้นฉบับและคำอธิบายมาตรฐานยุคปัจจุบันรวมอยู่ใน Mangel & Samaniego, "Abraham Wald's Work on Aircraft Survivability," Journal of the American Statistical Association 79 (1984)) Wald สมมติว่ากระสุนเข้าตีกระทบเครื่องบินอย่างสุ่มโดยประมาณ แล้วสร้างสมการเรียกซ้ำเชื่อมการแจกแจงรอยกระสุนที่สังเกตได้บนผู้รอด เข้ากับประชากรเครื่องบินที่ไม่เคยกลับมาซึ่งสังเกตไม่ได้ — เท่ากับแก้หาฝิจากการที่มันไม่อยู่ในห้อง เขาไม่เคยบันทึกการรศของสั้นๆ เรื่องจะขึ้นเพราะตรงไหนได้เลย นโยบายที่ SRG ประกาศไว้คือตอบคำถามทางเทคนิคที่ถูกถาม ไม่ออกคำแนะนำทางยุทธวิธี บันทึกปฐมภูมิที่ใกล้ที่สุด จาก W. Allen Wallis ผู้อำนวยการ SRG ระบุเพียงว่า Wald "สมมติบนหลักฐานที่ดีว่ารอยกระสุนในการกระจายอย่างสม่ำเสมอทั่วตัวเครื่องบิน" แล้วอนุมานค่าประมาณความเปราะบางจากสมมติฐานนั้น — น่าจะเป็นเมล็ดที่งอกเป็นการเล่าใหม่อันมีสีสันในยุคปัจจุบัน

EN

None of this makes the underlying insight less real, or less important to this book. The reframing at the heart of the Wald story — stop treating "the data I have" as "all the data there is," and ask what process filtered the data before it reached you — is exactly the kind of representation change this book is chasing. But the popular version has smoothed a subtle, careful piece of statistical reasoning into a punchy anecdote with a villain (the obtuse generals) and a hero delivering a mic-drop line. The real Wald is more interesting: a mathematician who looked at a table of surviving-aircraft hit-counts and, using nothing but the assumption that bullets don't discriminate, reconstructed a population he could never see. That is a longer path to a shorter insight, and it is the more instructive story of the two.

TH

ทั้งหมดนี้ไม่ได้ทำให้การหยั่งเห็น (insight) ที่อยู่ข้างใต้จึจึจึน้อยลง หรือสำคัญกับหนังสือเล่มนี้น้อยลง การเปลี่ยนกรอบ (reframe) ที่เป็นหัวใจของเรื่อง Wald — เลิกถือว่า "ข้อมูลที่มีอยู่ในมือ" คือ "ข้อมูลทั้งหมดที่มี" แล้วถามว่ากระบวนการอะไรกรองข้อมูลก่อนถึงมือ — คือการเปลี่ยนรูปแบบการแทนปัญหา (representation) ชนิดที่หนังสือเล่มนี้ไล่ล่าอยู่พอดี แต่เวอร์ชันยอดเยี่ยมของการให้เหตุผลทางสถิติที่ละเอียดและระมัดระวังขึ้นหนึ่งจนเรียบ เหลือเกร็ดสั้นกระชับที่มีตัวร้าย (นายพลหัวทึบ) และพระเอกที่ปล่อยวรรคทองปิดไมค์ Wald ตัวจริงน่าสนใจกว่า: นักคณิตศาสตร์ที่มองตารางนับรอยกระสุนบนเครื่องบินที่รอด ใช้เพียงสมมติฐานว่ากระสุนไม่เลือกที่ตก แล้วสร้างประชากรที่เขาไม่มีวันได้เห็นขึ้นมาใหม่ นั่นคือเส้นทางที่ยาวกว่าสู่ insight ที่สั้นกว่า และเป็นเรื่องที่สอนได้มากกว่าในสองเรื่องนี้

One Country's Plywood, Another Country's Menu

ไม้อัดของประเทศหนึ่ง เมนูอาหารของอีกประเทศ

EN

While Blackett's circus tuned depth charges, a young Leningrad mathematician named Leonid Kantorovich was solving a problem with no evident connection to war at all. In 1938 the local Plywood Trust, a Soviet industrial cooperative, brought him an ordinary-sounding scheduling headache: eight lathes, five kinds of veneer, and a demand to keep the *ratio* of the five plywood types fixed while maximizing total output. The trust's own engineers had been unable to find a satisfying answer by trial and adjustment.

TH

ปี 1938 Plywood Trust ในท้องถิ่น สหกรณ์อุตสาหกรรมโซเวียตแห่งหนึ่ง เอาปัญหามาจัดตารางที่ฟังดูธรรมดามาให้ Leonid Kantorovich นักคณิตศาสตร์หนุ่มจากเลนินกราด: เครื่องกลึงแปดเครื่อง ไม้วีเนียร์ห้าชนิด และข้อเรียกร้องให้คงอัตราส่วนของไม้อัดทั้งห้าชนิดไว้คงที่ พร้อมทำผลผลิตรวมให้มากที่สุด โจทย์นี้ไม่มีความเกี่ยวข้องกับสงครามให้เห็นเลย ขณะที่คณะละครสัตว์ของ Blackett จูกระเบิดลี้ภัยอยู่ วิศวกรของทรัสต์เองลองและปรับแล้วก็หาคำตอบที่น่าพอใจไม่ได้

EN

Kantorovich noticed something that turned out to matter far beyond plywood: the problem was not really about lathes. Any situation where you must allocate a fixed pool of resources among competing uses, subject to a straight-line (linear) relationship between inputs and outputs, has the same skeleton, whatever the resources happen to be — labor hours, tons of steel, hectares of farmland, machine-time. In 1939 he published *The Mathematical Method of Production Planning and Organization*, showing that an enormous class of allocation problems could be written as one equation: maximize (or minimize) a linear function of many variables, subject to linear constraints (MacTutor History of Mathematics, biography of Kantorovich). He called the technique linear programming before that name existed in the West, and he had no map of how to solve the general version efficiently — only the recognition that the general version existed and that Soviet industry was riddled with disguised copies of it. Decades later, in 1975, he shared the Nobel Memorial Prize in Economic Sciences for exactly this

insight (Britannica, "Leonid Vitalyevich Kantorovich") — a Soviet mathematician recognized by a Western economics prize for having spotted, in a plywood factory, the shape of a problem that would eventually run supply chains on both sides of the Iron Curtain.

TH

Kantorovich เห็นว่าโจทย์นี้ไม่ได้เกี่ยวกับเครื่องกลึงจริงๆ และข้อสังเกตนี้สำคัญไกลเกินไม่อด สถานการณ์ใดก็ตามที่ต้องจัดสรรทรัพยากรกองหนึ่งที่มีจำกัดให้การใช้งานที่แข่งกันอยู่ ภายใต้ความสัมพันธ์แบบเส้นตรง (linear) ระหว่างปัจจัยเข้ากับผลผลิต มีโครงสร้างเหมือนกันหมด ไม่ว่าทรัพยากรจะเป็นชั่วโมงแรงงาน ต้นของเหล็ก เhekตาร์ของที่นา หรือเวลาเครื่องจักร ปี 1939 เขาตีพิมพ์ The Mathematical Method of Production Planning and Organization แสดงว่าโจทย์การจัดสรรกลุ่มใหญ่มาเขียนเป็นสมการเดียวได้: หาค่าสูงสุด (หรือต่ำสุด) ของฟังก์ชันเชิงเส้นของตัวแปรจำนวนมาก ภายใต้ข้อจำกัดเชิงเส้น (MacTutor History of Mathematics, biography of Kantorovich) เขาเรียกเทคนิคนี้ว่ากำหนดการเชิงเส้น (linear programming) ก่อนที่ชื่อนี้จะมีในโลกตะวันตก เขาไม่มีแผนที่ว่าจะแก้เวอร์ชันทั่วไปอย่างมีประสิทธิภาพได้อย่างไร — มีเพียงการมองเห็นว่าเวอร์ชันทั่วไปนั้นมีอยู่ และอุตสาหกรรมโซเวียตเต็มไปด้วยสำเนาปลอมตัวของมัน หลายทศวรรษต่อมา ปี 1975 เขาร่วมรับรางวัลโนเบลสาขาเศรษฐศาสตร์จากการยังเห็นข้อนี้พอดี (Britannica, "Leonid Vitalyevich Kantorovich") — นักคณิตศาสตร์โซเวียตที่รางวัลเศรษฐศาสตร์ของตะวันตกยอมรับ เพราะมองเห็นในโรงงานไม้อัดรูปทรงของโจทย์ที่สุดท้ายจะเดินห่วงโซ่อุปทานทั้งสองฝั่งมานเหล็ก

EN

An ocean away and unaware of Kantorovich's work, the American economist George Stigler ran into the same shape of problem from a completely different door. In 1945 he asked how a moderately active man could eat as cheaply as possible while still meeting the U.S. National Research Council's recommended intake of nine nutrients, choosing freely among 77 available foods. No known method could find the true cheapest diet outright, so Stigler searched by hand among a limited set of promising combinations and reported a diet — wheat flour, evaporated milk, cabbage, spinach, dried navy beans — costing \$39.93 per year in 1939 prices (Wikipedia, "Stigler

diet"). He knew he had found a good answer, not necessarily the best one, and he said so.

TH

George Stigler นักเศรษฐศาสตร์อเมริกัน เดินทางโจทย์รูปทรงเดียวกันจากประตูคนละบาน อีกฟากมหาสมุทรและไม่รู้เรื่องงานของ Kantorovich ปี 1945 เขาถามว่าชายที่กระฉับกระเฉงพอประมาณคนหนึ่งจะกินให้ถูกที่สุดได้อย่างไร โดยยังได้สารอาหารโภชนาการตาม U.S. National Research Council แนะนำ และเลือกได้อิสระจากอาหาร 77 รายการ ไม่มีวิธีที่รู้จักกันตอนนั้นหาชุดอาหารที่ถูกที่สุดจริงได้ตรงๆ Stigler จึงค้นด้วยมือในชุดผสมที่ดูมีแนวโน้มจำกัด แล้วรายงานอาหารชุดหนึ่ง — แป้งสาลี นมข้นระเหย กะหล่ำปลี ผักโขม ถั่วขาวแห้ง — ราคา 39.93 ดอลลาร์ต่อปีตามราคาปี 1939 (Wikipedia, "Stigler diet") เขารู้ว่าเจอคำตอบที่ดีชุดหนึ่ง ไม่จำเป็นต้องเป็นชุดที่ดีที่สุด และเขาพูดออกมาตรงๆ

The Method That Ate All the Problems

วิธีที่กลืนโจทย์ทั้งหมด

EN

Two years later, an American mathematician named George Dantzig gave Stigler's diet problem, and every problem shaped like it, a general solution — and the way he arrived there has its own well-documented origin story, separate from the one that would later attach to his name.

TH

สองปีต่อมา George Dantzig นักคณิตศาสตร์อเมริกัน ให้คำตอบทั่วไปแก่โจทย์อาหารของ Stigler และทุกโจทย์ที่มีรูปทรงเดียวกัน เส้นทางที่เขาไปถึงจุดนั้นมีเรื่องต้นกำเนิดที่มีหลักฐานดีของตัวเอง แยกจากเรื่องที่ภายหลังจะมาติดกับชื่อเขา

EN

Dantzig's simplex method was born of wartime bureaucracy: as a young statistician for the U.S. Army Air Forces, he had grown expert at the mechanics of logistics planning done by hand, on desk calculators, one plan at a time. In June 1947 he formulated a general model for these planning problems as a system of linear inequalities, and by that August had devised the simplex algorithm to solve it — a way of walking, systematically and effi-

ciently, along the edges of the geometric shape formed by a problem's constraints until an optimal corner was reached (George Dantzig, "Origins of the Simplex Method," ACM, 1990; MacTutor biography of Dantzig). Within months, colleagues at RAND had renamed his subject "linear programming," and by 1951 the method was public.

TH

วิธีซิมเพล็กซ์ของ Dantzig เกิดจากระบบราชการยามสงคราม เขาเป็นนักสถิติหนุ่มของกองทัพอากาศบกสหรัฐฯ และชำนาญกลไกของการวางแผนโลจิสติกส์ที่ทำด้วยมือบนเครื่องคิดเลขตั้งโต๊ะ ที่ละแผ่น มิถุนายน 1947 เขาสร้างแบบจำลองทั่วไปสำหรับโจทย์วางแผนพวกนี้เป็นระบบสมการเชิงเส้น สิงหาคมปีนั้นเขาคิดอัลกอริทึมซิมเพล็กซ์ขึ้นมาแถม — เดินอย่างเป็นระบบและมีประสิทธิภาพไปตามขอบของรูปทรงเรขาคณิตที่ข้อจำกัดของโจทย์ก่อกำหนด จนถึงมุมที่เหมาะสมที่สุด (George Dantzig, "Origins of the Simplex Method," ACM, 1990; MacTutor biography of Dantzig) ภายในไม่กี่เดือน เพื่อนร่วมงานที่ RAND เปลี่ยนชื่อสาขาของเขาเป็น "กำหนดการเชิงเส้น" ("linear programming") และปี 1951 วิธีนี้ก็เผยแพร่สู่สาธารณะ

EN

That same fall, a researcher named Jack Laderman at the National Bureau of Standards decided to stress-test the brand-new algorithm on the hardest real instance he could find: Stigler's diet problem, nine nutritional constraints across seventy-seven foods. Using hand-cranked desk calculators — this was still years before general-purpose electronic computers were commonplace — a team spent roughly 120 person-days grinding through the simplex method's arithmetic by hand. The true minimum-cost diet, it turned out, cost only about 24 cents less per year than Stigler's hand-searched estimate (Wikipedia, "Stigler diet"; George Dantzig, INFORMS oral history). Stigler's instincts, honed by an economist's judgment, had gotten within a rounding error of optimal. But the simplex method had gotten there by a procedure, one that would work whether the next problem was a diet, a plywood mill, or a bombing campaign, and would keep working as the number of variables climbed into the thousands, then the millions.

T H

ชุดอาหารต้นทุนต่ำสุดจริงถูกกว่าค่าประมาณที่ Stigler คำนวณด้วยมือเพียงราว 24 เซนต์ต่อปี ฤดูใบไม้ร่วงปีเดียวกันนั้น Jack Laderman นักวิจัยที่ National Bureau of Standards ตัดสินใจทดสอบความทนของอัลกอริทึมใหม่เอี่ยมนี้กับอินสแตนซ์จริงที่ยากที่สุดเท่าที่หาได้: โจทย์อาหารของ Stigler ข้อจำกัดทางโภชนาการเก้าข้อบนอาหารเจ็ดสิบเจ็ดรายการ ทีมหนึ่งใช้เวลาราว 120 คน-วันบนเดสคอปของวิธีซิมเพล็กซ์ด้วยเครื่องคิดเลขตั้งโต๊ะแบบหมุนมือ — ตอนนั้นยังอีกหลายปีก่อนคอมพิวเตอร์อิเล็กทรอนิกส์จะมาเป็นของธรรมดา — และได้ตัวเลขนั้นออกมา (Wikipedia, "Stigler diet"; George Dantzig, INFORMS oral history) สัญชาตญาณของ Stigler ที่ลบล้างวิจารณ์ของนักเศรษฐศาสตร์ เข้าใกล้ค่าเหมาะที่สุดจนต่างกันแค่เศษปัดเศษ แต่วิธีซิมเพล็กซ์ไปถึงตรงนั้นด้วยขั้นตอนวิธีขั้นตอนที่ใช้ได้ไม่ว่าโจทย์ถัดไปจะเป็นอาหาร โรงไม้อัด หรือปฏิบัติการทิ้งระเบิด และยังใช้ได้เมื่อจำนวนตัวแปรได้ขึ้นถึงหลักพัน แล้วหลักล้าน

E N

Dantzig's name is also attached to a second, older, and unrelated anecdote, which deserves to be told with its correct date rather than folded into the simplex story: in 1939, as a first-year doctoral student in statistics at Berkeley, Dantzig arrived late to a lecture by Jerzy Neyman and copied down two problems chalked on the blackboard, assuming they were the week's homework. He found them harder than usual but turned in solutions a few days later. They were, in fact, two long-unsolved problems in mathematical statistics, and Dantzig — not knowing this, and so never discouraged from trying — had solved both. The core of the story checks out; the widely circulated embellishments (a ten-problem final exam, a claim that "even Einstein" had failed the problems) do not, and were traced to a later, dramatized retelling by a television preacher rather than to Dantzig himself (Snopes, "The Legend of the 'Unsolvable Math Problem'"). The lesson Dantzig himself drew from it — that not knowing a problem is supposed to be impossible is sometimes the only reason you solve it — belongs to this book's thesis in its own right: the frame around a problem, including the frame of *how hard people believe it to be*, is part of what makes the space large or small.

T H

ชื่อของ Dantzig ยังติดอยู่กับเกร็ดอีกเรื่องที่น่าสนใจและไม่เกี่ยวข้องกัน เรื่องนี้สมควรเล่าพร้อมวันที่ถูกต้อง ไม่ใช่พักรวมเข้ากับเรื่องซิมเพล็กซ์ ปี 1939 Dantzig เป็นนักศึกษาปริญญาเอกปีหนึ่งสาขาสถิติที่ Berkeley เขามาสายในคาบบรรยายของ Jerzy Neyman ลอกโจทย์สองข้อที่เขียนด้วยชอล์กไว้บนกระดานดำลงมา และเข้าใจว่าเป็นการบ้านประจำสัปดาห์ เขาพบว่ามันยากกว่าปกติ แต่ก็ส่งคำตอบไม่กี่วันถัดมา จริ่งแล้วมันคือโจทย์สองข้อในสถิติเชิงคณิตศาสตร์ที่ไม่มีใครแก้ได้มานาน Dantzig ไม่รู้เรื่องนี้ จึงไม่เคยถูกทำให้ท้อที่จะลอง และแก้ได้ทั้งสองข้อ แก่นของเรื่องตรวจสอบแล้วเป็นจริง ส่วนรายละเอียดประดับที่แพร่หลายกันมาก (ข้อสอบไล่สิบข้อ คำอ้างว่า "แม้แต่ Einstein" ก็ทำโจทย์พวกนี้ไม่ได้) ไม่จริง และตามรอยไปพบว่ามาจากการเล่าใหม่แบบเสริมดราม่าในภายหลังของนักเทศน์ทางโทรทัศน์คนหนึ่ง ไม่ใช่จากตัว Dantzig เอง (Snopes, "The Legend of the 'Unsolvable Math Problem'") บทเรียนที่ Dantzig ดึงออกมาเอง — การไม่รู้ว่าโจทย์หนึ่งควรจะเป็นไปไม่ได้ บางครั้งคือเหตุผลเดียวที่แก้มันได้ — เป็นของแก่นความคิดในหนังสือเล่มนี้ในสิทธิ์ของมันเอง: กรอบที่ล้อมโจทย์อยู่ รวมถึงกรอบว่าคนเชื่อกันว่ามันยากแค่ไหน เป็นส่วนหนึ่งของสิ่งที่ทำให้พื้นที่ใหญ่หรือเล็ก

A Name to Hide the Math

ชื่อที่ตั้งไว้ซ่อนคณิตศาสตร์

E N

The most quietly radical idea to come out of the RAND Corporation in this decade belonged to Richard Bellman, and it began, by Bellman's own account, as an act of camouflage.

T H

ไอเดียที่สุดโต่งที่สุดอย่างเงียบๆ จาก RAND Corporation ในทศวรรษนี้เป็นของ Richard Bellman และตามคำบอกเล่าของ Bellman เอง มันเริ่มต้นในฐานะการพรางตัว

E N

Bellman arrived at RAND in 1949 already restless in a conventional academic career, and by 1950 was working on what he called multistage decision pro-

cesses: problems where a sequence of choices, made one at a time, has to add up to a good overall outcome — how to allocate an investment across years, how to route a process through successive stages, how to plan a sequence of moves toward a goal. His key mathematical insight, later formalized as the *principle of optimality*, is easy to state once seen: an optimal overall plan must also be optimal for every sub-plan it contains, starting from wherever you happen to be partway through. If the last leg of your best possible road trip from Bangkok to Chiang Mai runs through Lampang, then the best route from Lampang onward to Chiang Mai must itself be optimal — because if a better route from Lampang existed, you could substitute it and improve the whole trip. This lets a solver break an enormous multistage problem into a chain of small ones, solved backward from the goal, each one easy once its successor is already known. Bellman called the technique dynamic programming, and the functional equations built on the principle of optimality remain, to this day, one of the handful of general-purpose engines for solving sequential decision problems in economics, engineering, and — chapter 10 will show — reinforcement learning.

TH

แผนโดยรวมที่เหมาะสมที่สุดต้องเหมาะสมที่สุดสำหรับแผนย่อยทุกแผนที่มันบรรจุอยู่ด้วย เริ่มนับจากตรงไหนก็ตามที่บังเอิญยืนอยู่กลางทาง — นี่คือการหยั่งเห็นทางคณิตศาสตร์ข้อสำคัญของ Bellman ภายหลังถูกทำให้เป็นรูปนัยในชื่อ หลักการเหมาะสมที่สุด (principle of optimality) และพูดออกมาง่ายเมื่อเห็นแล้ว Bellman มาถึง RAND ในปี 1949 กระสับกระส่ายกับเส้นทางวิชาการแบบเดิมอยู่แล้ว และพอถึงปี 1950 ก็ทำงานกับสิ่งที่เขาเรียกว่ากระบวนการตัดสินใจหลายชั้น: โจทย์ที่ลำดับของทางเลือกซึ่งเลือกทีละอย่างต้องรวมกันเป็นผลลัพธ์โดยรวมที่ดี — จัดสรรการลงทุนข้ามปีอย่างไร เดินกระบวนการผ่านขั้นตอนที่ต่อกันอย่างไร วางลำดับของท่าไปสูเป้าหมายอย่างไร ถ้าช่วงสุดท้ายของทริปถนนที่ดีที่สุดเท่าที่เป็นไปได้จากกรุงเทพฯ ไปเชียงใหม่ผ่านลำปาง เส้นทางที่ดีที่สุดจากลำปางต่อไปเชียงใหม่ก็ต้องเหมาะสมที่สุดในตัวมันเอง — ถ้ามีเส้นทางที่ดีกว่าจากลำปางอยู่ ก็เอามาแทนแล้วทั้งทริปดีขึ้น สิ่งนี้ทำให้นักแก้ปัญหาแตกโจทย์หลายชั้นขนาดใหญ่เป็นโซ่ของโจทย์เล็กๆ แก่ย้อนกลับจากเป้าหมาย แต่ละข้อง่ายเมื่อรู้คำตอบของข้อที่ตามมาแล้ว Bellman เรียกเทคนิคนี้ว่ากำหนดการพลวัต (dynamic programming) และสมการเชิงฟังก์ชันที่

สร้างบนหลักการเหมาะสมที่สุดยังคงเป็นหนึ่งในเครื่องยนต์อเนกประสงค์ไม่กี่ตัวสำหรับ
แก้โจทย์การตัดสินใจแบบลำดับในเศรษฐศาสตร์ วิศวกรรม และ — บทที่ 10 จะแสดง
ให้เห็น — การเรียนรู้แบบเสริมกำลัง มาจนถึงทุกวันนี้

E N

The name itself, though, is where Bellman's own account gets interesting to hold up against the calendar. In a 1984 autobiography excerpted by Stuart Dreyfus, Bellman explained that he needed a label that would not sound like "mathematical research," because RAND's ultimate boss at the Department of Defense — a "Secretary Wilson" with, in Bellman's words, "a pathological fear and hatred of the word research" — could not be allowed to suspect he was funding pure mathematics. So Bellman chose *programming*, a word with an unthreatening, practical ring, and *dynamic*, an adjective he judged impossible to use pejoratively, and combined them into an umbrella "not even a Congressman could object to" (Stuart Dreyfus, "Richard Bellman on the Birth of Dynamic Programming," *Operations Research* 50, no. 1 (2002), quoting Bellman's *Eye of the Hurricane*, 1984). It is a wonderful story about a bureaucrat too literal-minded to tolerate the word "research." There is one problem with the timeline: Bellman's own first published paper using the term "dynamic programming" appeared in the *Proceedings of the National Academy of Sciences* in August 1952 (Bellman, "On the Theory of Dynamic Programming," *PNAS* 38, no. 8 (1952)) — while Charles Wilson, the defense secretary Bellman names, did not take that office until January 1953. Bloggers and historians tracking the discrepancy have noted this is at least a year out of order for the story as told (see the discussion at Arcane Sentiment, "Why 'dynamic programming'?"). Bellman may simply have misremembered which official prompted his caution, decades after the fact, or folded two memories into one — autobiographies are not sworn depositions, and his own memoir elsewhere warns, with disarming self-awareness, that "all this contributes to the misleading nature of conventional history." What is not in question is the mathematics: the principle of optimality is real, powerful, and exactly as general as Bellman claimed.

TH

เส้นเวลาไม่ลงรอยกับคำบอกเล่าของ Bellman เรื่องตัวชื่อ ในอัตชีวประวัติปี 1984 ที่ Stuart Dreyfus คัดมาบางส่วน Bellman อธิบายว่าเขาต้องการป้ายชื่อที่ไม่ฟังดูเหมือน "การวิจัยทางคณิตศาสตร์" เจ้านายสูงสุดของ RAND ที่กระทรวงกลาโหม — "รัฐมนตรี Wilson" คนหนึ่ง ซึ่งในคำของ Bellman "กลัวและเกลียดคำว่า research อย่างเป็นโรค" — ต้องไม่ถูกปล่อยให้สงสัยว่าเขากำลังจ่ายเงินสนับสนุนคณิตศาสตร์บริสุทธิ์ Bellman จึงเลือกคำว่า programming คำที่ให้เสียงปฏิบัติจริงและไม่คุกคามกับ dynamic คำคุณศัพท์ที่เขาตัดสินใจว่าใช้ในแง่ลบไม่ได้ แล้วรวมเป็นรุ่มคำที่ "แม้แต่สมาชิกสภาองเกรสก็ค้านไม่ได้" (Stuart Dreyfus, "Richard Bellman on the Birth of Dynamic Programming," *Operations Research* 50, no. 1 (2002) อ้าง Eye of the Hurricane ของ Bellman ปี 1984) เป็นเรื่องที่พิเศษมากเกี่ยวกับข้าราชการที่ตีความตรงตัวเกินกว่าจะทนคำว่า "research" ได้ ปัญหาอยู่ที่ปฏิทิน: บทความตีพิมพ์ชิ้นแรกของ Bellman เองที่ใช้คำว่า "dynamic programming" ปรากฏใน *Proceedings of the National Academy of Sciences* เดือนสิงหาคม 1952 (Bellman, "On the Theory of Dynamic Programming," *PNAS* 38, no. 8 (1952)) — ขณะที่ Charles Wilson รัฐมนตรีกลาโหมที่ Bellman เอย์ชื่อ ยังไม่เข้ารับตำแหน่งจนกระทั่งมกราคม 1953 บล็อกเกอร์และนักประวัติศาสตร์ที่ตามความไม่ลงรอยนี้ตั้งข้อสังเกตว่าเรื่องตามที่เล่ากันมาคลาดเคลื่อนไปอย่างน้อยหนึ่งปี (see the discussion at *Arcane Sentiment*, "Why 'dynamic programming'?") Bellman อาจแค่จำผิดว่าเจ้าหน้าที่คนไหนทำให้เขาระวังตัว หลังเหตุการณ์ผ่านไปหลายทศวรรษ หรืออาจพบความทรงจำสองชิ้นเป็นขึ้นเดียว — อัตชีวประวัติไม่ใช่คำให้การได้คำสาบาน และบันทึกความทรงจำของเขาเองก็เตือนไว้ที่อื่นด้วยความรู้ตัวอย่างน่าเอ็นดูว่า "ทั้งหมดนี้มีส่วนทำให้ประวัติศาสตร์แบบที่เล่ากันมามีลักษณะชวนเข้าใจผิด" สิ่งที่ไม่ใช่ข้อสงสัยคือคณิตศาสตร์: หลักการเหมาะที่สุดนั้นจริง ทรงพลัง และทั่วไปเท่าที่ Bellman อ้างไว้พอดี

When Searching Is Hopeless, Gamble

เมื่อการค้นหาสิ้นหวัง ก็เล่นพนัน

EN

Not every wartime problem yielded to an equation. Some problems were so tangled with chance — a neutron careening through a chunk of fissile material, colliding, scattering, occasionally triggering another collision — that no formula could capture their behavior in closed form, and no exhaustive search was remotely possible even for a room-sized calculating machine.

TH

ไม่ใช่ทุกโจทย์ยามสงครามจะยอมแพ้ให้สมการ บางโจทย์พันกับความบังเอิญแน่นอนเกินไป — นิวตรอนพุ่งผ่านก้อนวัสดุฟิสไซล์ ชน กระเจิง และบางครั้งจุดชนวนการชนอีกครั้ง — จนไม่มีสูตรไหนจับพฤติกรรมของมันในรูปปิดได้ และค้นหาให้ครบถ้วนก็เป็นไปไม่ได้เลย แม้กับเครื่องคำนวณขนาดเท่าห้อง

EN

The way out came, by Stanislaw Ulam's own account, from a card game. Recovering from an illness in 1946 and playing endless hands of Canfield solitaire, Ulam found himself wondering what fraction of shuffled decks could actually be won — and realized that after failing to derive the answer by combinatorial analysis, the fastest path to a good estimate was simply to lay out many hands and count. If you cannot calculate the probability of an event from first principles, you can often *simulate* the event many times and let frequency stand in for probability. Ulam brought the idea to John von Neumann, who saw immediately that the same trick — replace an intractable calculation with a large number of random trials — could model how neutrons diffuse through a bomb core, a problem central to the hydrogen bomb project at Los Alamos. Nicholas Metropolis, working alongside them, gave the resulting family of techniques its lasting name, after the Monaco casino district, in a nod to Ulam's uncle, who reputedly could never resist a trip there (Los Alamos National Laboratory, "Hitting the Jackpot: The Birth of the Monte Carlo Method").

TH

ทางออกมาจากเกมไพ่ ตามคำบอกเล่าของ Stanislaw Ulam เอง ปี 1946 เขาพักฟื้นจากการเจ็บป่วยและเล่นไพ่คนเดียวแบบ Canfield ไม่รู้จบ Ulam สงสัยว่าสำหรับที่สับแล้วมีสัดส่วนเท่าไรที่ชนะได้จริง อนุมานด้วยการวิเคราะห์เชิงการจัดหมู่ไม่สำเร็จ เขาจึงเห็นว่าเส้นทางเร็วที่สุดสู่ค่าประมาณที่ดีคือวางไพ่หลายๆ ตาแล้วนับเอา ถ้าคำนวณความน่าจะเป็นของเหตุการณ์จากหลักการพื้นฐานไม่ได้ ก็มักจะจำลองเหตุการณ์นั้นหลายๆ ครั้ง แล้วให้ความถี่แทนความน่าจะเป็นได้ Ulam เอาไอเดียนี้ไปหา John von Neumann ผู้เห็นทันทีว่าลูกไม่เดียวกัน — แทนการคำนวณที่จัดการไม่ไหวด้วยการทดลองสุ่มจำนวนมาก — จำลองการแพร่ของนิวตรอนผ่านแกนระเบิดได้ โจทย์นี้เป็นหัวใจของโครงการระเบิดไฮโดรเจนที่ Los Alamos Nicholas Metropolis ที่ทำงานอยู่ข้างๆ พวกเขา ตั้งชื่อที่อยู่ยืนยันให้เทคนิคตระกูลนี้ตามย่านคาสีโนของโมนาโก เพื่อพยักถึงลุงของ Ulam ที่ว่ากันว่าอดใจไม่ไปที่นั่นไม่เคยได้ (Los Alamos National Laboratory, "Hitting the Jackpot: The Birth of the Monte Carlo Method")

EN

Monte Carlo methods are, in the vocabulary of this book, a confession and a workaround at once: a confession that the space is too large or too tangled to search or compute exactly, and a workaround that trades a guaranteed exact answer for a probable good one, obtained fast. It is the same bet a pollster makes surveying a thousand voters instead of forty million, applied to physics instead of politics — and it would go on to become one of the standard tools of modern computing, invoked whenever a problem's exact structure resists analysis but its *outcomes* can be cheaply sampled.

TH

วิธีมอนติคาร์โล (Monte Carlo) คือคำสารภาพและทางเลี้ยวในเวลาเดียวกัน ในคำศัพท์ของหนังสือเล่มนี้ สารภาพว่าพื้นที่ใหญ่เกินไปหรือพันกันเกินกว่าจะค้นหาคำนวณอย่างแม่นยำได้ และเลี้ยวด้วยการแลกคำตอบแม่นยำที่รับประกันได้ กับคำตอบที่น่าจะดีและได้มาเร็ว มันคือการเติมพันแบบเดียวกับที่นักสำรวจความคิดเห็นทำเมื่อสอบถามผู้มีสิทธิ์เลือกตั้งหนึ่งพันคนแทนที่จะเป็นสี่สิบล้านคน เพียงแต่ใช้กับฟิสิกส์แทนการเมือง — และมันจะกลายเป็นเครื่องมือมาตรฐานชิ้นหนึ่งของการคำนวณสมัยใหม่ ถูกเรียกใช้ทุกครั้งที่โครงสร้างที่แน่นอนของโจทย์ด้านการวิเคราะห์แต่ผลลัพธ์ของมันสุ่มตัวอย่างได้ในราคาถูก

One Problem, Many Costumes

โจทย์เดียว หลายเครื่องแต่งกาย

EN

Step back from the individual stories and a single, almost eerie pattern comes into view. A team of physicists calculating depth-charge fuses. A statistician reconstructing invisible aircraft losses. A Soviet mathematician scheduling plywood lathes. An American economist pricing a subsistence diet. A young Berkeley graduate student solving problems he didn't know were famous. A RAND consultant hiding calculus inside a politically safe name. A convalescent playing solitaire. None of these people were working on "the same problem" in any sense a layperson would recognize. And yet Kantorovich's lathes, Stigler's groceries, and Dantzig's Army logistics turned out to be one mathematical object — a linear program — in three disguises; Bellman showed that an enormous family of sequential decisions, from in-

vestment schedules to shortest routes, share a single recursive skeleton; and Wald's and Ulam's contributions both amount to extracting a signal from a process too large or too obscured to observe directly.

TH

รูปแบบเดียวปรากฏขึ้นเมื่อถอยออกจากเรื่องเล่าแต่ละเรื่อง และมันเกือบจะน่าขนลุกที่นักฟิสิกส์คำนวณชนวนระเบิดเล็ก นักสถิติสร้างความสูญเสียของเครื่องบินที่มองไม่เห็นขึ้นมาใหม่ นักคณิตศาสตร์โซเวียตจัดตารางเครื่องกลิ้งไม้อัด นักเศรษฐศาสตร์อเมริกันตั้งราคาอาหารประทังชีวิต นักศึกษาระดับบัณฑิตศึกษาหนุ่มที่ Berkeley แก่โจทย์ที่เขาไม่รู้ว่าจะต้องทำที่ปรึกษาของ RAND ชอนแคลคูลัสไว้ในชื่อที่ปลอดภัยทางการเมือง คนใช้พักผ่อนเล่นไฟคนเดียว ไม่มีใครในนี้ทำงานกับ "โจทย์เดียวกัน" ในความหมายที่คนทั่วไปจะรับรู้ได้เลย กระนั้น เครื่องกลิ้งของ Kantorovich ของข้าของ Stigler และโลจิสติกส์กองทัพบกของ Dantzig กลับเป็นวัตถุทางคณิตศาสตร์ขึ้นเดียว — กำหนดการเชิงเส้น — ในสามชุดปลอมตัว Bellman แสดงว่าการตัดสินใจแบบลำดับตระกูลใหญ่ ตั้งแต่ตารางการลงทุนถึงเส้นทางที่สั้นที่สุด ใช้โครงกระดูกเรียกซ้ำเดียวกัน และสิ่งที่ Wald กับ Ulam ให้ไว้ล้วนเท่ากับศักดิ์สัญญาออกจากกระบวนการที่ใหญ่หรือถูกบังเกินกว่าจะสังเกตตรงๆ ได้

EN

This is the discovery this chapter's decade made, and it is the one that turns problem-solving from a craft into an engineering discipline: the *representation* is portable. Once you have proven that resource allocation, diet planning, and factory scheduling are the same equation, you do not need a new theory for each one — you need one solver, applied three times. This is also why the unit of optimization stopped being the individual weapon, the individual convoy, or the individual factory line, and became the *system*: the whole convoy fleet, the whole depth-charge doctrine, the whole supply chain. Optimizing a single ship's speed is a local trick; optimizing the convoy geometry is a system-level insight that saves an ocean's worth of tonnage. The operational researchers of the 1940s were the first people paid, as a full-time job, to ask "what is the actual shape of this problem?" before touching a single lever — and the answer, again and again, was that the shape was smaller and more general than anyone doing the job by feel had ever suspected.

T H

representation พกพาได้ — นี่คือการค้นพบของทศวรรษในบั้นนี้ และเป็นสิ่งที่เปลี่ยนการแก้ปัญหาจากงานฝีมือให้เป็นสาขาวิศวกรรม พิสูจน์แล้วว่าการจัดสรรทรัพยากร การวางแผนอาหาร และการจัดตารางโรงงานคือสมการเดียวกัน ก็ไม่ต้องมีทฤษฎีใหม่สำหรับแต่ละอย่าง — ต้องการตัวแก้ตัวเดียว ใช้สามครั้ง นี่ยังเป็นเหตุผลที่หน่วยของการปรับให้เหมาะสมที่สุดเล็กเป็นอาวุธชิ้นเดียว กองเรือชุดเดียว หรือสายการผลิตสายเดียว และกลายเป็นระบบ: กองเรือทั้งกองเรือ หลักนิยมระเบิดลึกทั้งหลักนิยม ห่วงโซ่อุปทานทั้งห่วงโซ่ ปรับความเร็วของเรือลำเดียวให้เหมาะสมที่สุดคือลูกไม้เฉพาะถิ่น ปรับเรขาคณิตของกองเรือให้เหมาะสมที่สุดคือ insight ระดับระบบที่ช่วยระวางบรรทุกไว้ได้ทั้งมหาสมุทร นักวิจัยดำเนินงาน (operational research) แห่งทศวรรษ 1940 คือคนกลุ่มแรกที่ได้เงินเดือนเต็มเวลาเพื่อถามว่า "รูปทรงจริงของโจทย์นี้คืออะไร" ก่อนจะแตะคันโยกสักคัน — และคำตอบ ครั้งแล้วครั้งเล่า คือรูปทรงนั้นเล็กกว่าและทั่วไปกว่าที่ใครก็ตามที่ทำงานด้วยความรู้สึกเคยสงสัย

The Move

ท่าประจำบท

E N

Move 04 — Optimize the whole system, and measure before you optimize.

T H

Move 04 — ปรับทั้งระบบ และวัดก่อนปรับ

E N

- **Draw the boundary wide before you draw a single conclusion.** Blackett's team did not ask "how do we protect this ship" — they asked "how do we protect this fleet," and the right answer to the wider question (larger convoys) was the opposite of the right answer to the narrower one. Whatever unit you are tempted to optimize — a task, a ship, a plane — ask whether the real unit is one level up.
- **Distrust an intuition until you have measured the process that generated your data.** Wald's real insight was not a punchline about armor placement; it was refusing to treat "the planes I can see" as "all the planes there are." Before optimizing anything, ask what filtered your sample.

- **Look for the linear program, or the recursive equation, hiding inside your problem.** Kantorovich's lathes, Stigler's groceries, and Dantzig's logistics were one equation wearing three coats. When a problem feels bespoke, check whether a known general solver already fits it exactly.
- **When exact search is hopeless, sample instead of guessing.** Ulam's solitaire hands and Metropolis's neutron simulations turned an unanswerable calculation into a countable experiment. A large number of honest trials beats a single confident guess.

TH

- **ลากขอบเขตให้กว้างก่อนจะลากข้อสรุปสักข้อ** ทีมของ Blackett ไม่ได้ถามว่า "จะปกป้องเรือลำนี้อย่างไร" — พวกเขาถามว่า "จะปกป้องกองเรือนี้อย่างไร" และคำตอบที่ถูกของคำถามที่กว้างกว่า (กองเรือใหญ่ขึ้น) คือตรงข้ามกับคำตอบที่ถูกของคำถามที่แคบกว่า ไม่ว่าจะถูกล่อให้ปรับหน่วยไหนให้เหมาะสมที่สุด — งานหนึ่งชิ้น เรือหนึ่งลำ เครื่องบินหนึ่งเครื่อง — ให้ถามว่าหน่วยจริงอยู่สูงขึ้นไปอีกชั้นหรือเปล่า
- **อย่าเชื่อสัญชาตญาณจนกว่าจะวัดกระบวนการที่ผลิตข้อมูลของคุณแล้ว** insight จริงของ Wald ไม่ใช่วรรคทองเรื่องตำแหน่งติดเกราะ มันคือการปฏิเสธที่จะถือว่า "เครื่องบินที่มองเห็นได้" คือ "เครื่องบินทั้งหมดที่มี" ก่อนจะปรับอะไรให้เหมาะสมที่สุด ให้ถามว่าอะไรรองรับตัวอย่างมา
- **มองหากำหนดการเชิงเส้น หรือสมการเรียกซ้ำ ที่ซ่อนอยู่ในโจทย์** เครื่องกลึงของ Kantorovich ของซ้ำของ Stigler และโลจิสติกส์ของ Dantzig คือสมการเดียวที่สวมเสื้อสามตัว เมื่อโจทย์รู้สึกเหมือนถูกตัดเย็บมาเฉพาะ ให้ตรวจว่ามีตัวแก้ทั่วไปที่รู้จักกันอยู่แล้วสวมพอดีกับมันหรือไม่
- **เมื่อการค้นหาแบบแม่นยำสิ้นหวัง ให้สุ่มตัวอย่างแทนการเดา** ไฟคนเดียวของ Ulam และการจำลองนิวตรอนของ Metropolis เปลี่ยนการคำนวณที่ตอบไม่ได้ให้เป็นการทดลองที่นับได้ การทดลองอย่างซื่อสัตย์จำนวนมากชนะการเดาอย่างมั่นใจครั้งเดียว

EN

Next — Chapter 05 · The Algorithmic Path: the search that operations research did with a slide rule, a new generation would do with a machine — and the first machines that searched would need a theory of their own.

TH

ต่อไป — บทที่ 05 · เส้นทางแบบอัลกอริทึม: การค้นหาที่วิจัยดำเนินงานทำด้วย
ไม้บรรทัดเลื่อน คนอีกรุ่นจะทำด้วยเครื่องจักร — และเครื่องจักรกลุ่มแรกที่ค้นหาได้ก็
จะต้องการทฤษฎีของตัวเอง

CHAPTER 05

The Algorithmic Path

เส้นทางแบบอัลกอริทึม

Amsterdam, a café terrace, 1956

อัมสเตอร์ดัม ระเบียงร้านกาแฟ ปี 1956

EN

Edsger Dijkstra was twenty-six, shopping with his fiancée Ria, and tired. He had recently taken a job at the Mathematisch Centrum in Amsterdam demonstrating a new computer called the ARMAC, and he wanted a problem simple enough to explain to a nontechnical audience yet impressive enough to show the machine earning its keep. Sitting down for coffee, without pencil, paper, or access to a library, he set himself the question of the shortest route between two Dutch cities and, in his own later telling, worked out the solution in about twenty minutes (David Gries, "Edsger Dijkstra and the Shortest-Path Algorithm," Cornell CS). The algorithm he sketched that morning — grow a set of cities whose shortest distance from the start is already known for certain, and at each step add whichever unvisited city is now closest — is still, seven decades later, the default answer whenever a computer needs to find the cheapest way from here to there. He did not publish it until three years afterward, as a spare three-page note that solved two unrelated graph problems in the space most papers need for an abstract (Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik* 1 (1959)).

TH

อัลกอริทึมที่ยังเป็นคำตอบตั้งต้นของคอมพิวเตอร์ทุกเครื่องเมื่อต้องหาทางที่ถูกที่สุดจากที่นี่ไปที่นั่น ถูกคิดขึ้นในร้านกาแฟ ในเวลาราวยี่สิบนาที Edsger Dijkstra อายุยี่สิบหก ออกไปเดินช้อปปิ้งกับ Ria คู่หมั้นของเขา และเหนื่อย เขาเพิ่งรับงานที่ Mathematisch Centrum ในอัมสเตอร์ดัม หน้าที่คือสาธิตคอมพิวเตอร์เครื่องใหม่ชื่อ ARMAC เขาต้องการโจทย์ที่ง่ายพอจะอธิบายให้คนนอกสายเทคนิคฟังได้ แต่ก็น่า

ประทับใจพอจะแสดงว่าเครื่องนี้คุ้มค่าที่เลี้ยงไว้ พอนั่งลงดื่มกาแฟ ไม่มีดินสอ ไม่มีกระดาษ ไม่มีห้องสมุดให้ค้น เขาตั้งคำถามกับตัวเองว่าเส้นทางที่สั้นที่สุดระหว่างเมืองดัตช์สองเมืองคือเส้นทางไหน ตัวเลขยี่สิบนาทีก่อนที่มาจากปากเขาเองในภายหลัง (David Gries, "Edsger Dijkstra and the Shortest-Path Algorithm," Cornell CS) วิธีที่เขาร่างเข้าวันนั้นมีสองก้าว — ขยายกลุ่มเมืองที่รู้ระยะสั้นที่สุดจากจุดตั้งต้นแน่นอนแล้ว และทุกก้าวเพิ่มเมืองที่ยังไม่ได้ไปซึ่งตอนนี้ใกล้ที่สุดเข้าไป — เจ็ดทศวรรษให้หลังยังไม่มีใครแทนมัน เขาไม่ได้ตีพิมพ์จนอีกสามปีต่อมา และตีพิมพ์เป็นโน้ตสามหน้าที่ยาวเขียนอย่างประหยัด แก้ปัญหาคณិតสองข้อที่ไม่เกี่ยวกับเลย ในเนื้อที่ที่บทความส่วนใหญ่ใช้เขียนแค่บทคัดย่อ (Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik* 1 (1959))

EN

There is a pleasing, almost too-neat symmetry in the fact that a book called *The Shortest Path* arrives, in its fifth chapter, at an algorithm literally named for finding shortest paths, invented over coffee in less time than it takes to read this paragraph twice. But the neatness is real, not manufactured, and it marks the hinge this chapter turns on. Chapter 04 showed a generation of scientists discovering that real-world problems — convoys, diets, factories — could be captured as mathematical objects and solved by general-purpose procedures. This chapter is about what happened next: those procedures themselves became objects of study. People began asking not just "does a method exist?" but "how good a method can exist, and how would we know?" Search — the raw, brute business of trying paths through a space of possibilities — became a subject with its own theory, its own proofs, and eventually its own machines.

TH

หนังสือชื่อ เส้นทางที่สั้นที่สุด เดินทางถึงบทที่ห้าแล้วเจออัลกอริทึมที่ตั้งชื่อตามการหาเส้นทางสั้นที่สุดตรงๆ คิดขึ้นระหว่างจิบกาแฟ ในเวลาน้อยกว่าที่ใช้อ่านย่อหน้าทั้งสองรอบ ความสมมาตรนี้น่าพอใจ เกือบจะเรียบร้อยเกินไป แต่มันเป็นของจริง ไม่ได้จัดฉาก และมันคือบานพับที่บทนี้หมุนอยู่บน บทที่ 04 แสดงให้เห็นนักวิทยาศาสตร์รุ่นหนึ่งค้นพบว่าปัญหาในโลกจริง — กองเรือคุ้มกัน สูตรอาหาร โรงงาน — จับใส่กรอบเป็นวัตถุทางคณิตศาสตร์แล้วแก้ด้วยขั้นตอนวิธีเนกประสงค์ได้ บทนี้ว่าด้วยสิ่งที่เกิด

ขึ้นต่อจากนั้น: ขั้นตอนวิธีเหล่านั้นเองกลายเป็นวัตถุที่ถูกรู้จักศึกษา คำถามเปลี่ยนจาก "มีวิธีอยู่ไหม" เป็น "วิธีที่ ดีแค่ไหน จึงจะมีอยู่ได้ และเราจะรู้ได้อย่างไร" การค้นหา (search) — งานดิบที่ใช้กำลังล้วนลงเดินเส้นทางต่างๆ ผ่านพื้นที่ของความเป็นไปได้ — กลายเป็นวิชาที่มีทฤษฎีของตัวเอง มีบทพิสูจน์ของตัวเอง และในที่สุดก็มีเครื่องจักรของตัวเอง

The Man Whose Name Became a Word

ชายผู้มีชื่อกลายเป็นคำศัพท์

EN

The word for what Dijkstra did that morning — devise a step-by-step procedure guaranteed to reach a correct answer — is *algorithm*, and the word has its own long detour before it lands in a computer science course.

TH

สิ่งที่ Dijkstra ทำเช้าวันนั้น — คิดขั้นตอนที่ละก้าวที่รับประกันว่าจะไปถึงคำตอบที่ถูกต้อง — มีชื่อเรียกว่า อัลกอริทึม คำนี้เดินทางอ้อมมายาวก่อนจะลงเอยในห้องเรียนวิทยาการคอมพิวเตอร์

EN

It comes from the Latinized name of a ninth-century scholar working in the House of Wisdom in Baghdad: Muhammad ibn Musa al-Khwarizmi. His treatise on Hindu-Arabic numerals, translated into Latin as *Algoritmi de numero Indorum*, gave medieval Europe both a new way to calculate — the positional decimal system, replacing Roman numerals — and, eventually, a new word: *algorismus*, referring at first specifically to arithmetic done this new way. Only much later, in the twentieth century, did "algorithm" widen into its modern sense: any finite, unambiguous, mechanically executable procedure for solving a class of problems (findanexpert.unimelb.edu.au, "Why are algorithms called algorithms?"; NPR, "How 'algorithm' got its name from a 9th-century Persian mathematician"). Al-Khwarizmi's other great book gave mathematics the word *algebra*, from *al-jabr*, "completion" or "restoration" — a term for the operation of moving a quantity from one side of an equation to the other. The scribes of chapter 01 wrote down individual solved problems as teaching examples; al-Khwarizmi wrote down the *pro-*

cedure itself, stripped of any particular numbers, so that it could be applied to numbers no one had thought of yet. That abstraction — the method as an object in its own right, portable across every instance of a problem — is the direct ancestor of everything in this chapter.

TH

คำนี้มาจากชื่อในรูปละตินของนักปราชญ์ศตวรรษที่เก้าผู้ทำงานอยู่ในบ้านแห่งปัญญา (House of Wisdom) ที่แบกแดด: Muhammad ibn Musa al-Khwarizmi ตีราของเขาวัดด้วยเลขฮินดู-อารบิก ถูกแปลเป็นละตินในชื่อ *Algoritmi de numero Indorum* มันมอบสองอย่างให้ยุโรปยุคกลาง อย่างแรกคือวิธีคำนวณแบบใหม่ — ระบบทศนิยมเชิงตำแหน่งที่มาแทนเลขโรมัน อย่างที่สองมาช้ากว่า คือคำใหม่คำหนึ่ง: *algorismus* ตอนแรกหมายถึงเลขคณิตที่ทำด้วยวิธีใหม่นี้โดยเฉพาะ คำว่า "algorithm" ต้องรอถึงศตวรรษที่ยี่สิบจึงขยายมาสู่ความหมายสมัยใหม่: ขั้นตอนวิธีใดๆ ที่มีจำนวนก้าวจำกัด ไม่กำกวม และเครื่องจักรทำตามได้ สำหรับแก้ปัญหานั้น (findanexpert.unimelb.edu.au, "Why are algorithms called algorithms?"; NPR, "How 'algorithm' got its name from a 9th-century Persian mathematician") หนังสือเล่มใหญ่อีกเล่มของ al-Khwarizmi มอบคำว่า *algebra* ให้คณิตศาสตร์ มาจาก *al-jabr* ที่แปลว่า "การเติมให้ครบ" หรือ "การคืนสภาพ" — ศัพท์เรียกการย้ายปริมาณจากข้างหนึ่งของสมการไปอีกข้างหนึ่ง ความต่างจากอาลักษณ์ในบทที่ 01 อยู่ตรงนี้: อาลักษณ์จดโจทย์ที่แก้แล้วที่ละข้อไว้เป็นตัวอย่างสอน; al-Khwarizmi จด ตัวขั้นตอนวิธีเอง ถอดตัวเลขเฉพาะออกหมด เพื่อให้เอาไปใช้กับตัวเลขที่ยังไม่มีใครนึกถึงได้ วิธีในฐานะวัตถุที่มีสถานะของตัวเอง ยกไปใช้ได้กับทุกอินสแตนซ์ (instance) ของปัญหา — นามธรรมข้อนี้คือบรรพบุรุษสายตรงของทุกอย่างในบทนี้

The Primitive Moves: Sweeping Outward, Plunging Deep

ทำพื้นฐาน: กวาดออกไปรอบตัว กับดำดิ่งลงลึก

EN

Before Dijkstra's algorithm, and underneath it, sit two even more basic strategies for exploring a space of possibilities, and it is worth naming them plainly because everything more sophisticated in this chapter is built from them.

TH

ได้อัลกอริทึมของ Dijkstra มีกลยุทธ์พื้นฐานกว่าอีกสองอย่างสำหรับสำรวจพื้นที่ของความเป็นไปได้ ทั้งสองมาก่อนมัน ควรเรียกชื่อให้ชัด เพราะทุกอย่างที่ซับซ้อนกว่านั้นในบทนี้สร้างขึ้นจากสองอย่างนี้

EN

Imagine a maze, or a family tree, or a web of friendships, represented the way chapter 02 taught Euler to represent Königsberg: as a graph of nodes and connecting edges. **Breadth-first search** starts at one node and explores outward in expanding rings — first everything one step away, then everything two steps away, and so on — guaranteeing that the first time it reaches any given node, it has found the *shortest* possible route to it, measured in number of steps. It was reinvented for exactly this purpose in 1959 by Edward F. Moore, who wanted a mechanical way to find the shortest way out of a maze, though a similar idea had been buried, unpublished, in Konrad Zuse's rejected 1945 doctoral thesis (Wikipedia, "Breadth-first search"). **Depth-first search**, by contrast, commits to a single path and follows it as far as it can go before backing up — plunging down one corridor of the maze to its very end before trying another. It has no built-in guarantee of finding the shortest route, but it uses far less memory to run and, in the hands of Robert Tarjan and John Hopcroft in the early 1970s, turned out to unlock efficient solutions to a surprising range of graph problems, from testing whether a map can be drawn without crossing lines to finding a network's structural weak points (Wikipedia, "Depth-first search").

TH

เริ่มจากเขาวงกต หรือแผนผังตระกูล หรือใยความสัมพันธ์ของเพื่อนฝูง แทนด้วยวิธีเดียวกับที่บทที่ 02 เล่าว่า Euler แทน Königsberg: เป็นกราฟของโหนดกับเส้นเชื่อม

ค้นหาความกว้าง (breadth-first search, BFS) เริ่มที่โหนดหนึ่งแล้วสำรวจออกไปเป็นวงที่ขยายกว้างขึ้น — ทุกอย่างที่อยู่ห่างหนึ่งก้าวก่อน แล้วทุกอย่างที่อยู่ห่างสองก้าวไล่ไปเรื่อยๆ ผลคือการรับประกันข้อหนึ่ง: ครั้งแรกที่เราไปถึงโหนดใดก็ตาม มันได้พบเส้นทางที่สั้นที่สุดไปยังโหนดนั้นแล้ว วัดด้วยจำนวนก้าว Edward F. Moore คิดมันขึ้นใหม่เพื่อจุดประสงค์นี้พอดีในปี 1959 เขาต้องการวิธีเชิงกลไกในการหาทางออกที่สั้นที่สุดจากเขาวงกต ไอเดียคล้ายกันฝังอยู่ก่อนแล้วโดยไม่ได้ตีพิมพ์ ในวิทยานิพนธ์ปริญญาเอกปี 1945 ของ Konrad Zuse ที่ถูกปฏิเสธ (Wikipedia, "Breadth-first search")

ค้นหาความลึก (depth-first search, DFS) ทำตรงกันข้าม: เลือกเส้นทางเดียวแล้วเดินตามให้ไกลที่สุดก่อนถอยกลับ — ดำดิ่งลงช่องทางหนึ่งของเขาวงกตจนสุดทาง แล้วจึงลองช่องอื่น มันไม่รับประกันว่าจะพบเส้นทางสั้นที่สุด แต่ใช้หน่วยความจำน้อยกว่ามาก และในมือของ Robert Tarjan กับ John Hopcroft ช่วงต้นทศวรรษ 1970 มันปลดล็อกคำตอบที่มีประสิทธิภาพให้ปัญหากราฟหลายชนิดที่ไม่มีใครคาด ตั้งแต่ทดสอบว่าแผนที่หนึ่งวาดโดยไม่มีเส้นตัดกันได้หรือไม่ ไปจนถึงหาจุดอ่อนเชิงโครงสร้างของเครือข่าย (Wikipedia, "Depth-first search")

E N

Neither strategy knows anything about the problem it is solving beyond the bare shape of the graph. That ignorance is their strength — they work on any graph whatsoever — and their weakness: they waste enormous effort exploring directions that, to anyone who actually understood the problem, would look obviously hopeless. Dijkstra's algorithm is breadth-first search's more disciplined cousin, expanding outward not ring by ring but always toward whichever unvisited node is currently cheapest to reach — which is why it finds not just *a* short path but provably *the* shortest one, for any graph where distances are never negative.

T H

ทั้งสองกลยุทธ์ไม่รู้อะไรเลยเกี่ยวกับปัญหาที่กำลังแก้ นอกจากรูปร่างเปล่าๆ ของกราฟ ความไม่รู้นั่นคือจุดแข็ง — ใช้ได้กับกราฟทุกชนิด — และคือจุดอ่อน: มันพยายามแรงไปกับทิศทางที่ใครก็ตามที่เข้าใจปัญหาจริงๆ จะเห็นทันทีว่าไร้ความหวัง อัลกอริทึมของ Dijkstra คือญาติที่มีวินัยกว่าของ BFS มันไม่ขยายทีละวง แต่มุ่งไปยังโหนดที่ยังไม่ได้ไปซึ่งตอนนี้ไปถึงได้ถูกที่สุดเสมอ นั่นคือเหตุผลที่มันไม่ได้แค่หาเส้นทางสั้นสั้นหนึ่ง แต่พิสูจน์ได้ว่าเป็นเส้นทางที่สั้นที่สุด สำหรับกราฟใดก็ตามที่ระยะทางไม่เคยติดลบ

A Mouse Named Theseus

หนูตัวหนึ่งชื่อ Theseus

EN

Six years before Dijkstra's coffee, at Bell Labs in 1950, Claude Shannon — already famous for founding information theory — built a smaller, stranger monument to the same idea: a maze-solving mechanical mouse he named Theseus, after the hero who navigated the Minotaur's labyrinth by unspooling a thread behind him.

TH

หกปีก่อนกาแฟถ้วยนั้นของ Dijkstra ที่ Bell Labs ในปี 1950 Claude Shannon สร้างอนุสาวรีย์ของไอเดียเดียวกันที่เล็กกว่าและแปลกกว่า เขามีชื่อเสียงอยู่แล้วจากการก่อตั้งทฤษฎีสารสนเทศ สิ่งที่เขาสร้างคือหนูกลไกแก้เขาวงกต เขาตั้งชื่อมันว่า Theseus ตามวีรบุรุษผู้หาทางในเขาวงกตของมิโนทอร์ด้วยการคลี่เส้นด้ายทิ้งไว้ข้างหลัง

EN

Theseus was, mechanically, a magnet wrapped in a metal shell, dragged along underneath a re-configurable five-by-five maze by an electromagnet hidden beneath the floor, itself driven by a bank of telephone relays functioning as memory. Released into an unfamiliar maze, the mouse would blunder forward, bumping walls, backing up, trying again — a physical enactment of depth-first search with backtracking — until it stumbled onto a hidden slice of "cheese" that closed an electrical contact. But every turn it made was recorded in the relay memory as it went. Reset to the start of the *same* maze, Theseus would now glide directly to the goal without a single wrong turn, having converted its earlier blind search into a stored, reusable solution (cyberneticzoo.com, "1952 - 'Theseus' Maze-Solving Mouse - Claude Shannon"). This is worth pausing on, because it is a genuinely early and genuinely physical demonstration of the idea that would come to define machine intelligence: search followed by memory is learning. The first pass through a space costs the full price of exploring it. Every later pass can spend that stored knowledge instead of paying again — the same insight, in miniature and in copper wire, that would let Bellman's dynamic program-

ming (chapter 04) avoid resolving identical sub-problems, and that lets a chess program in chapter 10 remember a position it has already evaluated.

TH

ในเชิงกลไก Theseus คือแม่เหล็กหุ้มเปลือกโลหะ มันถูกลากไปได้เขาวงกตขนาดห้าคูณห้าที่ปรับผังใหม่ได้ ด้วยแม่เหล็กไฟฟ้าที่ซ่อนอยู่ใต้พื้น แม่เหล็กไฟฟ้านั้นถูกขับด้วยแผงรีเลย์โทรศัพท์ที่ทำหน้าที่เป็นหน่วยความจำ ปลอ่ยเข้าเขาวงกตที่ไม่คุ้นเคย หนูจะเดินซุ่มซำไปข้างหน้า ชนกำแพง ถอยหลัง ลองใหม่ — DFS พร้อมการย้อนรอย (backtracking) ในรูปธรรม — จนสะดุดเข้ากับ "ซีส" ขึ้นที่ซ่อนอยู่ ซึ่งทำให้หน้าสัมผัสไฟฟ้าปิดวงจร ทุกการเลี้ยวระหว่างทางถูกบันทึกไว้ในหน่วยความจำรีเลย์ ตั้งกลับไปจุดเริ่มต้นของเขาวงกตอันเดิม Theseus จะเล่นตรงไปยังเป้าหมายโดยไม่เลี้ยวผิดแม้แต่ครั้งเดียว การค้นหาแบบตาบอดรอบก่อนกลายเป็นคำตอบที่เก็บไว้และใช้ซ้ำได้ (cyberneticzoo.com, "1952 – 'Theseus' Maze-Solving Mouse – Claude Shannon") ตรงนี้ควรหยุดดู เพราะมันคือการสาธิตที่เก๋จริงและเป็นรูปธรรมจริงของไอเดียที่ต่อมาจะนิยามปัญญาของเครื่องจักร: การค้นหาที่ตามด้วยความจำ คือการเรียนรู้ รอบแรกจ่ายราคาเต็มของการสำรวจพื้นที่ ที่ถูกรอบหลังจากนั้นใช้ความรู้ที่เก็บไว้แทนการจ่ายซ้ำ นี่คือการหยั่งเห็น (insight) เดียวกัน ในขนาดจิ๋ว และในสายทองแดง ที่ทำให้กำหนดการพลวัต (dynamic programming) ของ Bellman (บทที่ 04) ไม่ต้องแก้ปัญหาย่อยที่เหมือนกันซ้ำ และที่ทำให้โปรแกรมหมากรุกในบทที่ 10 จำตำแหน่งที่มันประเมินไปแล้วได้

Optimism That Never Lies

การมองโลกในแง่ดีที่ไม่เคยโกหก

EN

Dijkstra's algorithm is exhaustive within its expanding frontier — it never skips a node that might turn out to matter — and that thoroughness is also its limit. On a graph with millions of nodes, exploring "everything closer than the current best" before considering anything farther away wastes effort in every direction that has nothing to do with where you are actually trying to go.

TH

ความครบถ้วนของอัลกอริทึมของ Dijkstra คือขีดจำกัดของมันด้วย มันไล่ครบทุกโหนดภายในแนวหน้าที่กำลังขยาย ไม่เคยข้ามโหนดที่อาจกลายเป็นเรื่องสำคัญ บนกราฟที่มีโหนดเป็นล้าน การสำรวจ "ทุกอย่างที่ใกล้กว่าค่าดีที่สุดในนี้" ก่อนจะพิจารณาอะไรที่ไกลกว่านั้น ผลาญแรงไปในทุกทิศที่ไม่เกี่ยวกับที่ที่ตั้งใจจะไปจริงๆ

EN

The fix arrived in 1968 from a robotics project at the Stanford Research Institute, where researchers Peter Hart, Nils Nilsson, and Bertram Raphael were trying to give a mobile robot named Shakey — one of the first machines able to reason about its own actions in a real room — a way to plan efficient routes around obstacles without exploring the entire building's floor plan every time (Engineering and Technology History Wiki, "Milestones: SHAKEY: The World's First Mobile Intelligent Robot, 1972"). Their algorithm, which came to be called A (*pronounced "A-star"*), adds one ingredient to Dijkstra's method: at each step, instead of ranking unexplored nodes purely by the distance already traveled to reach them, A ranks them by that distance plus a guess — a heuristic estimate of how much farther the goal must still be. On a road map, the straight-line, "as the crow flies" distance to the destination is a natural guess: it is easy to compute and it can never overestimate the true remaining driving distance, since no road is ever shorter than a straight line between the same two points.

T H

ทางแก้มมาถึงในปี 1968 จากโครงการหุ่นยนต์ที่ Stanford Research Institute นักวิจัยสามคน Peter Hart, Nils Nilsson และ Bertram Raphael ต้องการให้หุ่นยนต์เคลื่อนที่ชื่อ Shakey — หนึ่งในเครื่องจักรรุ่นแรกที่ทำให้เหตุผลถึงสิ่งที่ตัวเองทำในห้องจริงได้ — วางแผนเส้นทางอ้อมสิ่งกีดขวางอย่างประหยัด โดยไม่ต้องสำรวจผืนพื้นที่อาคารทุกครั้ง (Engineering and Technology History Wiki, "Milestones: SHAKEY: The World's First Mobile Intelligent Robot, 1972") อัลกอริทึมของพวกเขาภายหลังได้ชื่อว่า A* (อ่านว่า "เอ-สตาร์") มันเติมส่วนผสมเพียงหนึ่งอย่างลงในวิธีของ Dijkstra: ในแต่ละก้าว แทนที่จะจัดอันดับโหนดที่ยังไม่สำรวจด้วยระยะทางที่เดินมาถึงมันแล้วล้วนๆ A* จัดอันดับด้วยระยะทางนั้น บวกกับการเดา — ค่าประมาณจากวิธีเดาอย่างมีหลัก (heuristic) ว่าเป้าหมายน่าจะยังอยู่ไกลออกไปอีกเท่าไร บนแผนที่ถนน ระยะเส้นตรงแบบ "นกบินตรง" ไปยังจุดหมายคือตัวเดาที่เหมาะสมเอง คำนวณง่าย และไม่มีทางประเมินระยะขั้นบริบทที่เลื้อมจริงเกินความจริง เพราะไม่มีถนนเส้นไหนสั้นกว่าเส้นตรงระหว่างจุดสองจุดเดียวกัน

E N

That last property — never overestimating — is the entire secret, and it is worth explaining in plain terms because it is one of the most beautiful small theorems in this book's whole story. A heuristic with this property is called *admissible*: it may be wrong, but it is never wrong in the optimistic direction beyond the truth — it can underestimate the remaining distance, but it can never claim a path is shorter than it actually is. Give A an *admissible heuristic*, however crude, and it is mathematically guaranteed to still find the truly shortest path, exactly as Dijkstra's algorithm would, while typically exploring only a sliver of the space Dijkstra's algorithm would have to touch. The heuristic doesn't need to be right. It only needs to never lie in the direction of false hope. Optimism, bounded so it can never overshoot the truth, turns out to be free — you get to search faster and you are never fooled into a wrong answer. Hart, Nilsson, and Raphael's original 1968 paper, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths in Graphs," struggled to find a journal willing to publish it, and Raphael, frustrated, eventually helped found a new AI journal partly so the paper would have somewhere to appear (IEEE Transactions on Systems Science and Cybernetics 4, no. 2 (1968); doi

10.1109/TSSC.1968.300136). A now runs, in one variant or another, inside video game pathfinding, logistics routing, and the descendants of Shakey still driving across the surface of Mars.

TH

คุณสมบัติข้อสุดท้ายนั้น — ไม่ประเมินเกิน — คือความลับทั้งหมด และสมควรอธิบายด้วยคำธรรมดา เพราะมันคือหนึ่งในทฤษฎีบทเล็กๆ ที่งามที่สุดในเรื่องราวทั้งหมดของหนังสือเล่มนี้ heuristic ที่มีคุณสมบัตินี้เรียกว่า ไม่ประเมินเกิน (admissible): มันผิดได้ แต่ไม่เคยผิดไปทางมองโลกในแง่ดีเกินความจริง มันประเมินระยะที่เหลือน้อยกว่าจริงได้ แต่ไม่มีวันอ้างว่าเส้นทางหนึ่งสั้นกว่าที่มันเป็นจริง ให้ admissible heuristic กับ A* ต่อให้หยาบแค่ไหน มันก็ยังรับประกันทางคณิตศาสตร์ว่าจะหาเส้นทางที่สั้นที่สุดจริงๆ พบ เหมือนที่อัลกอริทึมของ Dijkstra ทำ แต่โดยทั่วไปตะแคงพื้นที่เพียงเสี้ยวเดียวของที่อัลกอริทึมของ Dijkstra ต้องตะแคง heuristic ไม่จำเป็นต้องถูก มันแค่ต้องไม่โกหกไปทางความหวังลมๆ แล้งๆ การมองโลกในแง่ดีที่ถูกล็อกไว้ไม่ให้เลยความจริง จึงกลายเป็นของฟรี ค้นหาได้เร็วขึ้น และไม่มีวันถูกหลอกไปสู่คำตอบที่ผิด บทความความต้นฉบับปี 1968 ของ Hart, Nilsson และ Raphael ชื่อ "A Formal Basis for the Heuristic Determination of Minimum Cost Paths in Graphs" หาวารสารที่ยอมตีพิมพ์แทบไม่ได้ ในที่สุด Raphael ที่ติดขัดกับเรื่องนี้ก็มีส่วนก่อตั้งวารสาร AI ฉบับใหม่ขึ้นมา ส่วนหนึ่งก็เพื่อให้อภัยความมีที่ลง (IEEE Transactions on Systems Science and Cybernetics 4, no. 2 (1968); doi 10.1109/TSSC.1968.300136) ทุกวันนี้ A* ทำงานอยู่ในรูปแบบใดรูปแบบหนึ่ง ทั้งในการหาเส้นทางของวิดีโอเกม การจัดเส้นทางโลจิสติกส์ และในลูกหลานของ Shakey ที่ยังเล่นอยู่บนพื้นผิวดาวอังคาร

Machines That Prove Theorems

เครื่องจักรที่พิสูจน์ทฤษฎีบท

EN

While Shannon's mouse learned mazes and Dijkstra timed his coffee, a separate group at the RAND Corporation was attempting something that, in 1955, sounded closer to science fiction than engineering: teaching a machine to do mathematics.

TH

ขณะที่หนูของ Shannon เรียนรู้เขาวงกตและ Dijkstra จับเวลากาแฟของเขา อีกกลุ่มหนึ่งที่ RAND Corporation กำลังสอนเครื่องจักรให้ทำคณิตศาสตร์ ในปี 1955 เรื่องนี้ฟังดูไกลันยายวิทยาศาสตร์มากกว่าวิศวกรรม

EN

Allen Newell, Herbert Simon, and the programmer Cliff Shaw set out to build a program that could prove theorems in formal logic — specifically, theorems from *Principia Mathematica*, the monumental three-volume attempt by Alfred North Whitehead and Bertrand Russell to derive all of mathematics from a small set of logical axioms. Their program, the Logic Theorist, represented each logical statement as a tree-shaped data structure and searched through chains of substitution, detachment, and inference rules — the same problem-space idea chapter 00 traces to Newell and Simon's later theoretical work — hunting for a sequence of legal steps connecting the axioms to the theorem to be proved. Because the space of possible deductions explodes combinatorially with every added step, Newell and Simon needed a way to prune obviously unpromising branches, and borrowed a word for the rules of thumb they used to do it: *heuristics*, a term Newell had encountered in George Pólya's *How to Solve It*, the handbook chapter 03 covers in full (Wikipedia, "Logic Theorist").

TH

Allen Newell, Herbert Simon และโปรแกรมเมอร์ Cliff Shaw ตั้งใจสร้างโปรแกรมที่พิสูจน์ทฤษฎีบทในตรรกศาสตร์รูปนัยได้ เป้าหมายเจาะจงคือทฤษฎีบทจาก Principia Mathematica งานสามเล่มของ Alfred North Whitehead กับ Bertrand Russell ที่พยายามอนุมานคณิตศาสตร์ทั้งหมดจากสัจพจน์ทางตรรกะชุดเล็กๆ

โปรแกรมของพวกเขาชื่อ Logic Theorist มันแทนข้อความทางตรรกะแต่ละข้อด้วยโครงสร้างข้อมูลรูปต้นไม้ แล้วค้นหาไปตามโซ่ของการแทนค่า การตัดออก และกฎการอนุมาน เพื่อลำดับข้อกฏที่ถูกคิดหาซึ่งเชื่อมสัมพันธ์เข้ากับทฤษฎีบทที่ต้องพิสูจน์ นี่คือไอเดียพื้นที่ปัญหา (problem space) อันเดียวกับที่บทที่ 00 สืบย้อนไปถึงงานทฤษฎีในภายหลังของ Newell กับ Simon พื้นที่ของการอนุมานที่เป็นไปได้ระเบิดออกแบบเชิงการจัดหมู่ทุกครั้งที่เพิ่มข้อเข้าไปหนึ่งข้อ Newell กับ Simon จึงต้องตัดกฏที่เห็นชัดว่าไม่มีอนาคตทิ้ง และหยิบยืมคำมาเรียกกฎสามัญสำนึกที่ใช้ทำเรื่องนี้: heuristic Newell เคยเจอคำนี้ใน How to Solve It ของ George Pólya คู่มือที่บทที่ 03 ครอบคลุมไว้เต็มๆ (Wikipedia, "Logic Theorist")

E N

Completed in 1956, the Logic Theorist went on to prove 38 of the first 52 theorems in Chapter 2 of *Principia Mathematica* — and, in one case, found a proof more elegant than the one Whitehead and Russell had labored over by hand. The theorem is commonly cited by the number 2.85, although a documented note on Wikipedia's account points out that in the original 1910 and 1963 Cambridge editions of *Principia* the same theorem is actually numbered 2.53 — a small numbering discrepancy in the popular retelling worth flagging exactly because this book asks its readers to distrust anecdotes that have been smoothed in the retelling. What is well attested is Russell's reaction: Simon showed him the machine's shorter proof, and Russell reportedly responded with delight. The two men then tried to publish the new proof in *The Journal of Symbolic Logic*, only to have it rejected on the grounds that a new proof of an elementary theorem was not a notable result — the reviewers apparently missing, or unmoved by, the fact that one of the "authors" credited was a computer program (Wikipedia, "Logic Theorist").

T H

Logic Theorist สร้างเสร็จในปี 1956 และพิสูจน์ทฤษฎีบทได้ 38 ข้อจาก 52 ข้อแรกในบทที่ 2 ของ *Principia Mathematica* ในกรณีหนึ่งมันพบบทพิสูจน์ที่สง่างามกว่าบทพิสูจน์ที่ Whitehead กับ Russell ตรากตรำทำด้วยมือ หมายเลขของทฤษฎีบทข้อนี้ไม่ตรงกันสองแหล่ง เรื่องเล่าที่แพร่หลายอ้างหมายเลข 2.85 แต่หมายเหตุที่มีเอกสารรองรับในบทความของ Wikipedia ชี้ว่าในฉบับ Cambridge ปี 1910 และ

1963 ของ Principia ทฤษฎีบทข้อเดียวกันนี้มีหมายเลข 2.53 เล่มนี้ทางความคลาดเคลื่อนนี้ออกให้เห็น เพราะเล่มนี้ขอให้ผู้อ่านไม่ใส่ใจเกร็ดเล่าที่ถูกขัดให้เรียบระหว่างการเล่นต่อ สิ่งที่มีหลักฐานรองรับดีคือปฏิกิริยาของ Russell: Simon เอาบทพิสูจน์ที่สั้นกว่าของเครื่องให้เขาดู และมีรายงานว่า Russell ตอบรับด้วยความยินดี ทั้งสองคนพยายามตีพิมพ์บทพิสูจน์ใหม่นี้ใน The Journal of Symbolic Logic วารสารปฏิเสธ เหตุผลคือบทพิสูจน์ใหม่ของทฤษฎีบทพื้นฐานไม่ใช่ผลงานที่น่าสนใจ ผู้ประเมินดูจะมองข้าม หรือไม่ก็ไม่รู้สึกละอาย กับข้อเท็จจริงที่ว่าหนึ่งใน "ผู้เขียน" ที่ให้เครดิตไว้คือโปรแกรมคอมพิวเตอร์ (Wikipedia, "Logic Theorist")

E N

Newell, Simon, and Shaw followed the Logic Theorist with a more ambitious successor between 1957 and 1959: the General Problem Solver, or GPS, intended not merely to prove logic theorems but to tackle any problem expressible as a set of states and legal operators — the general problem-space formalism this book leans on throughout. GPS's central technique, means-ends analysis, is simple to state and turned out to be extraordinarily durable: compare your current state to your goal state, identify the single biggest difference between them, and select whichever available operator is known to reduce differences of exactly that kind. Solve the biggest gap first, then repeat on whatever gap remains. It is, in miniature, a general-purpose engine for narrowing an enormous space one deliberate step at a time, rather than wandering it at random (Wikipedia, "General Problem Solver"; Wikipedia, "Means-ends analysis"). GPS eventually ran into the same wall every general method runs into — later chapters, especially 06, will explain why generality and efficiency pull in opposite directions — but the technique it introduced still shows up, essentially unchanged, inside modern automated planning systems.

T H

ระหว่างปี 1957 ถึง 1959 Newell, Simon และ Shaw ต่อยอด Logic Theorist ด้วยทฤษฎีบทที่ทะเยอทะยานกว่า: General Problem Solver หรือ GPS มันไม่ได้ตั้งใจแค่พิสูจน์ทฤษฎีบทตรรกะ แต่จัดการปัญหาใดก็ตามที่เขียนออกมาเป็นชุดของสถานะ (state) กับตัวดำเนินการ (operator) ที่ถูกกติกาได้ — รูปนัยของพื้นที่ปัญหาแบบทั่วไปที่หนังสือเล่มนี้พึงอยู่ตลอด เทคนิคหลักของ GPS คือการวิเคราะห์วิธี-เป้าหมาย

(means–ends analysis) พุดออกมาง่าย และทนทานเกินคาด: เทียบสถานะปัจจุบันกับสถานะเป้าหมาย ระบุความต่างที่ใหญ่ที่สุดข้อเดียว แล้วเลือก operator ที่มีอยู่ตัวที่รู้ว่าจะลดความต่างชนิดนั้นพอดี แก่ช่องว่างที่ใหญ่ที่สุดก่อน แล้วทำซ้ำกับช่องว่างที่เหลือ มันคือเครื่องยนต์อเนกประสงค์ขนาดจิ๋วสำหรับบีบพื้นที่มหาศาลให้แคบลงทีละก้าวอย่างตั้งใจ แทนที่จะเดินเตร็ดเตร่ในนั้นแบบสุ่ม (Wikipedia, "General Problem Solver"; Wikipedia, "Means–ends analysis") ในที่สุด GPS ก็ชนกำแพงเดียวกับที่วิธีทั่วไปทุกวิธีชน บทหลังจากนี้ โดยเฉพาะบทที่ 06 จะอธิบายว่าทำไมความเป็นทั่วไปกับประสิทธิภาพถึงดิ่งกันคนละทาง แต่เทคนิคที่มันแนะนำไว้อยู่แทบไม่เปลี่ยนรูป ในระบบวางแผนอัตโนมัติสมัยใหม่

Five Bets About the Shape of the Space

ห้าเดิมพันว่าด้วยรูปร่างของพื้นที่

EN

Step back from any single algorithm in this chapter and a shared logic becomes visible: every search strategy is, underneath its details, a bet about what kind of space it is searching. Naming the bets explicitly turns them from a grab-bag of tricks into a genuine toolkit, one you can hold up against a new problem and ask which bet fits.

TH

กลยุทธ์การค้นหาทุกแบบ ได้รายละเอียดของมัน คือการเดิมพันว่าพื้นที่ที่มันกำลังค้นหาเป็นพื้นที่ชนิดไหน ถอยออกมาจากอัลกอริทึมตัวใดตัวหนึ่งในบทนี้แล้วตรกระร่วมข้อนี้จะปรากฏ เรียกชื่อเดิมพันเหล่านี้ให้ชัด แล้วถ่วงลูกเล่นคละกันจะกลายเป็นชุดเครื่องมือจริง ชุดที่ยกขึ้นเทียบกับปัญหาใหม่แล้วถามได้ว่าเดิมพันไหนเข้ากัน

E N

Greedy algorithms bet that making the locally best choice at every step, without ever reconsidering it, will add up to a globally good result. This is a dangerous bet — it is easy to construct problems where it fails outright — but when a problem's structure guarantees the bet pays off (a property mathematicians call the matroid or exchange condition in the cleanest cases), greedy strategies are unbeatable for their simplicity and speed. Dijkstra's algorithm is, in this light, a greedy algorithm that happens to be provably safe: always finalize whichever unvisited node is currently cheapest, and never look back.

T H

อัลกอริทึมแบบโลภ (greedy) เดิมพันว่าเลือกสิ่งที่ดีที่สุดเฉพาะถิ่นในทุกก้าว ไม่ย้อนกลับมาทบทวนเลย แล้วผลรวมจะดีในระดับทั้งหมด เดิมพันนี้อันตราย — สร้างปัญหาที่มันล้มเหลวโดยสิ้นเชิงได้ง่ายๆ — แต่เมื่อโครงสร้างของปัญหารับประกันว่า เดิมพันได้ผล (คุณสมบัติที่นักคณิตศาสตร์เรียกว่าเงื่อนไขมาทรอยด์หรือเงื่อนไขแลกเปลี่ยนในกรณีที่ดีที่สุด) กลยุทธ์ greedy ไม่มีอะไรสู้ได้ในแง่ความเรียบง่ายและความเร็ว มองแบบนี้ อัลกอริทึมของ Dijkstra คืออัลกอริทึม greedy ที่บังเอิญพิสูจน์ได้ว่าปลอดภัย: ปิดตายโหนดที่ยังไม่ได้ไปซึ่งถูกที่สุดตอนนี้เสมอ และไม่หันกลับไปมองอีก

E N

Divide-and-conquer bets that a hard problem can be split into smaller, structurally identical copies of itself, solved independently, and stitched back together. It is the strategy behind fast sorting and multiplication algorithms, and its bet is about *decomposability*: does solving the pieces really tell you how to solve the whole?

T H

แบ่งแล้วพิชิต (divide and conquer) เดิมพันว่าปัญหายากข้อหนึ่งแยกออกเป็นสำเนาของตัวเองที่เล็กลงและมีโครงสร้างเหมือนกันได้ แยกกัน แล้วเย็บกลับเข้าด้วยกัน มันคือกลยุทธ์เบื้องหลังอัลกอริทึมเรียงลำดับและคูณเลขที่เร็ว เดิมพันของมันคือ การแยกส่วนได้: แก่ชิ้นส่วนแล้วรู้จริงหรือว่าจะแก้ทั้งก้อนอย่างไร

E N

Dynamic programming, which chapter 04 introduced through Bellman's principle of optimality, bets that a problem's sub-problems *overlap* — that the same smaller question will recur many times inside a larger search — and that storing each answer the first time it is computed will save enormous repeated effort later. Where divide-and-conquer's pieces are independent, dynamic programming's pieces share work, and the technique exists precisely to avoid redoing that shared work, the way Shannon's mouse never re-explored a corridor it had already mapped.

T H

dynamic programming ที่บทที่ 04 แนะนำผ่านหลักความเหมาะสมที่สุดของ Bellman เดิมพันว่าปัญหาย่อยของปัญหาหนึ่ง ซ้อนทับกัน — คำถามเล็กข้อเดิมจะวนกลับมาหลายครั้งภายในการค้นหาที่ใหญ่กว่า — และเก็บคำตอบแต่ละข้อไว้ตั้งแต่ครั้งแรกที่คำนวณจะประหยัดแรงที่ต้องทำซ้ำในภายหลัง ชิ้นส่วนของแบ่งแล้วพิชิตเป็นอิสระต่อกัน ชิ้นส่วนของ dynamic programming แบ่งงานกันใช้ เทคนิคนี้มีอยู่เพื่อเลี่ยงการทำงานร่วนนั้นซ้ำพอดี เหมือนที่หนูของ Shannon ไม่เคยกลับไปสำรวจช่องทางที่มันทำแผนที่ไว้แล้ว

E N

Backtracking bets that a space can be searched depth-first, provided the search can recognize a dead end early and retreat before wasting further effort down it — the disciplined version of what Theseus did by trial, bump, and correction.

T H

backtracking เดิมพันว่าพื้นที่หนึ่งค้นหาแบบ DFS ได้ トラバิดที่การค้นหารู้ตัวเร็วว่าเจอทางตัน แล้วถอยกลับก่อนผลาญแรงลงไปโน้นอีก มันคือเวอร์ชันมีวินัยของสิ่งที่ Theseus ทำด้วยการลอง ชน และแก้

E N

Branch-and-bound, formalized in 1960 by Ailsa Land and Alison Doig for problems requiring whole-number rather than fractional solutions, bets that you can compute a cheap *bound* on how good any solution branching from a given partial choice could possibly be — and use that bound to prune away entire branches of the search tree without ever visiting them, the instant the bound proves they cannot beat a solution you already have in hand (Wikipedia, "Branch and bound"). It is A^* 's admissible heuristic, generalized from finding paths on a map to finding optimal points in a much wider variety of combinatorial spaces.

T H

แตกกิ่งแล้วตัดกรอบ (branch and bound) เดิมพันว่าเราคำนวณ ขอบเขต ราคาถูก ได้ว่าคำตอบใดก็ตามที่แตกกิ่งออกจากตัวเลือกบางส่วนอันหนึ่งจะดีได้มากที่สุดแค่ไหน ทันทีที่ขอบเขตพิสูจน์ว่ากิ่งนั้นชนะคำตอบที่ถืออยู่ในมือแล้วไม่ได้ ก็ตัดกิ่งทั้งกิ่งของตนไม่การค้นหาลำบากโดยไม่ต้องเข้าไปดูเลย Ailsa Land กับ Alison Doig ทำให้มันเป็นรูปนัยในปี 1960 สำหรับปัญหาที่ต้องการคำตอบเป็นจำนวนเต็มแทนที่จะเป็นเศษส่วน (Wikipedia, "Branch and bound") มันคือ admissible heuristic ของ A^* ที่ถูกทำให้ทั่วไปขึ้น จากการหาเส้นทางบนแผนที่ ไปสู่การหาจุดเหมาะสมที่สุดในพื้นที่เชิงการจัดหมู่ที่หลากหลายกว่ามาก

E N

Five bets, five different guesses about the hidden shape of a problem space. None of them is "the" right way to search; each is the right way *when its bet about the space happens to be true*. Reading a new problem well enough to know which bet applies is, itself, most of the skill this book is trying to name.

T H

ห้าเดิมพัน คือห้าการเดาที่ต่างกันเกี่ยวกับรูปร่างที่ซ่อนอยู่ของพื้นที่ปัญหา ไม่มีอันไหนเป็นวิธีค้นหาที่ถูก "อันนั้น"; แต่ละอันถูก เมื่อเดิมพันของมันเกี่ยวกับพื้นที่บังเอิญเป็นจริง อ่านปัญหาใหม่ได้ดีพอจะรู้ว่าเดิมพันไหนใช้ได้ — นั่นคือทักษะเกือบทั้งหมดที่หนังสือเล่มนี้พยายามเรียกชื่อ

An Algorithm Is a Heuristic That Earned a Proof

อัลกอริทึมคือ heuristic ที่หาบทพิสูจน์มาได้

E N

There is a temptation, looking backward from a world of route planners and chess engines, to treat "algorithm" and "heuristic" as opposites — one rigorous and guaranteed, the other a rough guess that might fail. The history in this chapter argues for a different picture. Dijkstra's algorithm began as an informal insight about which city to visit next; what made it an algorithm, rather than a hunch, was that Dijkstra could prove the insight always worked. A*'s heuristic estimate is explicitly, by name, a guess — yet paired with the single provable property of admissibility, that guess becomes part of an algorithm with an ironclad optimality guarantee. Newell and Simon's "rules of thumb" for pruning theorem-proving searches were heuristics in the older, looser sense: useful, unproven, sometimes wrong. The field's entire maturation across this chapter is the slow conversion of hunches into guarantees — not by abandoning intuition, but by discovering exactly which intuitions can be proven never to betray you.

T H

"อัลกอริทึม" กับ "heuristic" ไม่ใช่ขั้วตรงข้าม มองย้อนจากโลกที่มีตัววางแผนเส้นทางและเครื่องยนต์หมากรุก มีสิ่งล่อใจให้ถือว่าอย่างหนึ่งเข้มงวดและรับประกันได้ อีกอย่างเป็นการเดาหยาบๆ ที่อาจพลาด ประวัติศาสตร์ในบทนี้เสี่ยงให้ภาพอีกแบบ อัลกอริทึมของ Dijkstra เริ่มจาก insight แบบไม่เป็นทางการว่าควรไปเมืองไหนต่อ; สิ่งที่ทำให้มันเป็นอัลกอริทึมแทนที่จะเป็นलगสังหรณ์ คือ Dijkstra พิสูจน์ได้ว่า insight นั้นใช้ได้เสมอ ค่าประมาณแบบ heuristic ของ A* เป็นการเดาอย่างเปิดเผย ตั้งแต่ชื่อ แต่พอจับคู่กับคุณสมบัติที่พิสูจน์ได้ข้อเดียวคือ admissibility การเดานั้นก็กลายเป็นส่วนหนึ่งของอัลกอริทึมที่รับประกันคำตอบเหมาะสมที่สุดอย่างหักล้างไม่ได้ กฎสามัญสำนึกของ Newell กับ Simon ที่ใช้ตัดกิ่งในการค้นหาบทพิสูจน์ทฤษฎีบทเป็น heuristic ในความหมายเก่าที่หลวมกว่า: มีประโยชน์ ยังไม่พิสูจน์ และบางทีก็ผิด สาขานี้เติบโตตลอดทั้งบทด้วยการค่อยๆ แปลงलगสังหรณ์ให้เป็นการรับประกัน ไม่ใช่ด้วยการทิ้งสัญชาตญาณ แต่ด้วยการค้นให้เจอว่าสัญชาตญาณข้อไหนกันแน่ที่พิสูจน์ได้ว่าจะไม่มีวันทรยศ

E N

This is also where the algorithmic tradition begins to separate from the operations-research tradition of chapter 04. Linear programming and dynamic programming solve well-defined numerical optimization problems. The algorithms in this chapter solve *search* problems over discrete, often combinatorial spaces — mazes, graphs, logical proofs, game positions — and their central discovery is that the structure of the space itself (its distances, its overlapping sub-problems, its decomposability) determines which search strategy can be proven correct and fast. That discovery sets up the next and darker discovery, waiting in chapter 06: some spaces have no exploitable structure of this kind at all, and no amount of cleverness can manufacture one where none exists.

T H

ตรงนี้เองที่ธรรมเนียมเชิงอัลกอริทึมเริ่มแยกออกจากธรรมเนียมวิจัยดำเนินงาน (operational research, OR) ของบทที่ 04 กำหนดการเชิงเส้น (linear programming) และ dynamic programming แก้ปัญหาการหาค่าเหมาะที่สุดเชิงตัวเลขที่นิยามไว้ชัดเจน อัลกอริทึมในบทนี้แก้ปัญหาการค้นหบนพื้นที่ไม่ต่อเนื่องซึ่งมักเป็นเชิงการจัดหมู่ — เชาวงกต กราฟ บทพิสูจน์ทางตรรกะ ตำแหน่งบนกระดาน การค้นพบใจกลางของพวกมันคือ โครงสร้างของพื้นที่เองต่างหาก (ระยะทางในนั้น ปัญหาย่อยที่ซ้อนทับกัน ความแยกส่วนได้ของมัน) ที่กำหนดว่ากลยุทธ์การค้นหาแบบไหนจะพิสูจน์ได้ว่าถูกและเร็ว การค้นพบนั้นปูทางไปสู่การค้นพบถัดไปที่มืดกว่ามันรออยู่ในบทที่ 06: พื้นที่บางแบบไม่มีโครงสร้างชนิดนี้ให้ใช้ประโยชน์เลย และไม่ว่าจะฉลาดแค่ไหนก็ผลิตมันขึ้นมาในที่ที่ไม่มีไม่ได้

The Move

ทำประจำบท

E N

Move 05 — Order the search: exploit structure so good paths come first.

T H

Move 05 — จัดลำดับการค้นหา — ใช้โครงสร้างดันเส้นทางที่ดีขึ้นมาก่อน

E N

- **Name which of the five bets fits your space before searching it.** Overlapping sub-problems call for dynamic programming; decomposable ones call for divide-and-conquer; a provably safe local choice calls for greedy; anything else usually calls for backtracking or branch-and-bound. Guessing the wrong bet wastes the search before it starts.
- **When you must guess, guess optimistically but never past the truth.** A^* shows that an estimate need not be accurate to be useful — it only needs to be admissible. An honest lower bound on remaining effort is worth more than an unreliable "best guess."
- **Remember what you have already searched.** Shannon's mouse and Bellman's functional equations share one lesson: the first pass through a space is allowed to be expensive, provided nothing forces you to pay for the same sub-problem twice.
- **Distinguish a real guarantee from a plausible heuristic, and say out loud which one you're relying on.** The Logic Theorist's "rules of thumb" and Dijkstra's proven-optimal frontier are both search strategies — only one of them promises it cannot be wrong.

T H

- เรียกชื่อให้ได้ว่าเดิมพันข้อไหนในห้าข้อเข้ากับพื้นที่ของคุณ ก่อนจะเริ่มค้นหามัน ปัญหาย่อยที่ซ้อนทับกันเรียกหา dynamic programming; ปัญหาที่แยกส่วนได้เรียกหาแบ่งแล้วพิชิต; ตัวเลือกเฉพาะถิ่นที่พิสูจน์ได้ว่าปลอดภัยเรียกหา greedy; นอกนั้นมักเรียกหา backtracking หรือ branch and bound การเดาเดิมพันผิดคือการผลาญการค้นหาทั้งตั้งแต่มยังไม่เริ่ม
- เมื่อต้องเดา จงเดาในแง่ดี แต่ห้ามเลยความจริง A^* แสดงให้เห็นว่าค่าประมาณไม่จำเป็นต้องแม่นยำถึงจะมีประโยชน์ — มันแค่ต้อง admissible ขอบล่างที่เชื่อถือได้ของแรงที่เหลือ มีค่ามากกว่า "การเดาที่ดีที่สุด" ที่เชื่อถือไม่ได้
- จำไว้ว่าค้นอะไรไปแล้วบ้าง หนูของ Shannon กับสมการเชิงฟังก์ชันของ Bellman มีบทเรียนร่วมกันข้อหนึ่ง: การเดินผ่านพื้นที่รอบแรกแพงได้ トラバิดที่ไม่มีอะไรบังคับให้จ่ายค่าปัญหาย่อยข้อเดียวกันสองครั้ง
- แยกการรับประกันของจริงออกจาก heuristic ที่ฟังดูมีเหตุผล แล้วพูดออกมาดังๆ ว่ากำลังพึ่งอันไหนอยู่ "กฎสามัญสำนึก" ของ Logic Theorist กับแนวหน้าที่

พิสูจน์ว่าเหมาะสมที่สุดของ Dijkstra ต่างก็เป็นกลยุทธ์การค้นหา — แต่มีแค่อันเดียวที่
สัญญาว่ามันผิดไม่ได้

EN

Next — Chapter 06 · The Wall: not every space bends to a clever order of search. This chapter's algorithms all found structure to exploit — the next chapter meets problems with no exploitable structure at all, and asks what a solver does when the space itself refuses to get shorter.

TH

ต่อไป — บทที่ 06 · กำแพง: ไม่ใช่ทุกพื้นที่ที่จะยอมโค้งให้ลำดับการค้นหาอันชาญฉลาด อัลกอริทึมทั้งหมดในบทนี้เจอโครงสร้างให้ใช้ประโยชน์ — บทหน้าจะพบปัญหาที่ไม่มีโครงสร้างให้ใช้ประโยชน์เลย และถามว่านักแก้ปัญหาทำอะไรเมื่อพื้นที่เองปฏิเสธที่จะสั้นลง

CHAPTER 06

The Wall

กำแพง

A letter to a dying man

จดหมายถึงชายที่กำลังจะตาย

EN

In the spring of 1956, John von Neumann was fifty-three years old and dying of cancer, confined increasingly to Walter Reed hospital, his once-astounding productivity narrowing to whatever his failing body would allow. Kurt Gödel — his old friend from Vienna, the logician who had shattered Hilbert's dream of a complete and consistent mathematics twenty-five years earlier — wrote to him that March, and by most accounts the letter was partly an attempt to give a brilliant, dying mind something to chew on besides its own decline.

TH

ฤดูใบไม้ผลิปี 1956 John von Neumann อายุห้าสิบสาม กำลังจะตายด้วยโรคมะเร็ง และถูกกักตัวอยู่ในโรงพยาบาล Walter Reed นานขึ้นทุกที ผลงานที่เคยไหลออกมาไม่หยุดหดเหลือเท่าที่ร่างกายที่กำลังทรุดจะยอมให้ทำ เดือนมีนาคมปีนั้น Kurt Gödel เขียนจดหมายถึงเขา Gödel คือเพื่อนเก่าจากเวียนนา นักตรรกวิทยาที่ทาบความฝันของ Hilbert เรื่องคณิตศาสตร์ที่ครบถ้วนและสอดคล้องในตัวเองจนแตกเมื่อยี่สิบห้าปีก่อน ตามคำบอกเล่าส่วนใหญ่ จดหมายฉบับนั้นส่วนหนึ่งเขียนขึ้นเพื่อหาอะไรให้สมองที่เหี่ยวแห้งและกำลังจะดับได้ขบคิด นอกเหนือไปจากความเสื่อมถอยของตัวเอง

EN

The question Gödel posed was technical, almost casual in its phrasing, and it would take computer science another decade and a half to realize what he had actually asked. Consider a mathematical statement of a certain length. Checking whether a proposed proof of that statement is *correct* can

obviously be done quickly — you just verify each step. But how long does it take to *find* a proof at all, if one exists? Gödel wanted to know whether this search could always be done in time roughly proportional to the length of the statement (or its square), rather than growing explosively as statements get longer. He was, in effect, asking whether *finding* is fundamentally harder than *checking* — the question that decades later would be given the name P versus NP (Cornell eCommons, "Gödel, von Neumann and the P=?NP Problem"; Panda Notes, "Gödel's Letter to von Neumann"). Von Neumann died the following February. Whether he ever replied is not known; no response survives. The question sat, essentially unnoticed outside a small circle, for fifteen more years.

TH

คำถามของ Gödel เป็นคำถามเชิงเทคนิค เขียนเกือบเหมือนเอ่ยผ่านๆ และวิทยาการคอมพิวเตอร์ต้องใช้เวลามากกว่าสิบห้าปีกว่าจะรู้ว่าเขาถามอะไรไปกันแน่ เอาข้อความทางคณิตศาสตร์ที่มีความยาวค่าหนึ่งมาตั้ง การตรวจว่าบทพิสูจน์ที่มีคนเสนอมาสำหรับข้อความนั้นถูกต้องหรือไม่ ทำได้เร็วอยู่แล้วอย่างเห็นได้ชัด — ตรวจสอบทีละก้าวก็จบ แต่ถ้าบทพิสูจน์มีอยู่ การหามันให้เจอตั้งแต่ต้นใช้เวลาแสนแค้นไหน Gödel อยากรู้ว่าการค้นหา (search) นี้ทำให้เสร็จได้เสมอไหมในเวลาที่เหมาะสมคร่าวๆ ตามความยาวของข้อความ (หรือกำลังสองของความยาวนั้น) แทนที่จะโตะระเบิดเมื่อข้อความยาวขึ้น เอาเข้าจริงเขากำลังถามว่าการหายากกว่าการตรวจโดยพื้นฐานหรือเปล่า คำถามนี้อีกหลายทศวรรษต่อมาจะได้ชื่อว่า P versus NP (Cornell eCommons, "Gödel, von Neumann and the P=?NP Problem"; Panda Notes, "Gödel's Letter to von Neumann") Von Neumann เสียชีวิตในเดือนกุมภาพันธ์ปีถัดมา เขาเคยตอบหรือไม่ ไม่มีใครรู้; ไม่มีคำตอบหลงเหลือมาถึงเรา คำถามนั้นอนึ่งอยู่อีกสิบห้าปี แทบไม่มีใครนึกถึงเลยสักที สังเกตเห็น

E N

This chapter is about what happened when the question was finally answered — not solved, but given a precise shape and a name — and about the strange, double-edged discovery that followed: that some problems appear to resist *every* clever reframing this book has celebrated so far, and that this resistance, once proven, becomes a kind of knowledge in its own right.

T H

บทนี้ว่าด้วยสิ่งที่เกิดขึ้นเมื่อคำถามนี้ได้คำตอบในที่สุด — ไม่ใช่ถูกแก้ แต่ได้รูปทรงที่ชัดเจนและได้ชื่อ — และว่าด้วยการค้นพบที่แปลกและมีคมสองด้านที่ตามมา ปัญหาบางข้อจะต้านทานการเปลี่ยนกรอบ (reframe) อันชาญฉลาดทุกแบบที่หนังสือเล่มนี้ยกย่องมาจนถึงตอนนี้ และเมื่อพิสูจน์การต้านทานนั้นได้แล้ว มันก็กลายเป็นความรู้ชนิดหนึ่งในตัวมันเอง

One Disease, Many Symptoms

โรคเดียว อาการมากมาย

EN

The formal birth of the field came in 1971, when Stephen Cook, then at the University of Toronto, published a short paper titled "The Complexity of Theorem-Proving Procedures." Cook studied the Boolean satisfiability problem, SAT: given a logical formula built from variables that can each be TRUE or FALSE, connected by ANDs, ORs, and NOTs, is there *some* assignment of values that makes the whole formula true? Checking a proposed assignment is trivial — plug in the values and evaluate. Finding one, when the formula has thousands of variables locked together by thousands of constraints, seemed to require something close to trying all possibilities.

TH

สาขานี้ถือกำเนิดอย่างเป็นทางการในปี 1971 เมื่อ Stephen Cook ซึ่งตอนนั้นอยู่ที่ University of Toronto ตีพิมพ์บทความสั้นชื่อ "The Complexity of Theorem-Proving Procedures" Cook ศึกษาปัญหาความสอดคล้องได้แบบบูล (Boolean satisfiability) หรือ SAT: ให้สูตรตรรกะที่สร้างจากตัวแปรที่แต่ละตัวเป็น TRUE หรือ FALSE ได้ เชื่อมกันด้วย AND, OR และ NOT — มีการกำหนดค่าชุดใดชุดหนึ่งที่ทำให้สูตรทั้งสูตรเป็นจริงหรือไม่ ตรวจสอบการกำหนดค่าที่เสนอมานั้นง่ายมาก — แทนค่าลงไปแล้วประเมิน หากการกำหนดค่าสักชุดนั้นอีกเรื่อง เมื่อสูตรมีตัวแปรเป็นพันตัวที่ล็อกเข้าด้วยกันด้วยข้อจำกัดเป็นพันข้อ จะต้องใช้อะไรที่ใกล้เคียงกับการลองทุกความเป็นไปได้

EN

Cook's insight was not to solve SAT faster. It was to show that SAT is, in a precise sense, the *hardest* problem of its kind: any problem whose solutions can be quickly checked (this class is called NP, for "nondeterministic polynomial time") can be systematically translated — "reduced" — into an equivalent instance of SAT, using only a modest, efficient amount of computation to perform the translation. If you could solve SAT quickly, you could solve *anything* checkable-in-NP quickly, by first translating it into SAT. This property earned SAT the label NP-complete: complete, in the sense of capturing the full difficulty of the entire class (GeeksforGeeks, "Cook-Levin

Theorem"). Working independently and in near-total isolation behind the Iron Curtain, the Soviet mathematician Leonid Levin arrived at essentially the same result at almost the same time, framed slightly differently, around 1973 — a discovery so cleanly convergent that the result now carries both names, the Cook–Levin theorem, an acknowledgment that important mathematical truths sometimes get discovered twice by people who have never read each other's work.

TH

การหยั่งเห็น (insight) ของ Cook ไม่ใช่การแก้ SAT ให้เร็วขึ้น แต่คือการแสดงว่า SAT เป็นปัญหาที่ยากที่สุดในชนิดของมัน ในความหมายที่แม่นยำ: ปัญหาใดก็ตามที่ตรวจสอบได้เร็ว (คลาสนี้เรียกว่า NP ย่อจาก "nondeterministic polynomial time") แปลอย่างเป็นระบบ — "ลดรูป" — ไปเป็นอินสแตนซ์ (instance) ของ SAT ที่สมมูลกันได้ และการเปลี่ยนนั้นใช้การคำนวณเพียงเล็กน้อยและมีประสิทธิภาพ ถ้าแก้ SAT ได้เร็ว ก็แก้อะไรก็ตามที่ตรวจสอบได้ใน NP ได้เร็ว เพียงแปลมันเป็น SAT ก่อน คุณสมบัตินี้ทำให้ SAT ได้ป้าย NP-complete: complete ในความหมายว่ามันจับความยากทั้งหมดของคลาสทั้งคลาสไว้ได้ (GeeksforGeeks, "Cook–Levin Theorem")

ผลลัพธ์เดียวกันในสาระสำคัญถูกค้นพบอีกครั้งในเวลาไล่เลี่ยกัน ราวปี 1973 โดยนักคณิตศาสตร์โซเวียต Leonid Levin ที่ทำงานอย่างเป็นอิสระและแทบจะโดดเดี่ยว สิ้นเชิงอยู่หลังม่านเหล็ก โดยจัดกรอบต่างออกไปเล็กน้อย การค้นพบบรรจบกัน สะอาดจนทฤษฎีบทนี้ทุกวันนี้แทบทั้งสองชื่อ คือทฤษฎีบท Cook–Levin เป็นการยอมรับว่าความจริงทางคณิตศาสตร์ที่สำคัญบางครั้งถูกค้นพบสองรอบ โดยคนที่ไม่เคยอ่านงานของกันและกัน

EN

It helps to make the checking-versus-finding distinction concrete with something more familiar than Boolean formulas. Take a hard Sudoku puzzle. If a friend hands you a completed grid and claims it solves the puzzle, verifying the claim is almost mechanical: scan each row, column, and box for repeated digits, and you are done in seconds, however large the grid. But *finding* that completed grid in the first place, starting from a sparse set of clues, can require searching through a genuinely enormous space of candidate fillings — there is no known shortcut that scales as gracefully as the checking does. Sudoku, scaled up to arbitrary grid sizes, is itself known to

be NP-complete. Every problem on this chapter's list shares that same lopsidedness: an answer that is cheap to confirm and, as far as anyone has proven, expensive to construct.

TH

ความต่างระหว่างการตรวจกับการหาจะเป็นรูปธรรมขึ้น ถ้าใช้อะไรที่คั่นกว่าสูตรตรรกะ เอาซูโดกุข้อยากๆ สักข้อ ถ้าเพื่อนยื่นตารางที่เติมครบแล้วให้และอ้างว่านี่คือคำตอบ การตรวจสอบคำตอบนั้นแทบเป็นงานกลไก: กวาดดูแต่ละแถว แต่ละหลัก และแต่ละกล่องว่ามีเลขซ้ำไหม จบในไม่กี่วินาที ไม่ว่าตารางจะใหญ่แค่ไหน แต่การหาตารางที่เติมครบนั้นตั้งแต่ต้น โดยเริ่มจากค่าใบ้เพียงไม่กี่ตัว อาจต้องค้นผ่านพื้นที่ของการเติมที่เป็นไปได้ที่ใหญ่จริงๆ — ไม่มีทางลัดที่รู้จักตัวไหนขยายตัวได้ราบรื่นเท่าการตรวจ ซูโดกุที่ขยายไปยังขนาดตารางใดก็ได้ นั่นเป็นที่รู้กันว่าเป็น NP-complete ในตัวมันเอง ปัญหาทุกข้อในรายการของบทนี้เปี้ยวข้างแบบเดียวกัน: คำตอบยืนยันได้ถูก และเท่าที่ใครพิสูจน์ได้ สร้างขึ้นมาได้แพง

EN

What a reduction actually *does*, in the vocabulary this book has been building, deserves to be named as a move in its own right, because it is one of the most powerful weapons a solver can pick up. A reduction says: *your problem, however unfamiliar it looks, is secretly a disguised copy of a problem someone has already fully understood.* Translate your instance into the known problem's language, solve that, translate the answer back. This is the same act of recognition that let chapter 04's operations researchers see a plywood mill and a wartime diet as the same linear program in different clothes — except here the recognition cuts the opposite way. Instead of finding a shortcut, a reduction can prove that no shortcut is possible, by showing that your problem is at least as hard as one already known to resist every attempt at a shortcut.

TH

สิ่งที่การลดรูป (reduction) ทำจริงๆ ในคำศัพท์ที่หนังสือเล่มนี้ก่อมา สมควรได้ชื่อเป็นท่าหนึ่งของมันเอง เพราะมันคือหนึ่งในอาวุธที่ทรงพลังที่สุดที่นักแก้ปัญหายิบขึ้นมาได้ การลดรูปพูดว่า: ปัญหาของคุณ ไม่ว่าจะดูแปลกหน้าแค่ไหน แท้จริงคือสำเนาปลอมตัวของปัญหาที่มีคนเข้าใจถ่องแท้ไปแล้ว แปลอินสแตนซ์ของคุณเป็นภาษาของปัญหาที่รู้จักกัน แก้อันนั้น แล้วแปลคำตอบกลับ นี่คือการรู้จำแบบเดียวกับ

ที่ทำให้นักวิจัยดำเนินงาน (operational research, OR) ในบทที่ 04 มองเห็นโรงงานไม่ติดกับสูตรอาหารยามสงครามเป็นกำหนดการเชิงเส้น (linear programming) อันเดียวกันที่ใส่เสื้อผ้าคนละชุด — เว้นแต่ตรงนี้การรู้จำเงื่อนไขคนละทาง การลดรูปไม่ได้พบทางลัด แต่พิสูจน์ได้ว่าไม่มีทางลัดใดเป็นไปได้ ด้วยการแสดงว่าปัญหาของคุณยากอย่างน้อยเท่ากับปัญหาที่รู้จักกันแล้วว่าด้านทฤษฎีความพยายามหาทางลัดทุกครั้ง

Karp's Rogues' Gallery

หอผู้ร้ายของ Karp

EN

Cook's theorem proved that one problem, SAT, was the hardest of its class. It took Richard Karp's 1972 paper, "Reducibility Among Combinatorial Problems," to show the world how *crowded* that class actually was. Karp took Cook's result and, using nothing but a chain of reductions, showed that twenty-one further problems — familiar, practical, seemingly unrelated problems that engineers and planners had been quietly struggling with for years — were all secretly NP-complete too (Karp, "Reducibility Among Combinatorial Problems" (1972); Wikipedia, "Karp's 21 NP-complete problems").

TH

ทฤษฎีบทของ Cook พิสูจน์ว่าปัญหาข้อหนึ่ง คือ SAT ยากที่สุดในคลาสของมัน คลาสนั้นแออัดแค่ไหนจริงๆ โลกต้องรอบทความปี 1972 ของ Richard Karp ชื่อ

"Reducibility Among Combinatorial Problems" Karp หยิบผลของ Cook มาใช้เพียงโซ่ของการลดรูปเท่านั้น แล้วแสดงว่าปัญหาอีกยี่สิบเอ็ดข้อล้วนเป็น NP-complete เหมือนกันหมดโดยที่ไม่มีใครรู้ — ปัญหาที่คุ้นเคย ใช้งานจริง และดูไม่เกี่ยวกันเลย ที่วิศวกรกับนักวางแผนคุ้นเคยกับมันเจิบๆ มาหลายปี (Karp, "Reducibility Among Combinatorial Problems" (1972); Wikipedia, "Karp's 21 NP-complete problems")

EN

Karp's list reads like a tour of applied mathematics: whether a graph can be colored with a given number of colors so that no two connected nodes share one; whether a set of items can be partitioned into two piles of equal total value; whether a network can be covered by a small set of chosen

nodes touching every edge; whether a Boolean formula in a certain restricted form is satisfiable at all. These problems had been attacked, separately, by graph theorists, operations researchers, and logicians who had no particular reason to think their struggles were connected. Karp's chain of reductions revealed they were all the same disease wearing twenty-one different costumes — precisely the discovery chapter 04 celebrated in Kantorovich's plywood and Stigler's diet, except that here the shared identity is a shared curse rather than a shared shortcut. If you found an efficient algorithm for any one of them, you would have found one for all of them, and for every problem in NP besides. Nobody has ever found one. Nobody has proven that nobody can. That gap — between decades of failed attempts and a rigorous proof of impossibility — is exactly where the Millennium Prize question lives.

TH

รายการของ Karp อ่านเหมือนทัวร์คณิตศาสตร์ประยุกต์: กราฟหนึ่งระบายสีด้วยจำนวนสีที่กำหนด โดยไม่ให้โหนดสองโหนดที่เชื่อมกันใช้สีเดียวกัน ได้หรือไม่; ชุดของสิ่งของหนึ่งแบ่งเป็นสองกองที่มีมูลค่ารวมเท่ากันได้หรือไม่; เครือข่ายหนึ่งถูกรอบคลุมด้วยชุดโหนดที่เลือกมาไม่กี่โหนดที่แตะทุกเส้นเชื่อมได้หรือไม่; สูตรตรรกะแบบบูลในรูปจำกัดรูปหนึ่งสอดคล้องได้หรือไม่ นักทฤษฎีกราฟ นักวิจัยดำเนินงาน และนักตรรกวิทยาโจมตีปัญหาเหล่านี้แยกกัน และไม่มีเหตุผลอะไรเป็นพิเศษให้คิดว่าความยากลำบากของแต่ละฝ่ายเชื่อมโยงกัน โช้การลดรูปของ Karp เผยว่าทั้งหมดคือโรคเดียวกันที่สวมเครื่องแต่งกายยี่สิบเอ็ดชุด — การค้นพบแบบเดียวกับที่บทที่ 04 เฉลิมฉลองในไม้อัดของ Kantorovich และสูตรอาหารของ Stigler พอดี เว้นแต่ว่าตรงนี้อัตลักษณ์ร่วมคือคำสาปร่วม ไม่ใช่ทางลัดร่วม พบอัลกอริทึมที่มีประสิทธิภาพสำหรับข้อใดข้อหนึ่ง ก็เท่ากับพบสำหรับทุกข้อ และสำหรับทุกปัญหาใน NP ด้วย ไม่เคยมีใครพบ ไม่เคยมีใครพิสูจน์ว่าไม่มีใครพบได้ ช่องว่างนั้น — ระหว่างความพยายามที่ล้มเหลวมาหลายทศวรรษ กับบทพิสูจน์ความเป็นไปไม่ได้อย่างเข้มงวด — คือที่ที่คำถามรางวัลมิลเลนเนียมอาศัยอยู่พอดี

The Million-Dollar Question

คำถามหนึ่งล้านดอลลาร์

EN

In 2000, the Clay Mathematics Institute selected seven problems it judged to be the deepest open questions in mathematics and placed a one-million-dollar prize on the first correct solution to each. P versus NP is one of the seven — formally, does every problem whose solutions can be efficiently *verified* also admit an efficient method of *discovery*? — and as of this writing it remains unsolved, one of six of the original seven still standing; only the Poincaré conjecture has fallen, to Grigori Perelman in the early 2000s (Clay Mathematics Institute, "P vs NP"; Wikipedia, "Millennium Prize Problems").

TH

ปี 2000 Clay Mathematics Institute เลือกปัญหาเจ็ดข้อที่ตัดสินว่าเป็นคำถามเปิดที่ลึกที่สุดในคณิตศาสตร์ และตั้งรางวัลหนึ่งล้านดอลลาร์ให้คำตอบที่ถูกต้องคำตอบแรกของแต่ละข้อ P versus NP เป็นหนึ่งในเจ็ดข้อนั้น — พูดอย่างเป็นทางการคือ ปัญหาทุกข้อที่ตรวจสอบคำตอบได้อย่างมีประสิทธิภาพ ยอมให้มีวิธีค้นพบที่มีประสิทธิภาพด้วยหรือไม่ — และ ณ เวลาที่เขียนอยู่นี้ยังไม่มีใครแก้ได้ มันเป็นหนึ่งในหกข้อจากเจ็ดข้อเดิมที่ยังยืนอยู่; มีเพียงข้อความคาดการณ์ Poincaré เท่านั้นที่ล้มลงโดยฝีมือของ Grigori Perelman ช่วงต้นทศวรรษ 2000 (Clay Mathematics Institute, "P vs NP"; Wikipedia, "Millennium Prize Problems")

E N

The traveling salesman problem has become this chapter's emblem, and for good reason: it is easy enough to state to a child (find the shortest possible route visiting every city on a list exactly once and returning home) and hard enough that no one has ever found a way to solve it exactly, for large instances, that beats trying an amount of work growing explosively with the number of cities. It sits, alongside SAT and Karp's twenty-one, on the far side of the wall — not proven impossible to solve quickly, but proven exactly as hard as every other problem in the same enormous, unresolved family.

T H

ปัญหาพนักงานขายเดินทาง (traveling salesman problem) กลายเป็นสัญลักษณ์ของบทนี้ ด้วยเหตุผลที่ดี: มันง่ายพอจะเล่าให้เด็กฟัง (หาเส้นทางที่สั้นที่สุดเท่าที่จะเป็นไปได้ ที่แวะทุกเมืองในรายการเมืองละครั้งพอดีแล้วกลับบ้าน) และยากพอจนไม่เคยมีใครหาวิธีแก้มันแบบตรงเป๊ะสำหรับอินสแตนซ์ขนาดใหญ่ ที่ชนะการลองด้วยปริมาณงานที่โตระเบิดตามจำนวนเมืองได้ มันนั่งอยู่ข้างๆ SAT และยี่สิบเอ็ดข้อของ Karp บนอีกฟากของกำแพง — ไม่มีใครพิสูจน์ว่ามันแก้เร็วไม่ได้ แต่มีบทพิสูจน์ว่ามันยากเท่ากับปัญหาอื่นทุกข้อในตระกูลมึนๆ ที่ยังไม่มีข้อยุติตระกูลเดียวกัน เป๊าะๆ

E N

It is worth pausing to be honest about what "the wall" actually claims, because it is easy to overstate. Nobody has proven $P \neq NP$. It remains logically possible — though almost no computer scientist believes it — that some ingenious algorithm for SAT is waiting to be found, and that finding it would efficiently solve every problem on Karp's list and every problem like them at a stroke. What has been proven, with total rigor, is the *equivalence*: solve one member of this vast family efficiently and you have solved them all. The wall may or may not be permanent. What is certain is that it is the same wall, everywhere you press against it.

T H

"กำแพง" อ้างอะไรกันแน่ ควรหยุดพูดตามตรงสักครู่ เพราะมันเปลืองอ้าเงินจริงได้ง่าย ไม่มีใครพิสูจน์ว่า $P \neq NP$ ยังเป็นไปได้ในเชิงตรรกะ — แม้แทบไม่มีนักวิทยาการคอมพิวเตอร์คนไหนเชื่อ — ว่าอัลกอริทึมอันชาญฉลาดสักตัวสำหรับ SAT กำลังรอให้ค้นพบอยู่ และการพบมันจะแก้ทุกปัญหาในรายการของ Karp และทุกปัญหาแบบเดียวกันได้อย่างมีประสิทธิภาพในคราวเดียว สิ่งที่พิสูจน์แล้วอย่างเข้มงวดเต็มทีคือความสมมูล: แก่สมาชิกหนึ่งตัวของตระกูลมี้มานี้ได้อย่างมีประสิทธิภาพ ก็เท่ากับแก้ได้ทั้งตระกูล กำแพงนี้จะถาวรหรือไม่ ยังไม่มีใครรู้ สิ่งที่น่าเชื่อมันเป็นกำแพงเดียวกัน ไม่ว่าจะไปดันตรงไหน

Life at the Wall

ชีวิตที่เชิงกำแพง

EN

If the theory stopped here, this would be a chapter about despair: a proof that a huge swath of practically important problems cannot be solved exactly and efficiently, full stop. What actually happened next is more interesting, and it is the part of this chapter's story that keeps this book's central thesis alive even at its darkest point — because solvers, faced with a wall, did not stop. They found doors.

TH

ถ้าทฤษฎีหยุดอยู่แค่นี้ บทนี้ก็จบที่ด้วยความสิ้นหวัง: บทพิสูจน์ว่าปัญหาสำคัญในทางปฏิบัติจำนวนมากแก้แบบตรงเป๊ะและมีประสิทธิภาพไม่ได้ จบ สิ่งที่เกิดขึ้นจริงต่อจากนั้นน่าสนใจกว่า และมันคือส่วนของเรื่องเล่าในบทนี้ที่ทำให้ข้อเสนองใจกลางของหนังสือเล่มนี้ยังมีชีวิตอยู่แม้ในจุดที่มืดที่สุด — เพราะนักแก้ปัญหาที่เจอกำแพงไม่ได้หยุด พวกเขาหาประตูเจอ

EN

The first door is **approximation**: give up on finding the provably optimal answer and settle, deliberately, for a solution guaranteed to be within some fixed, known factor of optimal. In 1976, Nicos Christofides published an algorithm for a common variant of the traveling salesman problem — one where the triangle inequality holds, meaning a direct route between two cities is never longer than a detour through a third — that is guaranteed to find a tour no more than 50 percent longer than the true shortest possible tour, no matter the instance. Remarkably, more than four decades later, Christofides' 1.5-factor guarantee for this problem had still not been beaten by any published algorithm as of the early 2020s, a striking testament to how hard even *bounded* progress on an NP-complete problem can be (arXiv, "A historical note on the $3/2$ -approximation algorithm for the metric traveling salesman problem"). An approximation algorithm does not defeat the wall. It negotiates terms with it: not the best answer, but a *provably* good-enough one, delivered in a time that does not explode.

TH

ประเด็นแรกคือการประมาณ (approximation): เลิกหาคำตอบที่พิสูจน์ได้ว่าเหมาะสมที่สุด แล้วตั้งใจยอมรับคำตอบที่รับประกันว่าห่างจากค่าที่เหมาะสมที่สุดไม่เกินตัวคูณค่าหนึ่งที่ยัดตัวและรู้ค่า ปี 1976 Nicos Christofides ตีพิมพ์อัลกอริทึมสำหรับปัญหาพนักงานขายเดินทางรูปแบบที่พบบ่อยแบบหนึ่ง — แบบที่อสมการสามเหลี่ยมเป็นจริง คือเส้นทางตรงระหว่างสองเมืองไม่มีวันยาวกว่าการอ้อมผ่านเมืองที่สาม — อัลกอริทึมนี้รับประกันว่าจะหาทัวร์ที่ยาวกว่าทัวร์ที่สั้นที่สุดจริงไม่เกิน 50 เปอร์เซ็นต์ ไม่ว่าอินสแตนซ์จะเป็นอย่างไร กว่าสี่ทศวรรษให้หลัง การรับประกันตัวคูณ 1.5 ของ Christofides สำหรับปัญหานี้ยังไมถูกอัลกอริทึมที่ตีพิมพ์ตัวไหนโค่นได้เลย ณ ช่วงต้นทศวรรษ 2020 ตัวเลขนั้นบอกเองว่าความคืบหน้าแม้แต่แบบมีขอบเขตบนปัญหา NP-complete ยากเพียงใด (arXiv, "A historical note on the 3/2-approximation algorithm for the metric traveling salesman problem") อัลกอริทึมประมาณไม่ได้เอาชนะกำแพง มันเจรจาเงื่อนไขกับกำแพง: ไม่ใช่คำตอบที่ดีที่สุด แต่เป็นคำตอบที่พิสูจน์ได้ว่าดีพอ ส่งมอบในเวลาที่ไม่ระเบิด

EN

The second door is **randomization** — the same willingness to trade certainty for speed that chapter 04 met in Monte Carlo methods, applied here to combinatorial search rather than physical simulation. A randomized algorithm makes some of its choices by coin flip rather than exhaustive comparison, and for many hard problems this yields algorithms that are fast on average and wrong only with controllable, vanishingly small probability — a guaranteed-correct answer traded for an almost-certainly-correct one delivered far sooner. The clearest everyday example sits, fittingly, inside the very cryptography this chapter is about to turn to: before RSA can lock anything, it first needs two very large prime numbers, and testing an enormous candidate number for primality by trial division is itself hopelessly slow. The Miller–Rabin test, devised by Gary Miller in 1976 and made fully practical by Michael Rabin shortly after, picks random witnesses and checks each one against the candidate number; a composite number will be caught with at least even odds by any single random witness, so after fifty independent draws the chance of a wrong answer falls below one in a quadrillion — small enough that every RSA key generated today is built on primes accepted by a

test that is not, strictly, certain, only certain enough (Cornell CS4820, "The Miller-Rabin Randomized Primality Test").

TH

ประตูบานที่สองคือ**การสุ่ม (randomization)** — ความเต็มใจแลกความแน่นอนกับความเร็วแบบเดียวกับที่บทที่ 04 พบในวิธีมอนติคาร์โล คราวนี้ใช้กับการค้นหาเชิงการจัดหมู่แทนการจำลองทางกายภาพ อัลกอริทึมแบบสุ่มตัดสินใจบางส่วนด้วยการโยนเหรียญแทนการเทียบทุกทางเลือก สำหรับปัญหาที่ยากหลายข้อ วิธีนี้ให้อัลกอริทึมที่เร็วโดยเฉลี่ยและผิดด้วยความน่าจะเป็นที่ควบคุมได้และเล็กจนแทบหายไป — แลกคำตอบที่รับประกันว่าถูก กับคำตอบที่แทบจะแน่นอนว่าถูกและมาถึงเร็วกว่ามาก ตัวอย่างในชีวิตประจำวันที่สุดที่อยู่ในวิทยาการรหัสลับที่ตอนนี้กำลังจะหันไปพูดถึงพอดี และก็เหมาะเจาะดี: ก่อน RSA จะล็อกอะไรได้ มันต้องมีจำนวนเฉพาะขนาดใหญ่มาสองจำนวนก่อน และการทดสอบว่าจำนวนมหึมาที่เป็นตัวเลือกเป็นจำนวนเฉพาะหรือไม่ด้วยการหารไล่ ตัวมันเองก็ช้าจนหมดหวัง การทดสอบ Miller-Rabin ที่ Gary Miller คิดขึ้นในปี 1976 และ Michael Rabin ทำให้ใช้งานได้จริงเต็มที่ในเวลาไม่นานหลังจากนั้น เลือกรหัสแบบสุ่มขึ้นมาแล้วตรวจแต่ละตัวกับจำนวนที่เป็นตัวเลือก; พยายามสุ่มตัวเดียวจับจำนวนประกอบได้ด้วยโอกาสอย่างน้อยครึ่งต่อครึ่ง ดังนั้นหลังจับพยานอิสระห้าสิบตัว โอกาสได้คำตอบผิดตกลงต่ำกว่าหนึ่งในพันล้านล้าน — เล็กพอจนถูกใจ RSA ทุกดอกที่สร้างขึ้นทุกวันนี้ตั้งอยู่บนจำนวนเฉพาะที่ผ่านการทดสอบที่พูดอย่างเคร่งครัดแล้วไม่ได้แน่นอน แค่อนุมูล (Cornell CS4820, "The Miller-Rabin Randomized Primality Test")

EN

The third door is the most surprising, and it is the one modern computing quietly walks through every day without most of its users ever noticing. In the decades since Cook and Levin's theorem, engineers built increasingly sophisticated SAT solvers — programs that attempt to find a satisfying assignment for exactly the problem Cook proved NP-complete in the first place. The modern generation of these tools, built around a technique called conflict-driven clause learning (CDCL), does something that should, on paper, be impossible: it routinely solves real-world SAT instances with *millions* of variables and clauses, arising from chip verification, software testing, and scheduling, in practical amounts of time (arXiv, "Relating Com-

plexity-theoretic Parameters with SAT Solver Performance"). CDCL works by learning from its own failures mid-search: every time it reaches a dead end, it derives a new logical constraint — a "clause" — summarizing *why* that dead end occurred, adds it permanently to the problem, and uses it to prune away every future branch that would repeat the same mistake for the same underlying reason. Over the course of a long search, the solver effectively rewrites the problem it is solving, filling it with accumulated scar tissue from every wrong turn, and the accumulated scars turn out to make the remaining search dramatically smaller. This does not contradict the Cook–Levin theorem. Worst-case instances of SAT, carefully constructed to defeat exactly this kind of learning, remain exponentially hard, and always will unless P equals NP. But the *industrial* instances that show up in real chip designs and real schedules are, empirically, richly structured rather than adversarially worst-case, and researchers have spent years trying to characterize exactly what kind of hidden structure — community clustering, "backdoor" variable sets, low treewidth — CDCL solvers are quietly exploiting to walk straight through problems that complexity theory says should be untouchable. The wall is real, in the worst case, for every instance anyone could construct. It is also, for the instances that actually show up in practice, full of doors nobody predicted would be there.

TH

ประตูปานที่สามน่าประหลาดใจที่สุด และเป็นบานที่คอมพิวเตอร์สมัยใหม่เดินผ่านเจียบๆ ทุกวันโดยที่ผู้ใช้ส่วนใหญ่ไม่เคยสังเกต ในหลายทศวรรษหลังทฤษฎีบทของ Cook กับ Levin วิศวกรสร้าง SAT solver ที่ซับซ้อนขึ้นเรื่อยๆ — โปรแกรมที่พยายามหาการกำหนดค่าที่ทำให้สูตรเป็นจริง สำหรับปัญหาที่ Cook พิสูจน์ว่าเป็น NP-complete ตั้งแต่แรกนั่นเอง เครื่องมือรุ่นใหม่ที่สร้างรอบเทคนิคชื่อ conflict-driven clause learning (CDCL) ทำสิ่งที่บนกระดาษควรเป็นไปได้ มันแก้อินสแตนซ์ SAT ในโลกจริงที่มีตัวแปรและอนุประโยคเป็นล้าน จากการตรวจสอบชิป การทดสอบซอฟต์แวร์ และการจัดตาราง ได้เป็นปกติ ในเวลาที่ใช้งานได้จริง (arXiv, "Relating Complexity-theoretic Parameters with SAT Solver Performance") CDCL เรียนรู้จากความล้มเหลวของตัวเองระหว่างค้นหา: ทุกครั้งทีไปถึงทางตัน มันอนุมานข้อจำกัดทางตรรกะข้อใหม่ — "อนุประโยค (clause)" — ที่สรุปว่าทำไมทางตันนั้นถึงเกิด เพิ่มมันเข้าไปในปัญหาอย่างถาวร แล้วใช้มันตัดกิ่งทุกกิ่งในอนาคตที่จะผิดซ้ำ

แบบเดิมด้วยเหตุผลพื้นฐานเดียวกันทั้ง ตลอดการค้นหาที่ยาวนาน ตัวแก้ปัญหาก็พบกับเขียนปัญหาที่มันกำลังแก้ขึ้นใหม่ เติมน้ำมันด้วยเนื้อแฟลเป็นทีละสมจากทุกการเลี้ยวผิด และแฟลเป็นทีละสมนั้นกลับทำให้การค้นหาที่เหลือเล็กลงมาก เรื่องนี้ไม่ได้ขัดกับทฤษฎีบท Cook–Levin อินสแตนซ์กรณีเลวร้ายที่สุดของ SAT ที่ถูกสร้างขึ้นอย่างพิถีพิถันเพื่อเอาชนะการเรียนรู้แบบนี้พอดี ยังคงยากแบบเอกซ์โพเนนเชียล และจะยากอย่างนั้นตลอดไป เว้นแต่ P จะเท่ากับ NP แต่อินสแตนซ์ระดับอุตสาหกรรมที่โผล่มาในการออกแบบชิปจริงและตารางงานจริง จากการวัดจริงมีโครงสร้างหนาแน่น ไม่ใช่กรณีเลวร้ายที่สุดที่ถูกออกแบบมาเล่นงาน นักวิจัยใช้เวลาหลายปีพยายามระบุให้ได้ว่าโครงสร้างที่ซ่อนอยู่ชนิดไหนกันแน่ — การจับกลุ่มเป็นชุมชน ชุดตัวแปร "ประตูหลัง" ค่า treewidth ต่ำ — ที่ SAT solver แบบ CDCL แอบใช้ประโยชน์อยู่เงียบๆ เพื่อเดินทะลุปัญหาที่ทฤษฎีความซับซ้อน (complexity) บอกว่าควรจะต้องไม่ได้ กำแพงนั้นเป็นของจริง ในกรณีเลวร้ายที่สุด สำหรับทุกอินสแตนซ์ที่ใครก็ตามจะสร้างขึ้นได้ และสำหรับอินสแตนซ์ที่โผล่มาในทางปฏิบัติจริงๆ มันก็เต็มไปด้วยประตูที่ไม่มีใครทำนายว่าจะอยู่ตรงนั้น

The Wall as Fortress

กำแพงในฐานะป้อมปราการ

EN

The final turn in this chapter's story inverts everything that came before it, and it is the most consequential inversion in this book. Every solver so far has treated hardness as an obstacle: a wall to be reduced away from, approximated around, or randomized through. In 1977, three MIT researchers — Ron Rivest, Adi Shamir, and Leonard Adleman — realized that hardness could instead be the *product*.

TH

จุดหักครั้งสุดท้ายในเรื่องเล่าของบทนี้กลับทุกอย่างที่มาก่อนหน้า และเป็นการกลับด้านที่ส่งผลมากที่สุดในหนังสือเล่มนี้ นักแก้ปัญหายุคก่อนจนถึงตอนนี้ปฏิบัติกับความยาก (hardness) เหมือนอุปสรรค: กำแพงที่ต้องลดรูปหนีออกมา ประมาณอ้อมไป หรือสุมทะลุผ่าน ปี 1977 นักวิจัย MIT สามคน — Ron Rivest, Adi Shamir และ Leonard Adleman — ตระหนักว่าความยากเป็นตัวสินค้าได้แทน

E N

Their cryptosystem, which took the researchers' initials as its name, RSA, is built on a specific, carefully chosen instance of computational hardness: multiplying two very large prime numbers together is fast, taking a computer a fraction of a second even for numbers hundreds of digits long, but *factoring* the resulting product back into its two original primes, without already knowing them, has no known efficient method — you are back to something resembling brute-force search through an astronomically large space of candidate factors (Splunk, "RSA Algorithm in Cryptography: Rivest Shamir Adleman Explained"). Anyone can publish the product — the "public key" — and use it to lock a message. Only someone who knows the two original primes — the "private key" — can unlock it efficiently; everyone else faces a computational wall as real as any traveling salesman's. RSA is, technically, believed to rely on the hardness of integer factorization rather than being proven NP-complete in the strict Cook-Levin sense — factorization sits in a curious, not-fully-resolved corner of complexity theory, apparently hard but not known to be NP-complete itself. What matters for this chapter's purposes is the philosophical hinge, not the technical fine print: an entire branch of applied mathematics discovered that a wall no one can climb is exactly as valuable as a treasure no one can steal. Every secure connection to a bank, a hospital record, or an encrypted message rests, at bottom, on the same kind of hardness that makes the traveling salesman problem resist every solver's cleverness. One field's dead end is another field's foundation.

T H

ระบบรหัสลับของพวกเขา คือ RSA ใช้อักขระย่อของนักวิจัยทั้งสามเป็นชื่อ และสร้างขึ้นบนอนิสแดนซ์ของความยากเชิงการคำนวณที่เจาะจงและเลือกมาอย่างพิถีพิถัน: คูณจำนวนเฉพาะขนาดใหญ่สองจำนวนเข้าด้วยกันนั้นเร็ว คอมพิวเตอร์ใช้เวลาเสี้ยววินาทีแม้กับจำนวนที่ยาวหลายร้อยหลัก แต่การแยกตัวประกอบผลคูณที่ได้กลับไปเป็นจำนวนเฉพาะสองจำนวนตั้งต้น โดยไม่รู้มันอยู่ก่อน ไม่มีวิธีที่มีประสิทธิภาพที่รู้จักเลย — กลับไปสู่อะไรที่คล้ายการค้นหาแบบไล่ทุกทาง (brute force) ผ่านพื้นที่ของตัวประกอบที่เป็นไปได้ที่ใหญ่ระดับดาราศาสตร์ (Splunk, "RSA Algorithm in Cryptography: Rivest Shamir Adleman Explained") ใครก็ประกาศผลคูณนั้นได้ — "กุญแจสาธารณะ" — และใช้มันล็อกข้อความ มีเพียงคนที่รู้จำนวนเฉพาะตั้งต้น

สองจำนวน — "กุญแจส่วนตัว" — เท่านั้นที่ปลดล็อกได้อย่างมีประสิทธิภาพ; คนอื่นทั้งหมดเจอกำแพงเชิงการคำนวณที่เป็นของจริงพอๆ กับกำแพงของพนักงานขายเดินทาง ในทางเทคนิค เชื่อกันว่า RSA พึ่งความยากของการแยกตัวประกอบจำนวนเต็ม มากกว่าจะมีบทพิสูจน์ว่าเป็น NP-complete ในความหมายเคร่งครัดแบบ Cook-Levin — การแยกตัวประกอบนั้นอยู่ในมุมที่แปลกและยังไม่มีข้อยุติเต็มที่ของทฤษฎีความซับซ้อน ดูจะยาก แต่ไม่มีใครรู้ว่าตัวมันเองเป็น NP-complete หรือไม่ สิ่งที่สำคัญสำหรับเป้าหมายของบทนี้คือบานพับทางปรัชญา ไม่ใช่ตัวหนังสือตัวเล็กเชิงเทคนิค: คณิตศาสตร์ประยุกต์ทั้งสาขาค้นพบว่ากำแพงที่ไม่มีใครปีนได้ มีค่าเท่ากับสมบัติที่ไม่มีใครขโมยได้พอดี ทุกการเชื่อมต่อที่ปลอดภัยไปยังธนาคาร เวชระเบียนของโรงพยาบาล หรือข้อความที่เข้ารหัส ล้วนตั้งอยู่บนความยากชนิดเดียวกันกับที่ทำให้ปัญหาพนักงานขายเดินทางด้านความฉลาดของนักแก้ปัญหาทุกคน ทางต้นของสาขาหนึ่งคือรากฐานของอีกสาขาหนึ่ง

Hardness Is Information

ความยากคือข้อมูล

EN

Step back, and this chapter completes something chapter 00 promised and chapter 04 and 05 built toward: the honest limits of this book's central claim. "Great solvers do not search harder; they change the space until the path is short" has held, gloriously, across every chapter so far — Euler's dots, Blackett's convoys, Dijkstra's frontier, A's *admissible optimism*. NP-completeness is this book's proof that the claim has a genuine boundary. Some spaces, as far as anyone has ever proven or is likely to prove soon, cannot be compressed by any representation change, however clever — not because no one has been clever enough yet, but because Cook, Levin, and Karp showed that compressing any one of them efficiently would compress all* of them efficiently, and seventy years of the finest minds in computer science trying to do exactly that have come back empty-handed.

TH

บทนี้ปิดสิ่งที่บทที่ 00 สัญญาไว้ และบทที่ 04 กับ 05 ปูทางมาถึง: ขีดจำกัดที่ซื่อสัตย์ของข้ออ้างใจกลางของหนังสือเล่มนี้ ถอยออกมามองจะเห็นชัด "นักแก้ปัญหาที่

ยิ่งใหญ่ไม่ค้นหาหนักขึ้น — เขาเปลี่ยนพื้นที่จนเส้นทางสั้นลง" ยืนอยู่ได้ตลอดทุกบทที่ผ่านมา — จุดของ Euler กองเรือคุ้มกันของ Blackett แนวหน้าของ Dijkstra การมองโลกในแง่ดีแบบ admissible ของ A\ NP-completeness คือบทพิสูจน์ของเล่มนี้ว่าข้ออ้างนั้นมีขอบเขตจริง พื้นที่บางแบบบีบอัดไม่ได้ด้วยการเปลี่ยนรูปแบบการแทนปัญหา (representation) ใดๆ ไม่ว่าจะฉลาดแค่ไหน เท่าที่ใครเคยพิสูจน์ได้หรือน่าจะพิสูจน์ได้ในเร็ววัน เหตุผลไม่ใช่ว่ายังไม่มีใครฉลาดพอ Cook, Levin และ Karp แสดงให้เห็นว่าถ้าบีบอัดพื้นที่ข้อใดข้อหนึ่งได้อย่างมีประสิทธิภาพ ก็จะบีบอัดทุกข้อ*ได้อย่างมีประสิทธิภาพไปด้วย และเจ็ดสิบปีที่สมมติดีที่สุดในวิทยาการคอมพิวเตอร์พยายามทำเรื่องนั้นพอดี ก็กลับมามือเปล่า

EN

But notice what kind of knowledge that failure actually is. Knowing that a problem sits behind the NP-complete wall is not the absence of a solution — it is a *piece of information about the problem*, as concrete and useful as knowing its size or its symmetries. It tells you, immediately, not to waste a career hunting for an exact, always-fast algorithm. It tells you to reach instead for approximation, randomization, or a solver built to exploit whatever real-world structure your instances actually have, rather than their theoretical worst case. And, in RSA's case, it tells you the wall itself might be worth more standing than knocked down. Naming the wall is itself an act of solving — it is the last, hardest-won representation change this section of the book has to offer: reframing "I cannot find a fast algorithm" from a personal failure into a proven structural fact about the space, and then choosing, with open eyes, which door to walk through instead.

TH

แต่ให้สังเกตว่าความล้มเหลวนั้นเป็นความรู้ชนิดไหนกันแน่ การรู้ว่าปัญหาข้อหนึ่งอยู่หลังกำแพง NP-complete ไม่ใช่การไม่มีคำตอบ — มันคือชิ้นข้อมูลเกี่ยวกับปัญหานั้น เป็นรูปธรรมและใช้ได้จริงพอๆ กับการรู้ขนาดหรือความสมมาตรของมัน มันบอกทันทีว่าอย่าผลาญทั้งอาชีพไปกับการล่าอัลกอริทึมที่ตรงเป๊ะและเร็วเสมอ มันบอกให้เอ๋อไปหาการประมาณ การสุ่ม หรือตัวแก้ที่สร้างมาเพื่อใช้ประโยชน์จากโครงสร้างในโลกจริงเท่าที่อื่นสแตนด์ของคุณมีอยู่จริง แทนกรณีเลวร้ายที่สุดในทางทฤษฎีของมัน และในกรณีของ RSA มันบอกว่กำแพงที่ยังยืนอยู่อาจมีค่ามากกว่ากำแพงที่ถูก

ทุบลง การเรียกชื่อกำแพงเป็นการแก้ปัญหในตัวเอง — มันคือการเปลี่ยน representation ครั้งสุดท้ายและได้มายากที่สุดที่ส่วนนี้ของหนังสือมีให้: เปลี่ยนกรอบ "ฉันทาอัลกอริทึมที่เร็วไม่ได้" จากความล้มเหลวส่วนตัว ให้เป็นข้อเท็จจริงเชิงโครงสร้างที่พิสูจน์แล้วเกี่ยวกับพื้นที่ แล้วเลือกด้วยตาที่เปิดกว้างว่าจะเดินผ่านประตูบานไหนแทน

The Move

ทำประจำบท

EN

Move 06 — Name the wall, then trade: reduce, approximate, randomize, or repurpose the hardness.

TH

Move 06 — เรียกชื่อกำแพง แล้วแลก: ลดรูป ประมาณ สุ่ม หรือเอาความยากไปใช้ประโยชน์

EN

- **Before searching for years, check whether your problem is a known monster in disguise.** A reduction from a Karp-list problem, or from SAT itself, is proof that no one else has found a fast general solution either — that is worth knowing before, not after, a career's worth of failed attempts.
- **When exact and fast cannot both be had, choose which one to give up, deliberately.** Christofides' bound trades exactness for a guaranteed-good approximation; Monte Carlo-style randomization trades certainty for overwhelming probability. Both are engineering decisions, not defeats.
- **Check whether your actual instances are the adversarial worst case or the friendly common case.** CDCL solvers prove that "NP-complete in theory" and "intractable in your specific data" are different claims; industrial SAT is proof that real problems are often far kinder than the theorems that bound them.

- **Ask whether the hardness itself has value.** RSA shows that a wall no one can climb can be turned into a fortress. Before treating intractability as pure obstacle, ask who would pay for a problem that stays hard forever.

TH

- ก่อนจะค้นหายุ่หลายปี ตรวจสอบว่าปัญหาของคุณคือสัตว์ประหลาดที่รู้จักกันดี ซึ่งปลอมตัวมาหรือเปล่า การลดรูปจากปัญหาในรายการของ Karp หรือจาก SAT เอง คือหลักฐานว่าไม่มีใครอื่นหาคำตอบทั่วไปที่เร็วเจอเหมือนกัน — เรื่องนี้ควรรู้ก่อน ไม่ใช่หลัง ความพยายามที่ล้มเหลวคุ้มค่าทั้งอาชีพ
- เมื่อตรงกับเร้ามีพร้อมกันไม่ได้ จงเลือกอย่างตั้งใจว่าจะสละอันไหน ขอบเขตของ Christofides แลกความตรงเป๊ะกับการประมาณที่รับประกันว่าดี; การสุ่มแบบมอนติคาร์โลแลกความแน่นอนกับความน่าจะเป็นที่ท่วมท้น ทั้งคู่คือการตัดสินใจเชิงวิศวกรรม ไม่ใช่ความพ่ายแพ้
- ตรวจสอบว่าอินสแตนซ์จริงของคุณคือกรณีเลวร้ายที่สุดที่ถูกออกแบบมาเล่นงาน หรือกรณีทั่วไปที่เป็นมิตร SAT solver แบบ CDCL พิสูจน์ว่า "NP-complete ในทางทฤษฎี" กับ "จัดการไม่ไหว (intractable) ในข้อมูลเฉพาะของคุณ" เป็นคนละข้ออ้างกัน; SAT ระดับอุตสาหกรรมคือหลักฐานว่าปัญหาลงใจดีกว่าทฤษฎีบทที่ตีกรอบมันไว้มาก
- ถามว่าความยากนั้นเองมีค่าหรือไม่ RSA แสดงให้เห็นว่ากำแพงที่ไม่มีใครปีนได้ เปลี่ยนเป็นป้อมปราการได้ ก่อนจะปฏิบัติกับความจัดการไม่ไหวเหมือนอุปสรรคล้วนๆ ให้ถามว่าใครจะยอมจ่ายเงินซื้อปัญหาที่ยากอยู่อย่างนั้นตลอดไป

EN

Next — Chapter 07 · Gallery I: The Single Move: theory now gives way to a gallery. Five solvers, four centuries, one move each — the history this book has built in the abstract, made concrete, one great solve at a time.

TH

ต่อไป — บทที่ 07 · หอเรื่องเล่า I: ทำเดียวจบ: ทฤษฎีถึงคราวหลีกทางให้หอเรื่องเล่า นักแก้ปัญหาห้าคน สี่ศตวรรษ คนละหนึ่งท่า — ประวัติศาสตร์ที่หนังสือเล่มนี้ก่อไว้ในเชิงนามธรรม ถูกทำให้เป็นรูปธรรม ที่ละการแก้ครั้งใหญ่

CHAPTER 07

Gallery I: The Single Move

หอเรื่องเล่า I: ทำเดียวจบ

EN

Five solves, four centuries, one pattern. Each of the people in this gallery faced a problem that looked, on the day they met it, like it demanded more of what had already failed — more travel, more staring at the sky, more scrubbing, more flying. Each instead found a single point of leverage that made the hard version of the problem unnecessary. Watch for the moment it happens in each story: it is almost always small, almost always fast, and almost always preceded by years in which nobody thought to look for it.

TH

ถ้าการแก้สี่ศตวรรษ หนึ่งรูปแบบ ปัญหาที่ทุกคนในหอนี้เจอ ในวันแรกดูเหมือนเรียกร้องสิ่งเดิมที่ล้มเหลวไปแล้วให้มากขึ้น — เดินทางให้ไกลขึ้น จ้องฟ้าให้นานขึ้น ขัดล้างให้หนักขึ้น บินให้บ่อยขึ้น แต่ทุกคนกลับเจอจุดคานงัดจุดเดียว จุดที่ทำให้ปัญหาเวอร์ชันยากไม่จำเป็นอีกต่อไป จับตาจังหวะนั้นในแต่ละเรื่อง มันมักเล็ก มักเร็ว และมักมีปีอันยาวนานนำหน้า ปีที่ไม่มีใครคิดจะมองหามัน

A Well at Noon — Alexandria and Syene, c. 240 BC

บ่อน้ำตอนเที่ยง — อะเล็กซานเดรียกับ Syene ราว 240 ปีก่อนคริสตกาล

EN

Eratosthenes of Cyrene ran the Library of Alexandria, the largest archive of the ancient world, and he had heard a piece of travelers' lore that most scholars would have filed away as trivia: in the town of Syene, far up the Nile at modern Aswan, there was a deep well into which the sun shone straight down to the water at noon on the summer solstice, casting no shadow at all (APS News, "Eratosthenes Measures Earth"). Nowhere in Alexandria did that happen. A vertical stick — a gnomon — planted in Alexandrian ground at the same hour on the same day always threw a shadow.

TH

Eratosthenes แห่ง Cyrene ดูแลหอสมุดแห่งอะเล็กซานเดรีย คลังเอกสารที่ใหญ่ที่สุดของโลกยุคโบราณ และเขาได้ยินเรื่องเล่าของนักเดินทางขึ้นหนึ่งที่นักปราชญ์ส่วนใหญ่คงเก็บเข้าลิ้นชักเป็นเกร็ดไร้สาระ: ที่เมือง Syene เหนือขึ้นไปตามแม่น้ำไนล์ ตรงที่วันนี้คืออัสวาน มีบ่อน้ำลึกบ่อหนึ่ง ตอนเที่ยงของวันครีษมายัน ดวงอาทิตย์ส่องถึงลงไปถึงผิวน้ำ ไม่ทิ้งเงาไว้เลยแม้แต่หยด (APS News, "Eratosthenes Measures Earth") ในอะเล็กซานเดรียไม่มีที่ไหนเป็นแบบนั้น ไม่ได้จากอันหนึ่ง — แท่งวัดเงา (gnomon) — ปักลงบนพื้นอะเล็กซานเดรียในช่วงโมงเดียวกันของวันเดียวกัน ทอดเงาออกมาเสมอ

EN

The obvious way to find out how large the Earth is would be to go around it, or at least a good long way across it, and measure. Nobody in the third century BC could do that. Eratosthenes realized he did not need to. If the sun's rays arrive at the Earth essentially parallel — a reasonable assumption, given the sun's enormous distance — then the difference between "no shadow at Syene" and "some shadow at Alexandria" is not a fact about two towns. It is a direct reading of the *curvature of the Earth between them*, expressed as an angle. He measured that angle at Alexandria: about 7.2 degrees, one-fiftieth of a full circle. He then needed only one more number — the ground distance between the two towns — which he obtained from *bematists*, professional pacers trained to walk a fixed stride and report distance by

counting steps, who gave him roughly 5,000 stadia (APS News). Multiply 5,000 by 50, because the measured angle was one-fiftieth of the full circle the Earth's circumference would trace, and the Earth is 250,000 stadia around.

T H

Eratosthenes เห็นว่าเขาไม่ต้องเดินทางเลย วิธีที่ซัดที่สุดในการหาว่าโลกใหญ่แค่ไหน คือเดินรอบมัน หรืออย่างน้อยตัดข้ามไปให้ไกลพอแล้ววัด ไม่มีใครในศตวรรษที่สามก่อนคริสตกาลทำแบบนั้นได้ แต่ถ้าวัดเส้นของดวงอาทิตย์มาถึงโลกในแนวขนานกันโดยประมาณ — ข้อสมมติที่สมเหตุสมผล เมื่อดวงอาทิตย์อยู่ไกลขนาดนั้น — ความต่างระหว่าง "ไม่มีเงาที่ Syene" กับ "มีเงาที่อะเล็กซานเดรีย" ก็ไม่ใช่ข้อเท็จจริงเกี่ยวกับเมืองสองเมือง มันคือค่าที่อ่านได้โดยตรงของความโค้งของโลกระหว่างสองเมืองนั้น ในรูปของมุม เขาวัดมุมที่อะเล็กซานเดรียได้ราว 7.2 องศา หรือหนึ่งในห้าสิบของวงกลมเต็มวง เหลือตัวเลขที่ต้องการอีกตัวเดียว — ระยะทางบนพื้นระหว่างสองเมือง — และเขาได้มาจาก bematist นักนับก้าวอาซีฟที่ถูกฝึกให้ก้าวเท่ากันทุกก้าว และรายงานระยะทางด้วยการนับก้าว พวกเขาให้ตัวเลขราว 5,000 สตาเดีย (APS News) คูณ 5,000 ด้วย 50 เพราะมุมที่วัดได้เป็นหนึ่งในห้าสิบของวงกลมเต็มวงที่เส้นรอบวงของโลกจะลากได้ โลกจึงมีเส้นรอบวง 250,000 สตาเดีย

E N

The honest version of this story has a hole in it, and the hole is worth naming rather than papering over. Eratosthenes' own treatise, *On the Measurement of the Earth*, is lost; everything known about his method comes down through a summary written some four hundred years later by the astronomer Cleomedes, who compressed and possibly simplified the original argument — the well may be a later illustrative flourish rather than Eratosthenes' own evidentiary step (Cleomedes, *On the Circular Motions*, discussed in the Eratosthenes literature). And the accuracy everyone likes to quote — "within 2 percent of the true value" — depends entirely on which ancient stadion you plug in for the unit, and no one today knows for certain which one Eratosthenes used. Values proposed for the stadion range from about 157 to 185 meters; on the low end the error comes out under 1 percent, on the high end it balloons past 15 percent (MAA, "Eratosthenes

and the Mystery of the Stades"). The number is famous. The precision attached to it is a modern retrofit.

TH

เรื่องนี้ในเวอร์ชันที่ชื่อตรงมีรูโหว่ และรูนั้นควรถูกเรียกชื่อ ไม่ใช่ถูกฉาบทับ ตำราของ Eratosthenes เอง On the Measurement of the Earth สูญหายไปแล้ว ทุกอย่างที่เราเกี่ยวกับวิธีของเขาตกทอดมาผ่านบทสรุปที่นักดาราศาสตร์ชื่อ Cleomedes เขียนขึ้นราวสี่ร้อยปีให้หลัง Cleomedes บีบย่อข้อโต้แย้งดั้งเดิม และอาจทำให้มันง่ายลงด้วย — บ่อน้ำอาจเป็นลูกเล่นเชิงภาพประกอบที่เติมเข้ามาทีหลัง ไม่ใช่ขั้นตอนหาหลักฐานของ Eratosthenes เอง (Cleomedes, On the Circular Motions, ที่อธิบายไว้ในวรรณกรรมว่าด้วย Eratosthenes) ความแม่นยำที่ใครๆ ชอบอ้าง — "คลาดจากค่าจริงไม่ถึง 2 เปอร์เซ็นต์" — ก็ขึ้นอยู่กับว่าแทนหน่วยstadia อื่นโบราณอันไหนลงไปทั้งหมด และวันนี้ไม่มีใครรู้แน่ชัดว่า Eratosthenes ใช้อันไหน ค่าที่มีผู้เสนอสำหรับstadia อยู่ระหว่างราว 157 ถึง 185 เมตร ที่ปลายด้านต่ำ ความคลาดเคลื่อนไม่ถึง 1 เปอร์เซ็นต์ ที่ปลายด้านสูง มันพองเกิน 15 เปอร์เซ็นต์ (MAA, "Eratosthenes and the Mystery of the Stades") ตัวเลขนั้นโด่งดัง ความแม่นยำที่ผูกติดมากับมันเป็นของที่ยุคสมัยใหม่ใส่กลับเข้าไปทีหลัง

EN

None of that touches the actual achievement, which was never really about hitting the right number. It was about realizing that the size of the entire planet could be read off two shadows and a walked distance, without a ship, a border crossing, or a single mile of travel beyond what the bematists had already covered for other reasons. Every other method available to the ancient world involved going somewhere. Eratosthenes' method involved standing in two places at once — one of them, notably, a place he never visited.

TH

ไม่มีข้อไหนในนั้นแต่ต้องความสำเร็จจริง เพราะความสำเร็จนั้นไม่เคยเป็นเรื่องของการยิงให้ตรงตัวเลข มันคือการเห็นว่าขนาดของดาวทั้งดวงอ่านออกมาได้จากเงาสองเงากับระยะทางที่มีคนเดินนับไว้ ไม่ต้องมีเรือ ไม่ต้องข้ามพรมแดน ไม่ต้องเดินทางเพิ่มแม้แต่ไมล์เดียวเกินกว่าที่ bematist เดินไปแล้วด้วยเหตุผลอื่น วิธีอื่นทุกวิธีที่โลกโบราณมี ล้วนต้องไปที่ไหนสักแห่ง วิธีของ Eratosthenes คือยืนอยู่สองที่พร้อมกัน — และที่หนึ่งในนั้น เขาไม่เคยไปเยือนเลย

EN

The move: the Earth was too big to walk; a shadow was small enough to measure.

TH

ท่า: โลกใหญ่เกินกว่าจะเดิน เงาเล็กพอที่จะวัด

The Sky Problem That Became a Clock Problem — London, 1714–1773

ปัญหาบนฟ้าที่กลายเป็นปัญหาของนาฬิกา — ลอนดอน 1714–1773

EN

By the early eighteenth century, a ship at sea could find its latitude in an afternoon, from the height of the sun. It had no reliable way to find its longitude — its position east or west — and the consequence was routine catastrophe. In October 1707 a Royal Navy squadron under Sir Cloudesley Shovell, badly wrong about its position, wrecked on the rocks of the Isles of Scilly and killed as many as 1,500 sailors. Parliament responded in 1714 with

the Longitude Act, which established a Board of Longitude and offered up to £20,000 — several million pounds today — to anyone who could determine longitude at sea to within half a degree (Royal Museums Greenwich, "Longitude found — the story of Harrison's timekeepers").

TH

ต้นศตวรรษที่สิบแปด เรือกลางทะเลหาละติจูดของตัวเองได้ภายในพายเดียวจาก ความสูงของดวงอาทิตย์ แต่หาลองจิจูด — ตำแหน่งตะวันออกหรือตะวันตก — ด้วย วิธีที่เชื่อถือได้ไม่ได้เลย ผลคือหายนะที่เกิดเป็นประจำ เดือนตุลาคม 1707 กองเรือ ของราชนาวีอังกฤษภายใต้ Sir Cloudesley Shovell เข้าใจตำแหน่งตัวเองผิดอย่าง หนัก จนโขดหินที่หมู่เกาะ Scilly และคร่าชีวิตลูกเรือมากถึง 1,500 คน รัฐสภาตอบ ด้วย Longitude Act ในปี 1714 กฎหมายนั้นตั้ง Board of Longitude ขึ้นมา และตั้ง รางวัลสูงถึง 20,000 ปอนด์ — หลายล้านปอนด์ในค่าเงินวันนี้ — ให้ใครก็ตามที่หา ลองจิจูดกลางทะเลได้แม่นยำในระยะครึ่งองศา (Royal Museums Greenwich, "Longitude found — the story of Harrison's timekeepers")

EN

For decades, the Board and most of the scientific establishment treated this as an astronomy problem, because longitude is, mathematically, a time-difference problem, and the sky was the only clock anyone trusted at sea. The first serious proposal predated the Act by a century: Galileo Galilei, having discovered Jupiter's four largest moons in 1610, proposed timing their frequent eclipses as a celestial clock visible from anywhere on Earth. It worked beautifully from a fixed observatory on land and failed completely at sea — a telescope precise enough to catch a Jovian eclipse could not be held steady on a rolling deck, and Jupiter itself disappeared behind the sun for roughly ten weeks of every year. The method that displaced it, refined over decades by the future Astronomer Royal Nevil Maskelyne into the "lunar distance method," used the moon's position against the fixed stars as a natural clock face, cross-referenced against printed tables that took teams of human computers years to calculate (RMG, "Longitude found: Nevil Maskelyne and the lunar method"; explorersweb.com, "How We Learned to Measure Longitude at Sea"). It worked, eventually, but it demanded a trained navigator, clear skies, and roughly four hours of finicky calculation per fix. A

century of the era's best astronomical minds had converged on the sky as the only place a solution could live.

TH

หลายทศวรรษ Board และแวดวงวิทยาศาสตร์ส่วนใหญ่ถือว่านี่คือปัญหาดาราศาสตร์ เหตุผลไม่เียง: ในทางคณิตศาสตร์ ลองจินตนาการถึงปัญหาผลต่างของเวลา และท้องฟ้าคือนาฬิกาเรือนเดียวที่คนกลางทะเลไว้ใจ ข้อเสนอจริงจึงขึ้นแรกมาก่อน กฎหมายฉบับนั้นหนึ่งศตวรรษ Galileo Galilei ค้นพบดวงจันทร์ใหญ่สี่ดวงของดาวพฤหัสบดีในปี 1610 แล้วเสนอให้จับเวลาอุปราคาที่เกิดถี่ของมัน เป็นนาฬิกาบนฟ้าที่มองเห็นได้จากทุกที่บนโลก มันทำงานได้ดีจากหอดูดาวที่ตั้งอยู่กับที่บนบก และล้มเหลวสิ้นเชิงกลางทะเล — กล้องโทรทรรศน์ที่แม่นยำจะจับอุปราคาของดาวพฤหัสบดี ประคองให้นั่งบนดาวฟ้าที่โคลงไม่ไหว และตัวดาวพฤหัสบดีเองก็หายไปหลังดวงอาทิตย์ราวสิบสัปดาห์ทุกปี วิธีที่มาแทนที่คือ "lunar distance method" Nevil Maskelyne ผู้ต่อมาได้เป็น Astronomer Royal ชัดเกลามันอยู่หลายทศวรรษ วิธีนี้ใช้ตำแหน่งของดวงจันทร์เทียบกับดาวฤกษ์เป็นหน้าปัดนาฬิกาตามธรรมชาติ แล้วเทียบกลับกับตารางที่พิมพ์ไว้ ตารางที่พิมพ์นี้ผู้ทำหน้าที่คำนวณต้องใช้เวลาหลายปีกว่าจะคิดออกมาได้ (RMG, "Longitude found: Nevil Maskelyne and the lunar method"; explorersweb.com, "How We Learned to Measure Longitude at Sea") มันใช้ได้ในที่สุด แต่มันเรียกร้องนักเดินเรือที่ผ่านการฝึก ท้องฟ้าที่ปลอดภัย และการคำนวณจุกจิกราวสี่ชั่วโมงต่อการหาตำแหน่งหนึ่งครั้ง หนึ่งศตวรรษของสมองดาราศาสตร์ที่ดีที่สุดในยุคนั้นบรรจบกันที่ท้องฟ้า ในฐานะที่เดียวที่คำตอบจะอยู่ได้

EN

John Harrison, a joiner and self-taught clockmaker from Lincolnshire, refused the sky. His reframing was almost embarrassingly literal: if longitude is a time-difference problem, then the instrument that solves it is not a telescope. It is a clock — one accurate enough to carry Greenwich time, unaltered, through months at sea, through temperature swings, through humidity, through the pitch and roll of a wooden hull. That had always sounded absurd; contemporary clocks lost or gained minutes a day on dry land. Harrison spent over three decades building larger and then, crucially, smaller mechanisms — H1, H2, H3 — before arriving in 1759 at H4, an instrument

the size of a large pocket watch, weighing about 1.5 kilograms, that kept time at sea to within a few seconds a day (RMG; Worn & Wound, "The Harrison H4").

T H

John Harrison ช่างไม้และช่างนาฬิกาที่เรียนรู้ด้วยตัวเองจาก Lincolnshire ปฏิเสธ ท้องฟ้า การเปลี่ยนกรอบ (reframe) ของเขาตรงตัวจนแทบน่าอาย: ถ้าลองจับจุดคือ ปัญหาผลต่างของเวลา เครื่องมือที่แก้มันก็ไม่ใช่อะไรที่ทรงเกียรติ แต่เป็นนาฬิกา — นาฬิกาที่แม่นยำจะแบกเวลากลับคืนมาได้โดยไม่เพี้ยน ผ่านหลายเดือนกลางทะเล ผ่าน อุณหภูมิที่แกว่ง ผ่านความชื้น ผ่านการกระดกและโคลงของตัวเรือไม้ เรื่องนี้ฟังดู เหลวไหลมาตลอด นาฬิกาในยุคนั้นเดินเร็วหรือช้าไปเป็นนาทีต่อวันแม้อยู่บนบก แท้ๆ Harrison ใช้เวลากว่าสามทศวรรษสร้างกลไกที่ใหญ่ขึ้น แล้ว — ตรงนี้คือหัวใจ — เล็กลง H1, H2, H3 ก่อนจะมาถึง H4 ในปี 1759 เครื่องขนาดเท่านาฬิกาพกเรือน ใหญ่ หนักราว 1.5 กิโลกรัม รักษาเวลากลางทะเลได้คลาดไม่กี่วินาทีต่อวัน (RMG; Worn & Wound, "The Harrison H4")

E N

On a 1761–62 trial voyage to Jamaica, H4 gave a longitude accurate to about ten miles. That was well inside the prize's terms. It took Harrison until 1773 — helped, near the end, by the personal intervention of King George III — to be paid the full reward, because the Board, dominated by astronomers with careers invested in the lunar-distance method, kept moving the terms of judgment (RMG). The lunar method also won its own consolation payments; both approaches, in the end, went to sea. But it is the clockmaker's reframe, not the astronomers' refinement, that survives in every ship and aircraft today as the chronometer.

T H

H4 ให้ลองจับจุดที่คลาดเคลื่อนราวสิบไมล์ ในการเดินเรือทดสอบไป Jamaica ปี 1761–62 นั้นอยู่ในเกณฑ์ของรางวัลอย่างสบาย แต่ Harrison ได้รับเงินรางวัลเต็ม จำนวนในปี 1773 — ช่วงท้ายได้พระเจ้า George ที่ 3 เข้ามาแทรกแซงด้วยพระองค์ เอง — เพราะ Board เลื่อนเงื่อนไขการตัดสินไปเรื่อยๆ Board นั้นมีนักดาราศาสตร์ ครองอยู่ และนักดาราศาสตร์เหล่านั้นลงทุนอาชีพไว้กับ lunar distance method (RMG) วิธีดวงจันทร์เองก็ได้เงินปลอบใจของมัน ทั้งสองแนวทางลงเรือกันทั้งคู่ในตอน

จบ แต่สิ่งที่รอดมาถึงเรือและอากาศยานทุกลำในวันนี้ในชื่อนาฬิกาเดินเรือ (marine chronometer) คือการเปลี่ยนกรอบของช่างนาฬิกา ไม่ใช่ว่าการขัดเกลาของนักดาราศาสตร์

E N

The move: everyone was improving the sky; the answer was in a pocket.

T H

ท่า: ทุกคนกำลังขัดเกลาท้องฟ้า คำตอบอยู่ในกระเป๋าเสื้อ

A Pump Handle in Soho — London, 1854

ตามสูบน้ำใน Soho — ลอนดอน 1854

E N

In late August 1854, cholera broke out in the streets around Broad Street in London's Soho district with a speed that shocked even a city inured to epidemics: over 600 deaths in the surrounding blocks within about ten days (London Museum, "John Snow: Cholera & the Broad Street pump"). The dominant medical theory of the day, championed by the statistician William Farr, held that cholera traveled through "miasma" — bad air rising from low, damp, organically rich ground; Farr had data from 1852 seemingly showing cholera deaths falling as elevation rose, and took that as confirmation (Farr, 1852 elevation report, discussed at brianaltonenmph.com).

T H

ปลายเดือนสิงหาคม 1854 อหิวาตกโรคระบาดขึ้นตามถนนรอบ Broad Street ในย่าน Soho ของลอนดอน เร็วจนเมืองที่ชินกับโรคระบาดยังตกใจ: กว่า 600 ศพในบล็อกโดยรอบภายในราวสัปดาห์ (London Museum, "John Snow: Cholera & the Broad Street pump") ทฤษฎีทางการแพทย์ที่ครองยุคนั้นถือว่าอหิวาต์เดินทางมากับ "อากาศเสีย" ตามทฤษฎีอากาศเสีย (miasma) — อากาศที่ลอยขึ้นจากพื้นดินต่ำชื้น และอุดมด้วยสารอินทรีย์ นักสถิติ William Farr เป็นหัวหอกของทฤษฎีนี้ Farr มีข้อมูลจากปี 1852 ที่ดูเหมือนแสดงว่าการตายด้วยอหิวาต์ลดลงเมื่อระดับความสูงเพิ่มขึ้น และเขาถือว่านั่นคือการยืนยัน (Farr, รายงานระดับความสูงปี 1852, อภิปรายที่ brianaltonenmph.com)

EN

John Snow, a physician who had already argued in print that cholera was waterborne, made a decision that looks obvious only in hindsight: he stopped treating "the neighborhood" as the unit of analysis and started treating each *address* as a data point. He went door to door recording where the dead had lived and, crucially, where they drew their water. Plotted on a street map, the deaths clustered tightly around a single public water pump on Broad Street (UCLA Department of Epidemiology, John Snow Archive). On September 7, Snow persuaded the local council to remove the pump's handle.

TH

John Snow เลิกใช้ "ย่าน" เป็นหน่วยวิเคราะห์ แล้วหันมาใช้แต่ละบ้านเลขที่เป็นจุดข้อมูล การตัดสินใจนี้ดูชัดเจนก็ต่อเมื่อมองย้อนกลับไป Snow เป็นแพทย์ที่เคยเขียนโต้แย้งตีพิมพ์ไว้แล้วว่าอหิวาต์มากับน้ำ เขาเดินเคาะประตูทีละหลัง บันทึกว่าผู้ตายเคยอาศัยอยู่ที่ไหน และที่สำคัญกว่านั้น ดักน้ำจากที่ไหน เมื่อลงจุดบนแผนที่ถนน การตายเกาะกลุ่มแน่นรอบเครื่องสูบน้ำสาธารณะเครื่องเดียวบน Broad Street (UCLA Department of Epidemiology, John Snow Archive) วันที่ 7 กันยายน Snow โน้มน้าวสภาท้องถิ่นให้ถอดด้ามของเครื่องสูบน้ำนั้นออก

EN

The map itself is more famous than its timing deserves: the polished version most textbooks reproduce, with its stacked bars of death marching down Broad Street toward the pump, was not the tool Snow used to find the pattern during the outbreak — he traced the cluster first through door-to-door interviews — but a version he prepared afterward and presented to the London Epidemiological Society that December, as an argument for a theory he had already reached by legwork (Esri, "Something in the water").

TH

ตัวแผนที่ที่มีชื่อเสียงเกินกว่าที่ลำดับเวลาของมันสมควรได้ เวอร์ชันขัดเงาที่ตำราส่วนใหญ่พิมพ์ซ้ำ พร้อมแท่งการตายเรียงซ้อนทอดลง Broad Street ไปหาเครื่องสูบน้ำ ไม่ใช่เครื่องมือที่ Snow ใช้หารูปแบบระหว่างการระบาด — เขาไล่ตามกลุ่มก้อนนั้นด้วยการสัมภาษณ์ตามบ้านก่อน — แต่เป็นเวอร์ชันที่เขาเตรียมขึ้นภายหลัง แล้วนำเสนอต่อ London Epidemiological Society ในเดือนธันวาคมปีนั้น เพื่อเป็นข้อโต้แย้งสนับสนุนทฤษฎีที่เขาไปถึงแล้วด้วยการเดินเท้า (Esri, "Something in the water")

EN

Here the popular version of the story gets ahead of the evidence, and Snow's own account is more careful than his admirers. By the time the handle came off, the outbreak was already sharply declining — most of the deaths had already occurred, and case counts had been falling for days. Snow himself later wrote that he could not say the removal had much effect on an epidemic already burning out (Esri, "Something in the water: the mythology of Snow's map of cholera"). The pump handle is a legend of clean cause and effect; the truth is messier and, for a book about method, more instructive: the map's real power was not stopping this outbreak but making the *mechanism* visible for the first time — later investigation found the well beneath the pump had been contaminated by a leaking cesspit only feet away, including waste from an infant who had contracted cholera days earlier.

TH

ตรนี้เรื่องเล่าเวอร์ชันยอดนิยมนิ่งแข็งหน้าหลักฐาน และคำบอกเล่าของ Snow เอง ระมัดระวังกว่าของบรรดาผู้ชื่นชมเขา ตอนที่ด้ามสูบถูกถอดออก การระบาดกำลังลดลงอย่างชัดเจนอยู่แล้ว — คนส่วนใหญ่ตายไปก่อนหน้านี้แล้ว และจำนวนผู้ป่วยลดลงมา

หลายวันแล้ว Snow เขียนเองในภายหลังว่าเขาไม่อาจบอกได้ว่าการถอดด้ามมีผลมากนักต่อโรคระบาดที่กำลังมอดลงอยู่แล้ว (Esri, "Something in the water: the mythology of Snow's map of cholera") ด้ามสูบน้ำเป็นตำนานของเหตุและผลที่สะอาดหมดจด ความจริงยิ่งกว่านั้น และสำหรับหนังสือที่ว่าด้วยวิธี ความจริงสอนได้มากกว่า: พลัจริยของแผนที่ไม่ใช่การหยุดการระบาดครั้งนี้ แต่คือการทำให้กลไกมองเห็นได้เป็นครั้งแรก การสอบสวนภายหลังพบว่าบ่อน้ำใต้เครื่องสูบน้ำเปื้อนจากบ่อเกรอะที่รั่ว ห่างออกไปเพียงไม่กี่ฟุต รวมถึงของเสียจากทารกที่ติดอหิวาต์ก่อนหน้านี้วันไม่กี่วัน

EN

Snow's strongest evidence, in fact, came from a second and quieter piece of work often called his "grand experiment." Two water companies, Lambeth and Southwark & Vauxhall, supplied overlapping streets of south London — customers on the same block, often the same building, could be served by either. Until 1852 both companies drew water from a badly polluted stretch of the Thames; Lambeth then moved its intake upstream, above the worst sewage discharge, while Southwark & Vauxhall did not. In the 1854 epidemic, houses supplied by Southwark & Vauxhall died of cholera at eight to nine times the rate of houses supplied by Lambeth — a comparison Snow could make because everything else about the two populations was, for practical purposes, identical (UCLA John Snow Archive). It took over a decade for Farr, and the medical establishment generally, to concede the point; Farr himself finally credited water as the primary vector only in 1866, eight years after Snow's death (Wiley Significance, "London cholera and the blind-spot of an epidemiology theory").

TH

หลักฐานที่แข็งแกร่งที่สุดของ Snow มาจากงานชิ้นที่สองที่เรียกว่า งานที่มักถูกเรียกว่า "การทดลองใหญ่" ของเขา บริษัทน้ำสองแห่ง Lambeth กับ Southwark & Vauxhall ส่งน้ำให้ถนนที่ทับซ้อนกันในลอนดอนใต้ — ลูกค้านับบล็อกเดียวกัน บ่อยครั้งในตึกเดียวกัน อาจใช้น้ำจากบริษัทไหนก็ได้ จนถึงปี 1852 ทั้งสองบริษัทสูบน้ำจากช่วงแม่น้ำเทมส์ที่สกปรกอย่างหนัก จากนั้น Lambeth ย้ายจุดรับน้ำขึ้นไปเหนือน้ำ ฟันจุดที่ปล่อยสิ่งปฏิกูลหนักที่สุด Southwark & Vauxhall ไม่ย้าย ในการระบาดปี 1854 บ้านที่ใช้น้ำของ Southwark & Vauxhall ตายด้วยอหิวาต์ในอัตราแปดถึงเก้าเท่าของ

บ้านที่ใช้น้ำของ Lambeth Snow เปรียบเทียบแบบนี้ได้เพราะทุกอย่างที่เหลือของประชากรสองกลุ่ม ในทางปฏิบัติแล้วเหมือนกันหมด (UCLA John Snow Archive) Farr และแวดวงการแพทย์โดยรวมใช้เวลาว่าสิบปีกว่าจะยอมรับ Farr เองยกให้น้ำเป็นพาหะหลักในปี 1866 แปดปีหลัง Snow เสียชีวิต (Wiley Significance, "London cholera and the blind-spot of an epidemiology theory")

E N

The move: the city was hiding a mechanism behind its geography; an address book found it.

T H

ท่า: เมืองซ่อนกลไกไว้หลังภูมิศาสตร์ของมัน สมุดบ้านเลขที่หามันเจอ

Two Doors, One Corridor — Vienna, 1847

สองประตู หนึ่งทางเดิน — เวียนนา 1847

E N

The Vienna General Hospital's maternity service ran two adjacent clinics that admitted patients on alternating days, which made them, unintentionally, one of history's cleanest natural experiments. Clinic 1 was staffed by physicians and medical students, who spent their mornings performing autopsies before walking to the maternity ward. Clinic 2 was staffed entirely by midwives, who never touched a cadaver. For years, women begged to be admitted to Clinic 2, sometimes preferring to give birth in the street rather than risk Clinic 1 — because everyone on the ward already knew, without knowing why, that Clinic 1 killed. Mortality from puerperal ("childbed") fever there ran as high as 18 percent of mothers in some months; Clinic 2's rate was a fraction of that (Science History Institute, "Ignaz Semmelweis").

T H

แผนกสูติกรรมของโรงพยาบาลกลางเวียนนาเปิดคลินิกสองแห่งติดกัน รับคนไข้สลับวันกัน และนั่นทำให้ทั้งสองกลายเป็นการทดลองตามธรรมชาติที่สะอาดที่สุดครั้งหนึ่งในประวัติศาสตร์ โดยไม่มีใครตั้งใจ คลินิก 1 มีแพทย์และนักศึกษาแพทย์ประจำ คนกลุ่มนี้ใช้เวลาช่วงเช้าผ่าศพก่อนเดินมาที่หอผู้ป่วยคลอด คลินิก 2 มีแต่ผดุงครรภ์ทั้งหมด คนที่ไม่เคยแตะศพเลย หลายปีที่ผู้หญิงอ่อนวอนขอเข้าคลินิก 2 บางคนยอม

คลอดกลางถนนดีกว่าเสี่ยงกับคลินิก 1 — เพราะทุกคนในหอผู้ป่วยรู้อยู่แล้วว่าคลินิก 1 ฆ่าคน แม้ไม่รู้ว่าจะเพราะอะไร อัตราตายจากไข้หลังคลอด (puerperal fever) ที่นั่น สูงถึง 18 เปอร์เซ็นต์ของมารดาในบางเดือน อัตราของคลินิก 2 เป็นเพียงเศษเสี้ยวของตัวเลขนั่น (Science History Institute, "Ignaz Semmelweis")

E N

Ignaz Semmelweis, a Hungarian obstetrician working in Clinic 1, tried the theories on offer — overcrowding, diet, even the shame of male examiners — and each collapsed against some counterexample. The break came in 1847, when Semmelweis's friend and colleague Jakob Kolletschka died after cutting his finger during an autopsy and developed a wasting illness whose autopsy findings looked identical to the ones killing mothers on the ward. Semmelweis's insight was to treat that as data, not tragedy: the doctors themselves, hands fresh from dissecting corpses, were carrying "cadaverous particles" from the morgue to the delivery bed. The two clinics were not different in staff skill or patient population. They differed in exactly one variable — what the attendants' hands had touched before they touched the mother — and Semmelweis had, without designing it, an experiment that isolated that variable perfectly.

T H

Ignaz Semmelweis สูติแพทย์ชาวฮังการีในคลินิก 1 ไล่ลองทฤษฎีที่มีอยู่ — ความแออัด อาหาร กระทั่งความอับอายเมื่อถูกผู้ชายตรวจ — และแต่ละข้อพังลงเมื่อเจอตัวอย่างค้าน จุดหักเหมาถึงในปี 1847 Jakob Kolletschka เพื่อนและเพื่อนร่วมงานของ Semmelweis มีบาดนี้ระหว่างผ่าศพ ป่วยด้วยอาการทรุดโทรม แล้วเสียชีวิต ผลชันสูตรของเขายกมาเหมือนกับที่กำลังคร่าชีวิตมารดาในหอผู้ป่วยทุกประการ การหยั่งเห็น (insight) ของ Semmelweis คือรับมันไว้เป็นข้อมูล ไม่ใช่โศกนาฏกรรม: หมอเองนั่นแหละ มือเพิ่งพ้นจากการชำแหละศพ กำลังหอบ "อนุภาคจากศพ" จากห้องเก็บศพมาถึงเตียงคลอด คลินิกสองแห่งไม่ต่างกันที่ฝีมือของบุคลากรหรือกลุ่มคนไข้ มันต่างกันที่ตัวแปรเดียว — สิ่งที่มีมือของผู้ทำคลอดสัมผัสมาก่อนจะสัมผัสมารดา — และ Semmelweis มีการทดลองที่แยกตัวแปรนั้นออกมาได้สมบูรณ์อยู่ในมือ โดยไม่ต้องออกแบบมันเอง

EN

He ordered doctors and students to scrub with a chlorinated lime solution before every examination, strong enough to remove the smell of decay that soap could not. The mortality rate in Clinic 1 fell from 18.3 percent in April 1847 to 2.2 percent by June; over 1848, the first full year of mandatory chlorine washings, the clinic's rate stood at 1.27 percent, converging with Clinic 2's (Science History Institute). By any modern standard this was a clean, replicated, dramatic result: one intervention, one variable, a fivefold to eightfold drop in deaths.

TH

เขาสั่งให้แพทย์และนักศึกษาขัดล้างมือด้วยสารละลายปูนคลอรีนก่อนตรวจทุกครั้ง เข้มพอจะขจัดกลิ่นเน่าที่สบู่ออกไม่ได้ อัตรามรณะในคลินิก 1 ลดจาก 18.3 เปอร์เซ็นต์ในเดือนเมษายน 1847 เหลือ 2.2 เปอร์เซ็นต์ในเดือนมิถุนายน ตลอดปี 1848 ปีเต็มปีแรกที่บังคับล้างมือด้วยคลอรีน อัตราของคลินิกอยู่ที่ 1.27 เปอร์เซ็นต์ เข้าใกล้ตัวเลขของคลินิก 2 (Science History Institute) วัดด้วยมาตรฐานสมัยใหม่ ข้อไหนก็ตาม นี่คือผลลัพธ์ที่สะอาด ทำซ้ำได้ และชัด: หนึ่งการแทรกแซง หนึ่งตัวแปร การตายลดลงห้าถึงแปดเท่า

EN

The story does not end there, and the second half is the part every account of "solving" needs and most omit. Semmelweis never published his reasoning in a form the medical establishment found persuasive; he had no germ theory to explain *why* chlorinated lime worked, only the correlation, and he defended it with an abrasiveness that alienated the senior figures who might have carried the idea forward. Vienna's obstetric establishment largely rejected the practice after he left the hospital in 1849; childbed fever deaths at his former clinic rose again. Semmelweis grew increasingly bitter and erratic over the following years, was committed to an asylum in 1865, and died there weeks later, reportedly from an infected wound — apparently the very kind of infection his hand-washing rule was designed to prevent (Wikipedia, "Ignaz Semmelweis"; Linda Hall Library, "Scientist of the Day: Ignaz Semmelweis"). His result was vindicated within two decades, once Pasteur and Lister supplied the germ theory that explained it. The lesson this book states plainly, because Semmelweis paid for it with his career

and arguably his life: solving a problem is not complete when the answer is correct. It is complete when the answer is *transmitted* — accepted, adopted, and used by people who were not there for the discovery. A right answer that dies with its discoverer has not finished being solved.

TH

เรื่องไม่จบตรงนั้น ครึ่งหลังคือส่วนที่คำบอกเล่าเรื่อง "การแก้ปัญห" ทุกชิ้นต้องการ และส่วนใหญ่ตัดทิ้ง Semmelweis ไม่เคยตีพิมพ์เหตุผลของเขาในรูปแบบที่แวดวง การแพทย์เห็นว่ามันน่าเชื่อถือ เขาไม่มีทฤษฎีเชื้อโรคไว้อธิบายว่าทำไมปูนคลอรีนถึงได้ผล มีแต่ความสัมพันธ์ที่สังเกตได้ และเขาปกป้องมันด้วยท่าทีบาดคน จนผลักผู้อาวุโสที่ อาจสานต่อความคิดนี้ให้ห่างออกไป หลังเขาออกจากโรงพยาบาลในปี 1849 แวดวง สุติศาสตร์ของเวียนนาปฏิเสธแนวปฏิบัตินี้เป็นส่วนใหญ่ การตายจากไข้หลังคลอดที่ คลินิกเดิมของเขาสูงขึ้นอีก ปีต่อๆ มา Semmelweis ขมขื่นและควบคุมตัวเองได้น้อย ลงเรื่อยๆ ถูกส่งเข้าสถานบำบัดจิตในปี 1865 และเสียชีวิตที่นั่นในอีกไม่กี่สัปดาห์ต่อ มา มีรายงานว่าจากผลติดเชื้อ — ดูเหมือนจะเป็นการติดเชื้อชนิดเดียวกับที่กฎล้าง มือของเขาถูกออกแบบมาเพื่อป้องกันพอดี (Wikipedia, "Ignaz Semmelweis"; Linda Hall Library, "Scientist of the Day: Ignaz Semmelweis") ผลของเขาได้รับการ พิสูจน์ว่าถูกภายในสองทศวรรษ เมื่อ Pasteur และ Lister ส่งทฤษฎีเชื้อโรคที่ อธิบายมันเข้ามา บทเรียนที่หนังสือเล่มนี้พูดตรงๆ เพราะ Semmelweis จำค่ามัน ด้วยอาชีพและอาจรวมถึงชีวิต: การแก้ปัญหายังไม่จบเมื่อคำตอบถูกต้อง มันจบเมื่อ คำตอบถูกส่งต่อ — ได้รับการยอมรับ ถูกนำไปใช้ และถูกใช้จริงโดยคนที่ไม่ได้อยู่ใน วันที่ค้นพบ คำตอบที่ถูกแต่ตายไปพร้อมผู้ค้นพบ ยังแก้ไม่เสร็จ

EN

The move: the variable was the hands, not the ward; and finding it was only half the job.

TH

ท่า: ตัวแปรคือมือ ไม่ใช่หอผู้ป่วย และการหามันเจอเป็นเพียงครึ่งเดียวของงาน

The Six-Foot Box That Replaced the Sky — Kitty Hawk, 1901–1902

กล่องยาวหกฟุตที่มาแทนท้องฟ้า — Kitty Hawk 1901–1902

E N

By the end of the 1901 flying season, Wilbur and Orville Wright were, by their own later account, close to giving up. Their new glider, built using the most trusted aerodynamic data available — tables of lift published by the German gliding pioneer Otto Lilienthal, refined through hundreds of his own flights — should have flown well. It generated only about a third of the lift the tables predicted (NASA Glenn Research Center, "Wright 1901 Wind Tunnel").

T H

สิ้นฤดูบินปี 1901 Wilbur กับ Orville Wright เกือบยอมแพ้ ตามที่ทั้งคู่เล่าเองในภายหลัง เครื่องร่อนลำใหม่ของพวกเขาคงจะบินได้ดี มันสร้างขึ้นจากข้อมูลอากาศพลศาสตร์ที่น่าเชื่อถือที่สุดเท่าที่มี — ตารางแรงยกที่ Otto Lilienthal ผู้บุกเบิกการร่อนชาวเยอรมันตีพิมพ์ไว้ ชัดเกลามาจากการบินของตัวเองหลายร้อยเที่ยว — แต่มันสร้างแรงยกได้เพียงราวหนึ่งในสามของที่ตารางทำนายไว้ (NASA Glenn Research Center, "Wright 1901 Wind Tunnel")

E N

The instinct of most engineers in that position is to keep flying, adjusting, and hoping the next glider closes the gap — to keep searching inside the expensive, dangerous, weather-dependent world of an actual flight test at Kitty Hawk, North Carolina, four hundred miles from their bicycle shop in Dayton, Ohio. The Wrights instead concluded that the data itself, not their execution, was the problem, and that no amount of further gliding at Kitty Hawk would tell them why. They went home and built a wind tunnel: a wooden box roughly six feet long, with a fan at one end driven by a belt off the gas engine that powered their shop tools, and — the genuinely novel part — two homemade balances that could measure the lift and drag of a small model wing directly, converting airflow into a needle deflection they could read through a viewing window (Wright-House, "Doing it Wright").

T H

พี่น้อง Wright สรุปว่าตัวข้อมูลเองคือปัญหา ไม่ใช่ฝีมือการทำของพวกเขา และการร่อนที่ Kitty Hawk อีกก็ครั้งก็ไม่มีทางบอกได้ว่าเพราะอะไร สัญชาตญาณของวิศวกรส่วนใหญ่ในสถานการณ์นั้นคือบินต่อ ปรับต่อ และหวังว่าเครื่องร่อนลำหน้าจะปิดช่องว่างได้ — ค้นหาต่อไปในโลกที่แพง อันตราย และขึ้นอยู่กับดินฟ้าอากาศ ของการทดสอบบินจริงที่ Kitty Hawk รัฐ North Carolina ห่างจากร้านจักรยานของพวกเขาที่ Dayton รัฐ Ohio สี่ร้อยไมล์ พวกเขากลับบ้านแล้วสร้างอุโมงค์ลม: กล่องไม้ยาวราวหกฟุต ปลายด้านหนึ่งมีพัดลมที่ขับเคลื่อนด้วยสายพานต่อจากเครื่องยนต์เบนซินที่ใช้ขับเคลื่อนมือในร้าน และ — ส่วนที่ใหม่จริง — ตราซังทำมือสองตัวที่วัดแรงยกและแรงต้านของปีกจำลองขนาดเล็กได้โดยตรง ตราซังแปลงกระแสลมให้เป็นการเบนของเข็ม อ่านได้ผ่านช่องกระจก (Wright-House, "Doing it Wright")

E N

Over the fall of 1901, working from strips of hammered and soldered sheet steel, they built and tested between one and two hundred miniature wing shapes, changing camber, aspect ratio, and section one variable at a time, then narrowed to roughly thirty designs for detailed, repeated measurement (NASA Glenn Research Center, "Wright 1901 Wind Tunnel"). This was the actual reframe, and it is easy to understate how large it was: the problem stopped being *how do we fly* and became *how do we measure lift, cheaply, indoors, a hundred times a day, without risking a neck*. An afternoon in the wind tunnel replaced a season at Kitty Hawk. Along the way they discovered that "Smeaton's coefficient," a constant used throughout aeronautics for over a century to convert air pressure into aerodynamic force, was itself wrong: the accepted value of 0.005 should have been closer to 0.0033, an error of roughly 40 percent sitting underneath every lift calculation the field had ever made, Lilienthal's included (NASA Glenn Research Center, "1901 Wind Tunnel Tests"). Modern aerodynamicists have since confirmed the Wrights' revised figure as substantially correct.

T H

ตลอดฤดูใบไม้ร่วงปี 1901 พวกเขาสร้างและทดสอบรูปทรงปีกจำลองระหว่างหนึ่งร้อยถึงสองร้อยแบบ จากแผ่นเหล็กที่ตีและบัดกรีเป็นแถบ เปลี่ยนความโค้ง อัตราส่วนกาง และหน้าตัด ทีละตัวแปร แล้วคัดเลือกราวสามสิบแบบไว้วัดซ้ำอย่าง

ละเอียด (NASA Glenn Research Center, "Wright 1901 Wind Tunnel") นี่คือการเปลี่ยนกรอบตัวจริง และง่ายมากที่จะพูดถึงมันน้อยกว่าที่มันใหญ่: ปัญหาเล็กเป็นเราจะบินได้อย่างไร แล้วกลายเป็นเราจะวัดแรงยกอย่างไรให้ถูก ในร่ม วันละร้อยครั้ง โดยไม่ต้องเอาออกไปเสี่ยง บ่ายเดียวในอุโมงค์ลมแทนที่ทั้งฤดูกาลที่ Kitty Hawk ระหว่างทางพวกเขาพบว่า "ค่าคงที่ Smeaton" ผิดเสียเอง ค่าคงที่นี้วงการการบินใช้กันมากกว่าหนึ่งศตวรรษเพื่อแปลงความดันอากาศเป็นแรงทางอากาศพลศาสตร์ ค่าที่ยอมรับกันคือ 0.005 ทั้งที่ควรใกล้ 0.0033 มากกว่า ความคลาดเคลื่อนราว 40 เปอร์เซ็นต์นั้นนอนอยู่ใต้การคำนวณแรงยกทุกครั้งที่วงการนี้เคยทำมา รวมถึงของ Lilienthal ด้วย (NASA Glenn Research Center, "1901 Wind Tunnel Tests") นักอากาศพลศาสตร์สมัยใหม่ยืนยันในภายหลังว่าตัวเลขที่พี่น้อง Wright แก้ใหม่นั้นถูกต้องเป็นส่วนใหญ่

EN

The 1902 glider, designed from wind-tunnel data instead of Lilienthal's tables, flew almost exactly as their calculations predicted. It is worth noting what the Wrights treated as solved first: not powered flight, but *control* — the 1902 glider's three-axis system of wing-warping, forward elevator, and rear rudder gave a pilot command over roll, pitch, and yaw before an engine was ever added, on the reasoning that an uncontrollable aircraft with a powerful motor is simply a fast way to crash (Smithsonian National Air and Space Museum, "Researching the Wright Way"). Power came in 1903, once control and lift were already answered questions. The famous first flight at Kitty Hawk was, by the time it happened, almost an anticlimax — the hard problem had already been solved, in a six-foot box, more than a year earlier.

TH

เครื่องร่อนปี 1902 ออกแบบจากข้อมูลอุโมงค์ลมแทนตารางของ Lilienthal และบินได้เกือบตรงตามที่มีการคำนวณของพวกเขาทำนายไว้ ควรสังเกตด้วยว่าพี่น้อง Wright ถือว่าอะไรถูกแก้ก่อน: ไม่ใช่การบินด้วยเครื่องยนต์ แต่คือการควบคุม — ระบบสามแกนของเครื่องร่อนปี 1902 ทั้งการบิดปีก แพนหางระดับด้านหน้า และหางเสือด้านหลัง ให้นักบินคุมการกลิ้ง การก้มเงย และการส่ายได้ก่อนจะมีการติดเครื่องยนต์เสียอีก เหตุผลของพวกเขา: อากาศยานที่ควบคุมไม่ได้แต่มีมอเตอร์แรงๆ ก็แค่วิธีตกที่เร็วขึ้น (Smithsonian National Air and Space Museum, "Researching the Wright

Way") กำลังขับมาถึงในปี 1903 เมื่อการควบคุมและแรงยกเป็นคำถามที่ตอบไปแล้ว การบินครั้งแรกอันโด่งดังที่ Kitty Hawk ในวันที่มันเกิดขึ้นจริง แทบเป็นฉากที่แผ่วลง — ปัญหาถูกแก้ไปแล้วในกล่องยาวหกฟุต ตั้งแต่กว่าหนึ่งปีก่อนหน้านั้น

EN

The move: the sky was the wrong instrument; a box on a workbench was the right one.

TH

ท่า: ท้องฟ้าคือเครื่องมือที่ผิด กล่องบนโต๊ะช่างคือเครื่องมือที่ถูก

The Move

ทำประจำบท

EN

Move 07 — Shrink the experiment: find the cheap measurement that stands in for the expensive world.

TH

Move 07 — ย่อการทดลอง — หาการวัดราคาถูกที่ยืนแทนโลกราคาแพงได้

EN

Every solver in this gallery faced a version of the same trap: the real problem — the size of the planet, a ship's position, a city's water, a ward's mortality, an aircraft's lift — was too large, too slow, too dangerous, or too far away to test directly. Each found a small, fast, repeatable stand-in that captured the part of the world that actually mattered and threw away the rest.

TH

นักแก้ปัญหาทุกคนในตอนนี้เจอกับดักเดียวกันในเวอร์ชันของตัวเอง ปัญหาจริง — ขนาดของดาวเคราะห์ ตำแหน่งของเรือ น้ำของเมือง อัตราตายของหอผู้ป่วย แรงยกของอากาศยาน — ใหญ่เกินไป ซ้ำเกินไป อันตรายเกินไป หรือไกลเกินกว่าจะทดสอบจริงๆ แต่ละคนหาตัวแทนที่เล็ก เร็ว และทำซ้ำได้ ตัวแทนที่จับส่วนของโลกที่สำคัญจริงๆ ไว้ แล้วทิ้งที่เหลือ

EN

- **Ask what the expensive world and the cheap surrogate share.** A shadow shares an angle with the Earth's curvature. A wind-tunnel model shares a coefficient of lift with a full-scale glider. The surrogate is only useful if that shared quantity is the one you actually need.
- **Isolate one variable, even when you cannot design the experiment yourself.** Semmelweis did not build his two clinics; he recognized that an accident of hospital scheduling had already isolated the variable for him. Snow's two water companies were an accident of corporate history. Half the skill in this move is noticing a natural experiment already running.
- **Distrust inherited data before distrusting your own execution.** The Wrights spent a season assuming their gliders were built wrong before they considered that Lilienthal's tables — the entire field's shared foundation — might be wrong instead. Reframing often starts with refusing to blame yourself for someone else's bad number.
- **A correct answer is not a finished one.** Semmelweis proved his case with numbers as clean as any in this book and still lost, because proof and persuasion are different problems with different solutions. Carrying an answer into other people's hands is part of the work, not an epilogue to it.

TH

- **ถามว่าโลกราคาแพงกับตัวแทนราคาถูกใช้อะไรร่วมกัน** เงานำใช้มุมร่วมกับความโค้งของโลก แบบจำลองในอุโมงค์ลมใช้สัมประสิทธิ์แรงยกร่วมกับเครื่องร่อนขนาดจริง ตัวแทนมีประโยชน์ก็ต่อเมื่อปริมาณที่ใช้ร่วมกันนั้นคือปริมาณที่ต้องการจริง
- **แยกตัวแปรเดียวออกมา แม้ออกแบบการทดลองเองไม่ได้** Semmelweis ไม่ได้สร้างคลินิกสองแห่งนั้น เขาแค่เห็นว่าอุบัติเหตุของตารางเวรโรงพยาบาลแยกตัวแปรไว้ให้เขาแล้ว บริษัทน้ำสองแห่งของ Snow ก็เป็นอุบัติเหตุของประวัติศาสตร์บริษัท ครึ่งหนึ่งของฝีมือในทำนั้นคือการสังเกตเห็นการทดลองตามธรรมชาติที่วิ่งอยู่แล้ว
- **ส่งสัยข้อมูลที่รับสืบทอดมา ก่อนจะสงสัยฝีมือตัวเอง** พี่น้อง Wright เสียไปทั้งฤดูกาลกับการสมมติว่าเครื่องร่อนของตัวเองสร้างผิด ก่อนจะคิดได้ว่าตารางของ Lilienthal — รากฐานร่วมของทั้งวงการ — อาจผิดแทน การเปลี่ยนกรอบมักเริ่มจากการปฏิเสธที่จะโทษตัวเองสำหรับตัวเลขห่วยๆ ของคนอื่น

- คำตอบที่ถูกยังไม่ใช่คำตอบที่เสร็จ Semmelweis พิสูจน์ข้อเสนอของเขาด้วยตัวเลขที่สะอาดเท่าตัวเลขไหนก็ตามในหนังสือเล่มนี้ แล้วก็ยังแพ้ เพราะการพิสูจน์กับการโน้มน้าวเป็นคนละปัญหาที่มีคนละทางแก้ การพาคำตอบไปวางในมือคนอื่นเป็นส่วนหนึ่งของงาน ไม่ใช่บทส่งท้ายของมัน

E N

Next — Chapter 08 · Gallery II: The Modern Solves: five more solves, all inside living memory, in which the single move is not a cheap measurement but a hidden constraint — the one condition, found and honored or found and broken, that decides everything else.

T H

ต่อไป — บทที่ 08 · หอเรื่องเล่า II: การแก้ครั้งใหญ่ยุคใหม่: อีกห้าการแก้ ทั้งหมดอยู่ในความทรงจำของคนที่ยังมีชีวิต ที่ซึ่งทำเดิวนั้นไม่ใช่การวัดราคาถูก แต่คือข้อจำกัดที่ซ่อนอยู่ — เงื่อนไขเดียวที่เมื่อหาเจอแล้วทำตาม หรือหาเจอแล้วหักทิ้ง จะชี้ขาดทุกอย่างที่เหลือ

CHAPTER 08

Gallery II: The Modern Solves

หอเรื่องเล่า II: การแก้ครั้งใหญ่ยุคใหม่

EN

The five solves in this gallery all happened within living or near-living memory, three of them broadcast or published in enough detail that we can watch the reframe happen almost in real time. The lesson changes shape from the previous gallery. Where chapter 07's solvers each found a cheap measurement to stand in for an expensive world, this gallery's solvers each found something narrower and often more decisive: the *one constraint* buried inside a sprawling problem that, once identified, made everything else fall into place — or fall apart.

TH

ทั้งห้าครั้งในหอนี้แก้กันในความทรงจำของคนที่ยังมีชีวิต หรือเพิ่งผ่านไปไม่นาน สามครั้งถูกถ่ายทอดหรือตีพิมพ์ละเอียดพอให้เฝ้าดูการเปลี่ยนกรอบ (reframe) เกิดขึ้นแทบจะสดๆ บทเรียนเปลี่ยนรูปไปจากหอก่อนหน้านี้ นักแก้ปัญหาในบทที่ 07 ต่างหาการวัดราคาถูกมาیینแทนโลกราคาแพง นักแก้ปัญหาในหอนี้หาสิ่งที่แคบกว่า และมักชี้ขาดกว่า: ข้อจำกัดข้อเดียว ที่ฝังอยู่ในปัญหาอันกว้างเกะกะ ระบุมันได้เมื่อไร ทุกอย่างที่เหลือก็เข้าที่ — หรือพังลง

The Theorem the Germans Never Saw Coming — Warsaw, 1932

ทฤษฎีบทที่เยอรมันไม่เคยคาดคิด — วอร์ซอ 1932

EN

By 1932, the German military's Enigma machine had been in service for years, and the Polish Cipher Bureau's small codebreaking staff had made almost no progress against it. Enigma's security did not rest on secrecy of design — commercial versions were sold openly — but on the machine's internal wiring, a configuration of three rotors and their electrical connections that the Poles had never seen and could not obtain. Every established approach to cryptanalysis assumed you needed either the machine or a great deal of matching plaintext and ciphertext. Neither was available in useful quantity.

TH

ถึงปี 1932 ทีมถอดรหัสขนาดเล็กของ Polish Cipher Bureau ยังแทบไม่คืบหน้ากับเครื่อง Enigma เลย ทั้งที่กองทัพเยอรมันใช้เครื่องนี้มาหลายปีแล้ว ความปลอดภัยของ Enigma ไม่ได้อยู่ที่แบบของเครื่องเป็นความลับ — รุ่นเชิงพาณิชย์วางขายอย่างเปิดเผย — แต่อยู่ที่สายไฟภายในเครื่อง คือโรเตอร์สามตัวกับวิธีต่อไฟฟ้าของมัน ฝ่ายโปแลนด์ไม่เคยเห็นและหามาไม่ได้ วิธีวิเคราะห์รหัสทุกแนวที่ตั้งมันอยู่ตั้งต้นจากข้อสมมติเดียวกัน: ต้องมีตัวเครื่อง หรือไมก็ต้องมีข้อความต้นฉบับกับข้อความเข้ารหัสที่จับคู่กันจำนวนมาก ทั้งสองอย่างไม่มีในปริมาณที่ใช้การได้

EN

Marian Rejewski, a 27-year-old mathematician recently recruited from Poznań University, took a different route entirely. German operators, following procedure, encrypted a three-letter message key twice in a row at the start of every transmission — a doubled indicator meant to guard against transmission errors. Rejewski realized that this doubling, combined with the fact that an Enigma rotor's wiring is a mathematical *permutation*, meant the relationships between the first and fourth, second and fifth, third and sixth letters of thousands of intercepted message openings encoded, in aggregate, the internal wiring of the machine itself. Working with pure permutation-group theory — and with two French-supplied operating manuals

that pinned down a few of the unknowns — Rejewski wrote out a system of equations and, by December 1932, reconstructed the wiring of a machine he had never touched (Engineering and Technology History Wiki, "First Breaking of Enigma Code by the Team of Polish Cipher Bureau, 1932–1939"; AMS, "Marian Rejewski and the First Break into Enigma"). Later popular accounts, including a widely used mathematics-education article, would call this "the theorem that won the war" — a title that belongs to a 2015 teaching piece rather than to Rejewski himself, who never claimed anything so grand, and the credit properly belongs jointly to him and his Poznań colleagues Jerzy Różycki and Henryk Zygalski, who built the mechanical and tabular tools that turned the theorem into daily practice (Mathematical Association of America, "The Theorem that Won the War").

TH

Marian Rejewski เลือกเส้นทางที่ต่างออกไปคนละเรื่อง เขาเป็นนักคณิตศาสตร์วัย 27 ปี เพิ่งถูกดึงตัวมาจากมหาวิทยาลัย Poznań พนักงานวิทยุเยอรมันทำตามระเบียบ เข้ารหัสกุญแจข้อความสามตัวอักษรซ้ำสองครั้งติดกันที่หัวของทุกการส่ง — ตัวบ่งชี้ซ้ำสองที่มีไว้กันความผิดพลาดในการส่ง Rejewski เห็นว่าการซ้ำสองนี้เปิดช่อง สายไฟในโรเตอร์ของ Enigma คือการเรียงสับเปลี่ยน (permutation) ทางคณิตศาสตร์ ดังนั้นความสัมพันธ์ระหว่างตัวอักษรที่หนึ่งกับที่สี่ ที่สองกับที่ห้า ที่สามกับที่หก จากหัวข้อความที่ดักฟังมาได้เป็นพัน เมื่อดูรวมกันย่อมเข้ารหัสสายไฟภายในของตัวเครื่องเองเอาไว้ Rejewski ใช้ทฤษฎีกรุปการเรียงสับเปลี่ยนล้วนๆ บวกกับคู่มือการใช้งานสองเล่มที่ฝรั่งเศสส่งมาให้ ซึ่งตรงค่าที่ไม่รู้ไว้ได้ไม่กี่ตัว เขาเขียนระบบสมการออกมา และภายในเดือนธันวาคม 1932 ก็ประกอบสายไฟของเครื่องที่เขาไม่เคยแตะขึ้นมาใหม่ได้ (Engineering and Technology History Wiki, "First Breaking of Enigma Code by the Team of Polish Cipher Bureau, 1932–1939"; AMS, "Marian Rejewski and the First Break into Enigma") คำบอกเล่ายอดนิยมในภายหลัง รวมถึงบทความการศึกษาคณิตศาสตร์ที่ถูกใช้อ้างกันกว้างขวาง เรียกสิ่งนี้ว่า "ทฤษฎีบทที่ชนะสงคราม" ชื่อนั้นเป็นของบทความสอนปี 2015 ไม่ใช่ของ Rejewski เอง เขาไม่เคยกล่าวอ้างอะไรใหญ่โตขนาดนั้น เครดิตที่ถูกต้องเป็นของเขา ร่วมกับเพื่อนร่วมงานจาก Poznań อีกสองคน Jerzy Różycki และ Henryk Zygalski

ผู้สร้างเครื่องมือเชิงกลและตารางที่เปลี่ยนทฤษฎีบทให้เป็นงานประจำวันได้
(Mathematical Association of America, "The Theorem that Won the War")

EN

The load-bearing constraint Rejewski found was procedural, not mechanical: a repeated indicator that operators were required to send. When Germany dropped that procedure in 1938 and tightened Enigma's security further, the Polish team's mathematical technique stopped working, and on 25–26 July 1939 — with war weeks away — Poland handed Britain and France everything it had: its reconstructed Enigma replicas, its methods, and the design for its *bomba kryptologiczna*, an electromechanical device built to automate the search (GCHQ, "The Pyry Forest Meeting"). At Bletchley Park, Alan Turing and Gordon Welchman then performed a second, equally sharp reframe on top of the first. Rather than trying to break individual messages, they built a machine — the Turing–Welchman bombe — to break the *daily* key that unlocked every message sent that day, working from "cribs," fragments of probable plaintext (a routine weather report was a favorite) whose position in a ciphertext could be guessed from operator habit (The National Museum of Computing, "The Turing–Welchman Bombe"). The load-bearing constraint here was Enigma's one mechanical guarantee: it could never encipher a letter as itself, a fact that let cribs be positioned and logical contradictions be chased electrically through loops of wiring until only one candidate rotor setting survived. Two generations of solvers, a few years apart, had each found the single condition — a doubled indicator, then a letter that could never map to itself — that made an otherwise astronomical search collapse to a morning's work.

TH

ข้อจำกัดที่รับน้ำหนักซึ่ง Rejewski หาเจอเป็นเรื่องระเบียบปฏิบัติ ไม่ใช่กลไก: ตัวบ่งชี้ที่พนักงานวิทยุถูกสั่งให้ส่งซ้ำ ปี 1938 เยอรมนีเลิกระเบียบนั้นและรัดความปลอดภัยของ Enigma ให้แน่นขึ้นอีก เทคนิคทางคณิตศาสตร์ของทีมโปแลนด์ก็หยุดทำงาน วันที่ 25–26 กรกฎาคม 1939 — เหลืออีกไม่กี่สัปดาห์ก่อนสงคราม — โปแลนด์มอบทุกอย่างที่มีให้อังกฤษและฝรั่งเศส: เครื่อง Enigma จำลองที่ประกอบขึ้นใหม่ วิธีการทั้งหมด และแบบของ *bomba kryptologiczna* อุปกรณ์ไฟฟ้ากลที่สร้างขึ้นให้ค้นหาโดยอัตโนมัติ (GCHQ, "The Pyry Forest Meeting") ที่ Bletchley Park Alan Turing

กับ Gordon Welchman เปลี่ยนกรอบครั้งที่สอง คมพอกัน ซ่อนลงบนครั้งแรก พวกเขาไม่ถอดทีละข้อความ แต่สร้างเครื่องขึ้นมา — Turing–Welchman bombe — เพื่อถอดกุญแจประจำวันที่ใช้ทุกข้อความที่ส่งในวันนั้น เครื่องทำงานจาก "crib" คือชิ้นส่วนของข้อความต้นฉบับที่น่าจะปรากฏ (รายงานอากาศประจำวันเป็นของโปรด) เดาตำแหน่งในข้อความเข้ารหัสได้จากนิสัยของพนักงานวิทยุ (The National Museum of Computing, "The Turing–Welchman Bombe") ข้อจำกัดที่รับน้ำหนักในรอบนี้คือหลักประกันเชิงกลข้อเดียวของ Enigma: มันเข้ารหัสตัวอักษรให้กลายเป็นตัวมันเองไม่ได้เลย ข้อเท็จจริงข้อนี้ทำให้วาง crib ลงตำแหน่งได้ และไล่ความขัดแย้งทางตรรกะด้วยไฟฟ้าผ่านวงจรที่ต่อเป็นลูป จนเหลือค่าตั้งโรเตอร์ผู้เข้าชิงเพียงค่าเดียว นักแก้ปัญหาสองรุ่น ห่างกันไม่กี่ปี ต่างหาเงื่อนไขเดียวเจอ — ตัวบ่งชี้ซ้ำสอง แล้วต่อมาคือตัวอักษรที่ไม่มีวันแมปเป็นตัวเอง — และเงื่อนไขนั้นทำให้การค้นหาระดับดาราศาสตร์ยุบลงเหลืองานเข้าเดียว

EN

The move: the machine was secret, but its procedure leaked a theorem.

TH

ท่า: ตัวเครื่องเป็นความลับ แต่ระเบียบปฏิบัติของมันทำทฤษฎีบทรั่วออกมา

The Mailbox Built From What Was Already Aboard — Apollo 13, 1970

ตู้ไปรษณีย์ที่สร้างจากของที่อยู่บนยานอยู่แล้ว — Apollo 13, 1970

EN

On April 13, 1970, an oxygen tank exploded aboard Apollo 13, roughly 200,000 miles from Earth, crippling the command module and forcing all three astronauts into the lunar module — a craft built as a two-man, two-day lifeboat, now asked to keep three men alive for four. Within days a slower, quieter emergency was building alongside the obvious ones: carbon dioxide. The lunar module's lithium hydroxide canisters, which scrubbed CO₂ from the cabin air, were designed for two occupants and were running out; the command module carried plenty of spare canisters, but they were square, built for a different air-scrubbing system, and the lunar module's

receptacles were round (Space Center Houston, "Apollo 13 Infographic: How did they make that CO2 scrubber?").

TH

วันที่ 13 เมษายน 1970 ถึงออกซิเจนระเบิดบนยาน Apollo 13 ห่างจากโลกราวสองแสนไมล์ โมดูลบัญชาการใช้การไม่ได้ นักบินอวกาศทั้งสามคนต้องย้ายเข้าโมดูลลงดวงจันทร์ — เรือชูชีพที่สร้างมาสำหรับสองคน สองวัน แต่ตอนนี้ต้องประคองสามคนไว้สี่วัน ภายในไม่กี่วัน เหตุฉุกเฉินที่ซ้ำกว่าและเจ็บกว่าก่อตัวขึ้นเคียงข้างเหตุที่เห็นชัดอยู่แล้ว: คาร์บอนไดออกไซด์ กระบอกลิเทียมไฮดรอกไซด์ของโมดูลลงดวงจันทร์ดูด CO2 ออกจากอากาศในห้องโดยสาร แต่ออกแบบมาสำหรับคนสองคนและกำลังจะหมดฤทธิ์ โมดูลบัญชาการมีกระบอกสำรองอยู่มากมาย ทว่าเป็นทรงสี่เหลี่ยมสร้างมาสำหรับระบบฟอกอากาศคนละแบบ ส่วนช่องรับของโมดูลลงดวงจันทร์เป็นทรงกลม (Space Center Houston, "Apollo 13 Infographic: How did they make that CO2 scrubber?")

EN

The constraint that mattered was not chemistry — lithium hydroxide would happily absorb CO2 in a square or round housing alike — it was geometry, and one further condition sat underneath it, tighter than the shape of a filter: the solution had to be built entirely from objects already on board a spacecraft in flight. Nothing could be sent up. A NASA engineer named Ed Smylie, who led the crew systems team, gathered a small group and imposed exactly that rule, then had them lay out, on a table, only what the astronauts had within reach: plastic bags meant for spacesuit liners, cardboard ripped from the covers of a flight plan, a roll of gray duct tape, and the hose from a spacesuit (The Washington Post, "Ed Smylie, engineer who helped save Apollo 13 crew with duct tape, dies"). Duct tape, notably, was the one material that could seal a joint reliably in zero gravity by unskilled hands under stress — which made it, on this flight, a load-bearing engineering material rather than a household convenience.

TH

ข้อจำกัดที่สำคัญไม่ใช่เคมี แต่คือเรขาคณิต — ลิเทียมไฮดรอกไซด์ดูด CO2 ได้สบายไม่ว่าตัวเรือนจะทรงสี่เหลี่ยมหรือทรงกลม — และได้เรขาคณียังมีเงื่อนไขอีกข้อเน้นกว่ารูปทรงของไส้กรอง: ทางแก้ต้องสร้างจากสิ่งของที่อยู่บนยานซึ่งกำลังบินอยู่

แล้วทั้งหมด ส่งอะไรขึ้นไปไม่ได้เลย Ed Smylie วิศวกรของ NASA ผู้นำทีมระบบ ลูกเรือ รวบรวมคนกลุ่มเล็กๆ แล้วตั้งกฎข้อนั้นลงไปตรงๆ จากนั้นให้วางลงบนโต๊ะ เฉพาะสิ่งที่นักบินอวกาศเอื้อมถึง: ถุงพลาสติกสำหรับซับในชุดอวกาศ กระดาษแข็งที่ฉีกมาจากปกแผนการบิน เทปกาวสีเทาหนึ่งม้วน และท่อจากชุดอวกาศ (The Washington Post, "Ed Smylie, engineer who helped save Apollo 13 crew with duct tape, dies") เทปกาวเป็นวัสดุอย่างเดียวที่ปิดผนึกรอยต่อได้อย่างเชื่อถือได้ในสภาวะไร้น้ำหนัก ด้วยมือของคนที่ไม่ได้ฝึกมาและอยู่ใต้ความกดดัน ในเที่ยวบินนี้มันจึงเป็นวัสดุวิศวกรรมที่รับน้ำหนัก ไม่ใช่ของอำนวยความสะดวกในบ้าน

EN

The team built and tested a working adapter — soon nicknamed "the mailbox" — on the ground in under a day, then radioed step-by-step assembly instructions up to the crew, who built their own copy from identical spare parts already stowed aboard (NASA, "Apollo 13 Lunar Module 'Mail Box'"). Cabin CO2 partial pressure, which mission rules treated as safe below about 8 mm Hg and dangerous approaching 15, had been climbing toward that danger zone with no working scrubber; once the crew wired the mailbox into the lunar module's cabin fan, the reading fell and held below 2 mm Hg for the rest of the flight home (Universe Today, "13 MORE Things That Saved Apollo 13, part 5: The CO2 Partial Pressure Sensor"). What makes this a re-frame rather than a repair is the discipline mission control imposed on itself before any design work started: every other spaceflight problem admits, in principle, an answer involving more material, more mass, more redundancy sent up in advance. This one did not. By declaring the parts list closed before the design began, Houston turned an open-ended engineering problem — build a working scrubber — into a narrow, solvable one: build a working scrubber *from this exact list of thirteen items*, a constraint tight enough that a handful of engineers could exhaust the possibilities in an afternoon.

TH

ทีมสร้างและทดสอบตัวแปลงที่ใช้งานได้จริงบนพื้นโลกภายในไม่ถึงหนึ่งวัน — ไม่นานมันก็ถูกเรียกเล่นๆ ว่า "ตู้ไปรษณีย์" — แล้วส่งวิทยุบอกขั้นตอนการประกอบทีละชิ้นขึ้นไปให้ลูกเรือ ลูกเรือสร้างของตัวเองจากอะไหล่แบบเดียวกันที่เก็บไว้บนยานอยู่

แล้ว (NASA, "Apollo 13 Lunar Module 'Mail Box'") ความดันย่อยของ CO2 ในห้องโดยสารได้ขึ้นหาเขตรันตรายเรื่อยๆ ในเมื่อไม่มีเครื่องฟอกที่ใช้การได้ กฎภารกิจถือว่าปลอดภัยเมื่อต่ำกว่าราว 8 มิลลิเมตรปรอท และอันตรายเมื่อเข้าใกล้ 15 พอลุกเรือต่อสู้ไปรษณีย์เข้ากับพัดลมห้องโดยสารของโมดูลลงดวงจันทร์ ค่าที่อ่านได้ก็ลดลงและอยู่ต่ำกว่า 2 มิลลิเมตรปรอทตลอดการเดินทางกลับบ้าน (Universe Today, "13 MORE Things That Saved Apollo 13, part 5: The CO2 Partial Pressure Sensor") นี่เป็นการเปลี่ยนกรอบ ไม่ใช่แค่การซ่อม เพราะวินัยที่ศูนย์ควบคุมภารกิจบังคับกับตัวเองก่อนงานออกแบบจะเริ่ม: ปัญหาการบินอวกาศข้ออื่นทุกข้อ ในหลักการยอมรับคำตอบที่ส่งวัสดุเพิ่ม มวลเพิ่ม ความสำรองเพิ่ม ขึ้นไปล่วงหน้าได้ ข้อนี้ไม่ยอม Houston ประกาศปิดรายการขึ้นส่วนก่อนเริ่มออกแบบ ปัญหาวิศวกรรมปลายเปิด — สร้างเครื่องฟอกที่ใช้ได้ — จึงกลายเป็นปัญหาที่แคบและแก้ได้: สร้างเครื่องฟอกที่ใช้ได้จากรายการสืบทอดชิ้นนี้เท่านั้น ข้อจำกัดแน่นพอให้วิศวกรหยิบมือหนึ่งไล่ความเป็นไปได้จนหมดในบ่ายเดียว

E N

The move: nothing could go up, so nothing except what was already there was allowed to count.

T H

ท่า: ไม่มีอะไรส่งขึ้นไปได้ ฉะนั้นไม่มีอะไรนับได้นอกจากของที่อยู่บนนั้นแล้ว

A Clamp, a Glass of Ice Water, and a Committee — Washington, 1986

แคลมป์ตัวหนึ่ง น้ำแข็งแก้วหนึ่ง กับคณะกรรมการชุดหนึ่ง — วอชิงตัน 1986

EN

The Space Shuttle *Challenger* broke apart 73 seconds after launch on January 28, 1986, killing all seven crew members, and the presidential commission convened to investigate — chaired by former Secretary of State William Rogers — spent weeks accumulating testimony, telemetry, and competing engineering theories about what had failed. The physicist Richard Feynman, appointed to the commission somewhat reluctantly, grew frustrated with a process he felt was drowning a simple mechanical question in procedural caution.

TH

กระสวยอวกาศ Challenger แตกออกเป็นชิ้น 73 วินาทีหลังปล่อยตัว วันที่ 28 มกราคม 1986 ลูกเรือทั้งเจ็ดคนเสียชีวิต คณะกรรมการธิการที่ประธานาธิบดีตั้งขึ้นเพื่อสอบสวน — William Rogers อดีตรัฐมนตรีต่างประเทศเป็นประธาน — ใช้เวลาหลายสัปดาห์สะสมคำให้การ ข้อมูลโทรมาตร และทฤษฎีวิศวกรรมที่ขัดแย้งกันว่าอะไรพัง Richard Feynman นักฟิสิกส์ ถูกแต่งตั้งเข้าคณะกรรมการอย่างไม่ค่อยเต็มใจนัก เขาหงุดหงิดกับกระบวนการที่ในสายตาเขากำลังจมคำถามเชิงกลไกง่ายๆ ข้อหนึ่งลงในความระมัดระวังเชิงระเบียบ

EN

The question Feynman wanted answered was narrow: did the rubber O-rings sealing the joints of the solid rocket boosters lose their resiliency — their ability to spring back into shape after being compressed — at the unusually cold temperature on the morning of the launch, around 36°F (2°C)? He arrived at a public hearing on February 11, 1986, with a piece of O-ring material, a small C-clamp bought that morning at a hardware store, and a glass of ice water requested from the hearing staff. During testimony from a NASA engineer, Feynman clamped the sample, submerged it in the ice water for a few minutes, then released the clamp on camera: the rubber stayed compressed instead of springing back, demonstrating in under a minute, in front of live television cameras, that cold rubber loses exactly the elastic

property the seal depended on (Caltech, Feynman's personal appendix to the Rogers Commission report, "F.: An Additional Comment"; NASA, "Report of the Presidential Commission on the Space Shuttle Challenger Accident," Volume IV).

TH

คำถามที่ Feynman อยากได้คำตอบนั้นแคบ: O-ring Yang ที่ปิดผนึกรอยต่อของจรวด ขั้วต้นเชื้อเพลิงแข็ง สูญเสียความยืดหยุ่นตัว — ความสามารถติดกลับคืนรูปหลังถูกกด — ที่อุณหภูมิเย็นผิดปกติของเข้าวันปล่อยตัว ราว 36 องศาฟาเรนไฮต์ (2 องศาเซลเซียส) หรือไม่ วันที่ 11 กุมภาพันธ์ 1986 เขามาถึงการไต่สวนสาธารณะพร้อมชิ้นวัสดุ O-ring แคลมป์รูปตัวซีอันเล็กที่ซื้อจากร้านฮาร์ดแวร์เข้าวันนั้น และน้ำแข็งแก้วหนึ่งที่ขโมยจากเจ้าหน้าที่ห้องไต่สวน ระหว่างวิศวกรของ NASA กำลังให้การ Feynman หนีบตัวอย่างด้วยแคลมป์ จุ่มลงในน้ำแข็งไม่กี่นาที แล้วคลายแคลมป์ออกต่อหน้ากล้อง: ยางยังถูกกดแบนอยู่ ไม่ติดกลับ ภายในไม่ถึงหนึ่งนาที ต่อหน้ากล้องโทรทัศน์ที่ถ่ายทอดสด เขาแสดงให้เห็นว่ายางที่เย็นจัดสูญเสียคุณสมบัติยืดหยุ่นข้อที่ซิลินั้นพึ่งพาอยู่พอดี (Caltech, ภาคผนวกส่วนตัวของ Feynman ในรายงานคณะกรรมการ Rogers, "F.: An Additional Comment"; NASA, "Report of the Presidential Commission on the Space Shuttle Challenger Accident," Volume IV)

EN

The demonstration is often told as a lone genius's improvisation, and Feynman's showmanship was certainly his own. But the tip that let him aim so precisely came from elsewhere, and the commission's own members kept that quiet for over two decades. Astronaut Sally Ride, also a commission member, had been shown internal NASA data linking O-ring damage to low launch temperatures by a source she wanted to protect; rather than raise it herself, she passed the chart to Air Force general Donald Kutyna, another commissioner. Kutyna, in turn, invited Feynman to his garage, showed him a leaking carburetor on a car he was restoring, and asked, apparently offhand, whether cold might affect an O-ring's resilience the same way — a question that pointed Feynman directly at the joint's temperature sensitivity without ever mentioning Ride's data (Britannica, "Sally Ride"). Only after Ride's death in 2012 did Kutyna disclose her role; Feynman, who died in 1988, may never

have known exactly whose evidence had reached him. The load-bearing constraint of the entire disaster — a rubber seal's resiliency threshold, crossed by a specific, knowable number of degrees — existed in NASA's own data well before the launch. What the ice-water demonstration did was compress a multi-volume technical argument that risked disappearing into bureaucratic hedging into thirty seconds that a television audience, a Congress, and a grieving public could see and understand without a single equation.

TH

เบาะแสที่ทำให้ Feynman เล็งได้แม่นยำขนาดนั้นมาจากที่อื่น และกรรมวิธีการด้วยกันเองเก็บเรื่องนี้เงียบไว้กว่าสองทศวรรษ การสาธิตมักถูกเล่าเป็นการด้นสดของอัจฉริยะผู้โดดเดี่ยว และความเป็นนักแสดงของ Feynman ก็เป็นของเขาเองแน่นอน Sally Ride นักบินอวกาศและกรรมวิธีการอีกคน ได้เห็นข้อมูลภายในของ NASA ที่โยงความเสียหายของ O-ring เข้ากับอุณหภูมิต่ำในวันปล่อยตัว จากแหล่งข่าวที่เธอต้องการปกป้อง เธอไม่หือยยกขึ้นมาเอง แต่ส่งแผนภูมินั้นต่อให้ Donald Kutyna นายพลกองทัพอากาศ กรรมวิธีการอีกคนหนึ่ง Kutyna ชวน Feynman ไปที่โรงรถของเขา ให้ดูคาร์บูเรเตอร์ที่รื้อบนรถที่เขากำลังบูรณะอยู่ แล้วถามขึ้นแบบที่ดูเหมือนลอยๆ ว่าความเย็นจะกระทบความยืดหยุ่นตัวของ O-ring แบบเดียวกันหรือเปล่า — คำถามที่ชี้ Feynman ตรงไปยังจุดที่รอยต่อไวต่ออุณหภูมิ โดยไม่เอ่ยถึงข้อมูลของ Ride เลยสักคำ (Britannica, "Sally Ride") Kutyna เปิดเผยแพร่บทบาทของเธอหลัง Ride เสียชีวิตในปี 2012 เท่านั้น Feynman เสียชีวิตในปี 1988 และอาจไม่เคยรู้เลยว่าหลักฐานที่มาถึงเขาเป็นของใครกันแน่ ข้อจำกัดที่รับน้ำหนักของหายนะทั้งครั้ง — ชีตเริ่มความยืดหยุ่นตัวของซีลยาง ถูกข้ามด้วยจำนวนองศาที่เจาะจงและรู้ได้ — อยู่ในข้อมูลของ NASA เองมานานก่อนการปล่อยตัว การสาธิตด้วยน้ำแข็งบีบข้อโต้แย้งทางเทคนิคหนาหลายเล่ม ที่เสี่ยงจะหายไปในการพุดคำกวมแบบราชการ ให้เหลือสามสิบวินาที ผู้ชมโทรทัศน์ สภาคองเกรส และสาธารณชนที่กำลังโศกเศร้า มองเห็นและเข้าใจได้โดยไม่ต้องมีสมการสักตัว

EN

The move: the failure had one number attached to it, and a glass of ice water made the number visible.

TH

ท่า: ความล้มเหลวมีตัวเลขติดอยู่ตัวเดียว และน้ำแข็งแก้วเดียวทำให้ตัวเลขนั้นมองเห็นได้

Seven Years in an Attic — Cambridge and Princeton, 1986–1995

เจ็ดปีในห้องใต้หลังคา — Cambridge กับ Princeton 1986–1995

EN

Fermat's Last Theorem — that no three positive integers a , b , c can satisfy $a^n + b^n = c^n$ for any integer n greater than 2 — had resisted proof for over 350 years since Pierre de Fermat scrawled his famous, almost certainly mistaken claim to have found "a truly marvelous proof" in the margin of a book in 1637. By the twentieth century it had become less a live research problem than a graveyard of failed attempts, the kind of statement mathematicians warned graduate students away from.

TH

ทฤษฎีบทสุดท้ายของ Fermat ด้านทานการพิสูจน์มากกว่า 350 ปี ตัวทฤษฎีบทว่าไม่มีจำนวนเต็มบวกสามจำนวน a , b , c ใดสอดคล้องกับ $a^n + b^n = c^n$ สำหรับจำนวนเต็ม n ที่มากกว่า 2 ปี 1637 Pierre de Fermat เขียนไว้ตรงขอบกระดาษของหนังสือเล่มหนึ่งว่าเขาพบ "บทพิสูจน์ที่งดงามอย่างแท้จริง" ("a truly marvelous proof") คำกล่าวอ้างอันโด่งดังนั้นแทบแน่นอนว่าผิด พอถึงศตวรรษที่ยี่สิบ มันเป็นสุสานของความพยายามที่ล้มเหลวมากกว่าปัญหาวิจัยที่ยังมีชีวิต เป็นข้อความประเภทที่นักคณิตศาสตร์เตือนนักศึกษาบัณฑิตให้ยั้งไว้ห่าง

EN

In 1986, mathematicians Gerhard Frey, Jean-Pierre Serre, and Ken Ribet established a bridge that changed the problem's shape entirely. They showed that if the Taniyama–Shimura conjecture — a deep statement claiming that every *elliptic curve* over the rational numbers is secretly connected to a highly structured object called a modular form — were true for a specific

restricted class of curves (semistable curves), then Fermat's Last Theorem would follow as a corollary. In other words: stop trying to prove Fermat's statement about three numbers and an exponent. Prove instead that a hypothetical counterexample to Fermat would generate an elliptic curve so strange it could not possibly be modular — and prove separately that no such non-modular curve can exist. Andrew Wiles, a number theorist at Princeton who had been fascinated by Fermat's Last Theorem since childhood, recognized that this reframe had, for the first time, connected the old problem to the living, well-resourced center of modern number theory. He set out in near-total secrecy, working essentially alone in the attic study of his house for seven years, to prove the modularity conjecture for semistable curves — enough to make Fermat fall out for free (Wikipedia, "Wiles's proof of Fermat's Last Theorem"). He told almost no one what he was working on — not department colleagues, not most collaborators, only his wife Nada, whom he later said "has only known me while I was working on Fermat" — periodically surfacing old, unrelated results to keep up the appearance of ordinary output while the real work stayed hidden (Nautilus, "How Math's Most Famous Proof Nearly Broke").

TH

ปี 1986 นักคณิตศาสตร์ Gerhard Frey, Jean-Pierre Serre และ Ken Ribet วางสะพานที่เปลี่ยนรูปทรงของปัญหาไปทั้งหมด พวกเขาแสดงว่าถ้าข้อคาดการณ์ Taniyama–Shimura เป็นจริงสำหรับเส้นโค้งกลุ่มจำกัดกลุ่มหนึ่ง (semistable curve) ทฤษฎีบทสุดท้ายของ Fermat จะตามมาเป็นบทแทรก ข้อคาดการณ์นั้นเป็นข้อความเชิงลึก: เส้นโค้งเชิงวงรี (elliptic curve) ทุกเส้นบนจำนวนตรรกยะ เชื่อมโยงอย่างลึกๆ กับวัตถุที่มีโครงสร้างสูงซึ่งเรียกว่า modular form พูดอีกอย่าง: เลิกพยายามพิสูจน์ข้อความของ Fermat เรื่องตัวเลขสามตัวกับเลขชี้กำลัง พิสูจน์แทนว่าตัวอย่างค้านสมมติของ Fermat จะสร้างเส้นโค้งเชิงวงรีที่ประหลาดจนเป็น modular ไม่ได้ — และพิสูจน์แยกอีกทางว่าเส้นโค้งที่ไม่เป็น modular แบบนั้นมีอยู่ไม่ได้ Andrew Wiles นักทฤษฎีจำนวนที่ Princeton หลงใหลทฤษฎีบทสุดท้ายของ Fermat มาตั้งแต่เด็ก เขาเห็นว่าการเปลี่ยนกรอบนี้เชื่อมปัญหาเก่าเข้ากับใจกลางที่ยังมีชีวิตและทรัพยากรพร้อมของทฤษฎีจำนวนสมัยใหม่ได้เป็นครั้งแรก เขาลงมือในความลับเกือบสมบูรณ์ ทำงานตามลำพังจริงๆ ในห้องทำงานใต้หลังคาบ้านตัวเองเจ็ดปี เพื่อ

พิสูจน์ข้อคาดการณ์มอดูลาริตีสำหรับเส้นโค้ง semistable — มากพอให้ Fermat หลุดออกมาฟรีๆ (Wikipedia, "Wiles's proof of Fermat's Last Theorem") เขาแทบไม่บอกใครเลยว่ากำลังทำอะไรอยู่ — ไม่บอกเพื่อนร่วมภาควิชา ไม่บอกผู้ร่วมงานส่วนใหญ่ บอกแต่ Nada ภรรยาของเขา ผู้ซึ่งเขาพูดในภายหลังว่า "รู้จักผมเฉพาะช่วงที่ผมทำงานเรื่อง Fermat" ("has only known me while I was working on Fermat") — และปล่อยผลงานเก่าที่ไม่เกี่ยวข้องออกมาเป็นระยะ เพื่อรักษาภาพว่ายังผลิตงานตามปกติ ขณะที่งานจริงซ่อนอยู่ (Nautilus, "How Math's Most Famous Proof Nearly Broke")

EN

Wiles announced a complete proof across three lectures at Cambridge's Isaac Newton Institute in June 1993 — reportedly finishing the final line of the argument in the closing minutes of the final lecture, to a stunned silence and then a standing ovation from an audience that had gradually realized, over three days, what it was witnessing — and the announcement drew global press coverage, with the *New York Times* putting it on the front page. Then, during the routine but rigorous process of peer review, a referee named Nick Katz found a gap: a step in the argument's construction of an "Euler system," a technical device load-bearing enough that without it the whole edifice risked being merely a very good sketch rather than a proof (American Mathematical Society, Faltings, "The Proof of Fermat's Last Theorem by R. Taylor and A. Wiles," *Notices of the AMS*, 1995). For over a year, Wiles worked to repair the gap, at first alone and then, from summer 1994, with his former student Richard Taylor. On September 19, 1994, examining precisely the approach he had abandoned years earlier for a different reason, Wiles saw why it had failed before and why, combined with the Euler-system machinery, it now succeeded — closing the gap with a method distinct enough from the original that Taylor and Wiles published it as a companion paper. The corrected proof appeared in 1995 (Wikipedia).

TH

Wiles ประกาศบทพิสูจน์ที่สมบูรณ์ในการบรรยายสามครั้งที่ Isaac Newton Institute ของ Cambridge เดือนมิถุนายน 1993 มีรายงานว่าเขาเขียนบรรทัดสุดท้ายของข้อโต้แย้งจบในนาทีสุดท้ายของการบรรยายครั้งสุดท้าย ห้องเงียบไปครู่

หนึ่ง แล้วผู้ฟังลุกขึ้นยืนปรบมือ พวกเขาค่อยๆ เข้าใจตลอดสามวันนั้นว่าตัวเองกำลังเป็นพยานให้อะไร ขาวนี้ขึ้นชื่อทั่วโลก New York Times ลงหน้าหนึ่ง แล้วช่องโทรทัศน์ในชั้น peer review งานประจำที่เข้มงวด ผู้ตรวจชื่อ Nick Katz พบว่าขั้นตอนหนึ่งในการสร้าง "Euler system" ของข้อโต้แย้งมีปัญหา อุปกรณ์ทางเทคนิคขั้นนี้รับน้ำหนักมากพอที่ถ้าขาดไป สิ่งปลูกสร้างทั้งหลายก็เสี่ยงจะเป็นแค่ร่างที่ดีมากร่างหนึ่ง ไม่ใช่บทพิสูจน์ (American Mathematical Society, Faltings, "The Proof of Fermat's Last Theorem by R. Taylor and A. Wiles," Notices of the AMS, 1995) Wiles ใช้เวลากว่าหนึ่งปีซ่อมช่องโหว่นั้น ตอนแรกทำคนเดียว ตั้งแต่ฤดูร้อนปี 1994 จึงทำร่วมกับ Richard Taylor อดีตลูกศิษย์ของเขา วันที่ 19 กันยายน 1994 ขณะตรวจดูแนวทางที่เขาเคยทิ้งไปหลายปีก่อนด้วยเหตุผลอื่นพอดี Wiles เห็นว่าทำไมมันถึงล้มเหลวในตอนนั้น และทำไมพอรวมกับกลไก Euler system แล้วมันถึงสำเร็จตอนนี้ วิธีปิดช่องโหว่นี้ต่างจากของเดิมมากพอจน Taylor กับ Wiles ตีพิมพ์มันเป็นบทความคู่กัน บทพิสูจน์ที่แก้แล้วออกมาในปี 1995 (Wikipedia)

E N

The gap is not an embarrassing footnote to this story; it is the story's clearest illustration of the book's larger claim. Wiles did not find the short path by avoiding error — he found it by refusing to let one uncorrected error force him back to the *old* problem space (attacking Fermat's equation directly), when the error was local to one technical tool inside the *new* one (the Euler system within the modularity bridge). The load-bearing constraint of the whole seven-year project was never Fermat's equation. It was always the bridge — and a bridge with one weak plank is repaired, not abandoned.

T H

ช่องโหว่นั้นไม่ใช่เชิงอรรถนำอายุของเรื่องนี้ มันคือภาพประกอบที่ชัดที่สุดของข้อกล่าวอ้างใหญ่ของหนังสือเล่มนี้ Wiles ไม่ได้เจอเส้นทางสั้นด้วยการหลบข้อผิดพลาด เขาเจอมันด้วยการไม่ยอมให้ข้อผิดพลาดที่ยังไม่ได้แก้ข้อเดียวไล่เขากลับไปยังพื้นที่ปัญหา (problem space) เดิม (บุกสมการของ Fermat ตรงๆ) ทั้งที่ข้อผิดพลาดนั้นอยู่เฉพาะที่เครื่องมือทางเทคนิคขึ้นเดียวภายในพื้นที่ใหม่ (Euler system ภายในสะพานมอดูลาริตี) ข้อจำกัดที่รับน้ำหนักของโครงการเจ็ดปีทั้งโครงการไม่เคยเป็นสมการของ Fermat มันคือสะพานมาตลอด — และสะพานที่มีไม้ผุอยู่แผ่นเดียว ต้องซ่อม ไม่ใช่ทิ้ง

E N

The move: don't lift the boulder; prove the bridge that makes the boulder irrelevant.

T H

ท่า: ย่อย้ายก้อนหิน จงพิสูจน์สะพานที่ทำให้ก้อนหินไม่สำคัญอีกต่อไป

A Fifty-Year Problem Closed by Ignoring Physics — CASP14, 2020

ปัญหาห้าสิบปีที่ถูกปิดด้วยการเมินฟิสิกส์ — CASP14, 2020

E N

Predicting the three-dimensional shape a protein folds into, given only the one-dimensional sequence of amino acids that builds it, had stood as one of biology's grand open problems since the molecular biologist Cyrus Levinth-

al pointed out in 1969 that a protein could not possibly be finding its shape by randomly trying configurations — the number of possible foldings for even a modest protein chain is so large that exhaustive search would take vastly longer than the age of the universe, yet real proteins fold correctly in milliseconds (Wikipedia, "Levinthal's paradox"). Since 1994, the biennial Critical Assessment of protein Structure Prediction (CASP) competition had measured the field's slow, incremental progress against experimentally solved structures kept secret from competitors until scoring (CASP history, cited via DeepMind, "AlphaFold: a solution to a 50-year-old grand challenge in biology").

T H

การทำนายรูปร่างสามมิติที่โปรตีนพับตัวเข้าไป จากลำดับกรดอะมิโนหนึ่งมิติที่ประกอบมันขึ้นมาเพียงอย่างเดียว ยืนยันเป็นหนึ่งในปัญหาเปิดใหญ่ของชีววิทยามาตั้งแต่ปี 1969 ปีนั้น Cyrus Levinthal นักชีววิทยาโมเลกุล ชี้ว่าโปรตีนไม่มีทางหารูปร่างของตัวเองเจอด้วยการสุ่มลองไปเรื่อยๆ — จำนวนวิธีพับที่เป็นไปได้ของสายโปรตีนแม้เพียงขนาดพอประมาณก็ใหญ่จนการค้นหาแบบไล่ทุกทางกินเวลานานกว่าอายุของเอกภพไปไกล ทว่าโปรตีนจริงพับถูกต้องภายในไม่กี่มิลลิวินาที (Wikipedia, "Levinthal's paradox") ตั้งแต่ปี 1994 การแข่งขัน Critical Assessment of protein Structure Prediction (CASP) จัดทุกสองปี วัดความคืบหน้าอันแข็งแกร่งที่ละนิดของวงการนี้ เทียบกับโครงสร้างที่หาคำตอบได้จากการทดลอง และปิดเป็นความลับจากผู้แข่งขันจนกว่าจะถึงเวลาให้คะแนน (ประวัติ CASP, อ้างผ่าน DeepMind, "AlphaFold: a solution to a 50-year-old grand challenge in biology")

E N

Most serious attempts at the problem tried, in one form or another, to simulate the underlying physics: model the forces between atoms and let a folding trajectory play out computationally, an approach that is faithful to how nature actually does it but computationally so expensive that it remained impractical for anything but small proteins over short timescales. DeepMind's AlphaFold team, entering CASP13 in 2018 and then rebuilding the system almost entirely for CASP14 in 2020, made a different bet: rather than simulate the physical process of folding, treat structure prediction as a problem of *completing a pattern* buried in evolutionary history. A protein

family that has been conserved across millions of years of evolution carries statistical fingerprints in its sequence variations — when one amino acid position mutates, a second, often physically distant position frequently mutates to compensate, because the two must remain close together in the folded, functional structure for the protein to keep working. AlphaFold's neural network learned to read those co-evolutionary correlations out of large alignments of related sequences and infer three-dimensional distances between amino acids directly, without ever simulating a single physical folding step (accounts of the co-evolutionary approach summarized in reviews of AlphaFold's method, e.g. PMC, "Chasing long-range evolutionary couplings in the AlphaFold era").

TH

ความพยายามจริงจังส่วนใหญ่ต่อปัญหานี้จำลองฟิสิกส์ที่อยู่เบื้องล่างในรูปแบบใดรูปแบบหนึ่ง: สร้างแบบจำลองแรงระหว่างอะตอม แล้วปล่อยให้วิถีการพับดำเนินไปในเครื่องคำนวณ แนวทางนี้เชื่อถือตรงต่อวิธีที่ธรรมชาติทำจริง แต่แพงเชิงการคำนวณ จนใช้จริงได้แค่กับโปรตีนขนาดเล็กในช่วงเวลาสั้นๆ ทีม AlphaFold ของ DeepMind เติบโตขึ้นอีกทาง ทีมลงแข่ง CASP13 ในปี 2018 แล้วรี้อระบบสร้างใหม่แทบทั้งหมดสำหรับ CASP14 ในปี 2020 แทนที่จะจำลองกระบวนการทางกายภาพของการพับ พวกเขาถือว่าการทำนายโครงสร้างคือปัญหาของการเติมรูปแบบให้ครบ ที่ฝังอยู่ในประวัติศาสตร์วิวัฒนาการ ตระกูลโปรตีนที่ถูกอนุรักษ์ไว้ตลอดวิวัฒนาการหลายล้านปีปกคลุมนี้อาจมีทางสถิติไว้ในความแปรผันของลำดับ — เมื่อกรดอะมิโนตำแหน่งหนึ่งกลายพันธุ์ อีกตำแหน่งหนึ่งที่มักอยู่ห่างออกไปในเชิงกายภาพจะกลายพันธุ์ตามเพื่อชดเชยอยู่บ่อยครั้ง เพราะทั้งสองต้องอยู่ชิดกันในโครงสร้างที่พับแล้วและทำงานได้ไม่อย่างนั้นโปรตีนก็ทำงานต่อไม่ได้ โครงข่ายประสาทของ AlphaFold เรียนรู้ที่จะอ่านสหสัมพันธ์เชิงวิวัฒนาการร่วมเหล่านั้นจากการเรียงเทียบลำดับที่เกี่ยวข้องกันเป็นชุดใหญ่ แล้วอนุมานระยะสามมิติระหว่างกรดอะมิโนได้โดยตรง ไม่เคยจำลองขั้นตอนการพับทางกายภาพแม้แต่ขั้นเดียว (คำอธิบายแนวทางเชิงวิวัฒนาการร่วมสรุปไว้ในบทปริทัศน์วิธีของ AlphaFold เช่น PMC, "Chasing long-range evolutionary couplings in the AlphaFold era")

EN

The result, announced at CASP14 on November 30, 2020, startled a field accustomed to grinding, decade-scale progress: AlphaFold achieved a median score of about 92.4 on the Global Distance Test across the competition's target proteins, a score competitive with the resolution of experimental methods like X-ray crystallography, and it did so consistently across proteins the model had never seen (C&EN, "Baker, Hassabis, and Jumper win 2024 Nobel Prize in Chemistry"). John Moult, the computational biologist who co-founded CASP in 1994 specifically to measure progress on this problem, told *Nature* at the time: "This is a big deal... In some sense the problem is solved" — a verdict CASP's own founder is entitled to render, and one he still qualified with "in some sense" (*Nature*, Callaway, "It will change everything": DeepMind's AI makes gigantic leap in solving protein structures," 30 November 2020). In October 2024, the Nobel Prize in Chemistry recognized DeepMind's Demis Hassabis and John Jumper for AlphaFold, sharing the prize with University of Washington biochemist David Baker for related work in computational protein design (*Nature*, "Chemistry Nobel goes to developers of AlphaFold AI that predicts protein structures").

TH

AlphaFold ทำคะแนนมัธยฐานได้ราว 92.4 บน Global Distance Test หัวโปรตีนเป้าหมายของการแข่งขัน ผลประกาศใน CASP14 วันที่ 30 พฤศจิกายน 2020 และทำให้วงการที่ขึ้นกับความคืบหน้าอันเชื่องช้าระดับทศวรรษถึงกับสะดุ้ง คะแนนนั้นสู้ได้กับความละเอียดของวิธีเชิงทดลองอย่างการเลี้ยวเบนรังสีเอกซ์จากผลึก และมันทำได้สม่ำเสมอกับโปรตีนที่โมเดลไม่เคยเห็นมาก่อน (C&EN, "Baker, Hassabis, and Jumper win 2024 Nobel Prize in Chemistry") John Moult นักชีววิทยาเชิงคำนวณ ร่วมก่อตั้ง CASP ในปี 1994 ขึ้นมาเพื่อวัดความคืบหน้าของปัญหานี้โดยเฉพาะ เขาบอกกับ *Nature* ในตอนนั้นว่า "นี่เรื่องใหญ่... ในความหมายหนึ่ง ปัญหานี้ถูกแก้แล้ว" ("This is a big deal... In some sense the problem is solved") — คำตัดสินที่ผู้ก่อตั้ง CASP เองมีสิทธิ์ออก และเขายังกำกับมันไว้ด้วยคำว่า "ในความหมายหนึ่ง" (*Nature*, Callaway, "It will change everything": DeepMind's AI makes gigantic leap in solving protein structures," 30 November 2020) เดือนตุลาคม 2024 รางวัลโนเบลสาขาเคมียกย่อง Demis Hassabis กับ John Jumper แห่ง DeepMind สำหรับ AlphaFold แบ่งรางวัลร่วมกับ David Baker นักชีวเคมีแห่ง

University of Washington สำหรับงานที่เกี่ยวข้องด้านการออกแบบโปรตีนเชิงคำนวณ (Nature, "Chemistry Nobel goes to developers of AlphaFold AI that predicts protein structures")

E N

The honest boundary of the achievement matters as much as its size. AlphaFold predicts a static structure — the shape a protein settles into — not the pathway or timing by which it gets there, and it struggles with proteins that adopt multiple functional shapes, with the effects of mutations and binding partners on structure, and with the roughly one-third of the human proteome made up of intrinsically disordered regions that never settle into one fixed shape at all (PubMed, "AlphaFold and Protein Folding: Not Dead Yet! The Frontier Is Conformational Ensembles"). Levinthal's paradox — how folding happens so fast — remains genuinely unsolved; AlphaFold answered a narrower, extremely valuable question (what shape does this sequence produce) by declining to answer the dynamical one at all.

T H

ขอบเขตที่ชัดเจนของความสำเร็จนี้สำคัญพอๆ กับขนาดของมัน AlphaFold ทำนายโครงสร้างนิ่ง — รูปทรงที่โปรตีนลงเอย — ไม่ใช่เส้นทางหรือจังหวะเวลาที่ยาวไปถึงตรงนั้น มันยังทำได้ไม่ดีกับโปรตีนที่มีรูปทรงใช้งานได้หลายแบบ กับผลของการกลายพันธุ์และคู่จับที่มีต่อโครงสร้าง และกับราวหนึ่งในสามของโปรตีนมนุษย์ที่เป็นบริเวณไร้ระเบียบโดยธรรมชาติ ไม่เคยลงเอยเป็นรูปทรงตายตัวรูปเดียวเลย (PubMed, "AlphaFold and Protein Folding: Not Dead Yet! The Frontier Is Conformational Ensembles") ปฏิทรรศน์ของ Levinthal — การพับเกิดขึ้นเร็วขนาดนั้นได้อย่างไร — ยังไม่ถูกแก้จริงๆ AlphaFold ตอบคำถามที่แคบกว่าและมีค่าสูง (ลำดับนี้ให้รูปทรงอะไร) ด้วยการไม่ตอบคำถามเชิงพลวัตนั้นเลย

E N

The move: stop simulating how the protein gets there; read the shape out of a hundred million years of relatives.

T H

ท่า: เลิกจำลองว่าโปรตีนไปถึงตรงนั้นได้อย่างไร แล้วอ่านรูปทรงออกมาจากญาติร้อยล้านปีของมัน

The Move

ทำประจําบท

EN

Move 08 — Find the load-bearing constraint: the one condition that, honored or broken, decides everything.

TH

Move 08 — หาข้อจำกัดที่รับน้ำหนัก — เงื่อนไขเดียวที่ทำตามหรือหักแล้วชี้ขาดทุกอย่าง

EN

Every problem this large contains hundreds of true facts and dozens of plausible-looking obstacles. Almost none of them matter equally. Each solver in this gallery found the single condition that the rest of the problem actually hinged on, and treated everything else as background.

TH

แทบไม่มีข้อเท็จจริงข้อไหนในปัญหาขนาดนี้สำคัญเท่ากันเลย ปัญหาแต่ละข้อบรรจุข้อเท็จจริงที่ถูกต้องอยู่เป็นร้อย และอุปสรรคที่ดูน่าเชื่อถืออีกเป็นสิบ นักแก้ปัญหาล่ะแต่ละคนในตอนนี้หาเงื่อนไขเดียวที่ปัญหาที่เหลือแขวนอยู่กับมันจริงๆ เจอ แล้วปฏิบัติกับทุกอย่างที่เหลือเป็นฉากหลัง

EN

- **Ask which fact, if it changed, would collapse the whole difficulty.** Enigma's daily-key search is astronomically large — until a doubled indicator, or a letter that can never encipher to itself, turns it into an afternoon's logical deduction. Look for the one rule the system cannot break.
- **Separate the dramatic obstacle from the deciding one.** Apollo 13's crisis produced a dozen simultaneous emergencies; the CO2 problem was solved by fixing the parts list, not the chemistry. Challenger's failure had a hundred contributing organizational causes; the physical cause was one seal's behavior below one temperature.
- **When an old attack keeps failing, ask whether you are fighting the real constraint or a decoy.** Three and a half centuries of failed attempts hit Fermat's equation directly. Frey, Serre, and Ribet found the actual load-

bearing structure one layer beneath it — modularity — and Wiles spent his effort there instead.

- **A found constraint survives its own repair.** Wiles's 1993 gap did not mean the reframe was wrong; it meant one plank of the bridge needed replacing. Distinguish a flaw in your central insight from a flaw in a technical tool serving it — the first sends you back to the old problem, the second just costs a year.

TH

- **ถามว่าข้อเท็จจริงข้อไหนที่ถ้าเปลี่ยนไป จะทำให้ความยากทั้งก้อนยุบลง** การค้นหากฎเฉพาะประจำวันของ Enigma ใหญ่ระดับดาราศาสตร์ — จนกระทั่งตัวบ่งชี้ซ้ำสอง หรือตัวอักษรที่ไม่มีวันเข้ารหัสเป็นตัวเอง เปลี่ยนมันให้เป็นการอนุมานเชิงตรรกะป้ายเดียว มองหากฎข้อเดียวที่ระบบนั้นละเมิดไม่ได้
- **แยกอุปสรรคที่คร้ามออกมาจากอุปสรรคที่ชี้ขาด** วิกฤต Apollo 13 ผลิตเหตุฉุกเฉินพร้อมกันเป็นโหล แต่ปัญหา CO2 ถูกแก้ด้วยการตรึงรายการชิ้นส่วน ไม่ใช่ด้วยเคมี ความล้มเหลวของ Challenger มีสาเหตุเชิงองค์การที่ส่งผลร่วมอยู่เป็นร้อย แต่สาเหตุทางกายภาพคือพฤติกรรมของซีลชิ้นเดียวที่ต่ำกว่าอุณหภูมิค่าเดียว
- **เมื่อการบุกแบบเก่าล้มเหลวซ้ำแล้วซ้ำเล่า ให้ถามว่ากำลังสู้กับข้อจำกัดจริง หรือสู้กับตัวล่อ** ความพยายามที่ล้มเหลวตลอดสามศตวรรษครึ่งพุ่งชนสมการของ Fermat ตรงๆ Frey, Serre และ Ribet หาโครงสร้างที่รับน้ำหนักจริงเจอในชั้นที่ลึกลงไปหนึ่งชั้น — มอดูลาริตี — แล้ว Wiles ก็ทุ่มแรงลงตรงนั้นแทน
- **ข้อจำกัดที่หาเจอแล้วย่อมรอดจากการซ่อมของตัวเอง** ช่องโหว่ปี 1993 ของ Wiles ไม่ได้แปลว่าการเปลี่ยนกรอบผิด มันแปลว่าไม้แผ่นหนึ่งของสะพานต้องเปลี่ยน ให้แยกความบกพร่องในการหยั่งเห็น (insight) แกนกลาง ออกจากความบกพร่องในเครื่องมือทางเทคนิคที่รับใช้มัน — อย่างแรกส่งกลับไปยังปัญหาเดิมอย่างหลังแค่เสียเวลาไปหนึ่งปี

EN

Next — Chapter 09 · *The Aesthetics of Solution: why the short path feels the way it does — beauty as the signal of compression, argued through Erdős's Book, Kolmogorov complexity, and the neuroscience that can now watch an insight arrive.*

TH

ต่อไป — บทที่ 09 · สุนทรียะของคำตอบ: ทำไมเส้นทางสั้นถึงให้ความรู้สึกแบบที่มันให้ — ความงามในฐานะสัญญาณของการบีบอัด (compression) เดินข้อโต้แย้งผ่าน คัมภีร์ (The Book) ของ Erdős ความซับซ้อนโคลโมโกรอฟ และประสาทวิทยาศาสตร์ที่วันนี้เฝ้าดูการมาถึงของ insight ได้แล้ว

CHAPTER 09

The Aesthetics of Solution

สุนทรียะของคำตอบ

Euclid's Proof, Twice Told

บทพิสูจน์ของ Euclid เล่าสองรอบ

EN

Take the primes: 2, 3, 5, 7, 11, and so on, thinning as the numbers grow. Do they ever run out? Around 300 BC, Euclid answered no, and the proof takes about four lines. Suppose, he said, that they *do* run out — suppose there is a largest prime, and a complete finite list of every prime that exists: $p_1, p_2, \dots, p_n$. Multiply them all together and add one. The resulting number is not divisible by p_1 (it leaves remainder 1), nor by p_2 , nor by any prime on the list — every division leaves a remainder of exactly one. So this new number is either itself prime, or divisible by some prime not on the list. Either way, the list was not complete. Contradiction. The primes cannot run out.

TH

จำนวนเฉพาะไม่มีวันหมด Euclid พิสูจน์ไว้ราว 300 ปีก่อนคริสตกาล และบทพิสูจน์ยาวราวสี่บรรทัด ตัวจำนวนเฉพาะเอง — 2, 3, 5, 7, 11 ไปต่อไปเรื่อยๆ — บางลงเมื่อตัวเลขใหญ่ขึ้น จึงชวนให้ถามว่ามันจะหมดไหม Euclid ตอบว่าไม่ วิธีของเขาคือสมมติว่ามันหมดจริง — มีจำนวนเฉพาะที่ใหญ่ที่สุด และมีรายการครบถ้วนของจำนวนเฉพาะทุกตัวที่มีอยู่: $p_1, p_2, \dots, p_n$ คุณทั้งหมดเข้าด้วยกัน แล้วบวกหนึ่ง จำนวนที่ได้หารด้วย p_1 ไม่ลงตัว (เหลือเศษ 1) หารด้วย p_2 ก็ไม่ลงตัว หารด้วยจำนวนเฉพาะตัวไหนในรายการก็ไม่ลงตัว ทุกการหารเหลือเศษหนึ่งพอดี จำนวนใหม่นี้จึงต้องเป็นจำนวนเฉพาะเสียเอง หรือไม่ก็หารลงตัวด้วยจำนวนเฉพาะที่ไม่อยู่ในรายการ ทางไหนก็ตาม รายการนั้นไม่ครบ ขัดแย้ง จำนวนเฉพาะหมดไม่ได้

EN

Nothing about that argument required computing a single actual prime beyond the ones you already knew. It required no machinery — no calculus, no advanced notation, nothing invented after Euclid's own era. And yet it settles a question about *infinity*, using only the arithmetic a bright twelve-year-old already owns. G. H. Hardy, the Cambridge number theorist, held this proof up as close to the platonic case of what a mathematician means by beauty, and in his 1940 book *A Mathematician's Apology* he tried to say exactly what that beauty consisted of. A beautiful proof, Hardy argued, combines "a very high degree of unexpectedness, combined with inevitability and economy" — the same two examples, Euclid's primes and the irrationality of the square root of 2, that Hardy himself chose to illustrate the point (Hardy, *A Mathematician's Apology*, 1940, quoted in "A Mathematician's Apology," *Plus Magazine*, University of Cambridge, 2016). Unexpectedness: the result must surprise — a proof that plods obviously toward its ending produces no aesthetic charge at all. Inevitability: once seen, the argument must feel as though it could not have gone any other way; the surprise and the necessity, oddly, do not cancel each other, they compound. Economy: nothing in the proof is wasted. Take away Euclid's "add one" and the whole argument collapses; the proof is exactly as long as the idea, no longer.

TH

ข้อโต้แย้งนั้นไม่ต้องคำนวณจำนวนเฉพาะจริงเพิ่มแม้แต่ตัวเดียวจากที่รู้อยู่แล้ว ไม่ต้องใช้เครื่องมือใดเลย — ไม่มีแคลคูลัส ไม่มีสัญกรณ์ชั้นสูง ไม่มีอะไรที่คิดขึ้นหลังยุคของ Euclid เอง แต่มันปิดคำถามเรื่องอนันต์ได้ ด้วยเลขคณิตที่เด็กสิบสองขวบหัวดีคนหนึ่งมีอยู่แล้ว G. H. Hardy นักทฤษฎีจำนวนแห่ง Cambridge ยกบทพิสูจน์นี้เป็นตัวอย่างที่ใกล้เคียงในอุดมคติที่สุดของสิ่งที่นักคณิตศาสตร์เรียกว่าความงาม และในหนังสือปี 1940 ของเขา *A Mathematician's Apology* เขาพยายามระบุให้ชัดว่าความงามนั้นประกอบด้วยอะไร คำตอบของ Hardy: บทพิสูจน์ที่งามผสม "ความคาดไม่ถึงในระดับสูงมาก เข้ากับความจำเป็นที่หลีกเลี่ยงไม่ได้และความประหยัด" ("a very high degree of unexpectedness, combined with inevitability and economy") และสองตัวอย่างที่ Hardy เลือกมาอธิบายประเด็นนี้ ก็คือจำนวนเฉพาะของ Euclid กับความเป็นจำนวนอตรรกยะของรากที่สองของ 2 (Hardy, *A Mathematician's Apology*, 1940, quoted in "A Mathematician's Apology,"

Plus Magazine, University of Cambridge, 2016) ความคาดไม่ถึง: ผลลัพธ์ต้องทำให้ประหลาดใจ บทพิสูจน์ที่ย้ำไปสู่ตอนจบอย่างเห็นชัดตั้งแต่ต้น ไม่ให้ประจุทางสุนทรียะใดๆ เลย ความจำเป็นที่หลีกเลี่ยงไม่ได้: พอเห็นแล้ว ข้อโต้แย้งต้องรู้สึกราวกับไปทางอื่นไม่ได้ ความประหลาดใจกับความจำเป็นไม่หักล้างกัน — แปลกดี — แต่ทบทวน ความประหลาด: ไม่มีอะไรในบทพิสูจน์เสียเปล่า เอา "บทหนึ่ง" ของ Euclid ออก ข้อโต้แย้งทั้งก่อนก็พัง บทพิสูจน์ยาวเท่ากับไอเดียพอดี ไม่ยาวกว่านั้น

EN

Notice what these three words are doing together. Unexpectedness says the proof compresses more surprise per line than you had any right to expect. Economy says it does this using the fewest possible lines. Inevitability says the compression, once achieved, feels not clever but *correct* — as if the four lines were always the true size of the fact, and everything before Euclid had simply been carrying it in an oversized box. This chapter's claim, argued through mathematics, psychology, and neuroscience, is that this feeling is not decoration on top of understanding. It is a real signal, generated by a real mechanism, that fires when a problem's representation has just gotten dramatically shorter — which means the felt experience of beauty in a solution is, in a precise sense this chapter will make literal, the shortest path arriving and being *recognized* as such from the inside.

TH

สามคำนี้ทำงานเป็นชุดเดียว ความคาดไม่ถึงบอกว่าบทพิสูจน์บีบความประหลาดใจต่อบรรทัดได้มากกว่าที่ใครมีสิทธิ์คาดหวัง ความประหลาดบอกว่ามันทำสิ่งนี้ด้วยจำนวนบรรทัดน้อยที่สุดเท่าที่เป็นไปได้ ความจำเป็นที่หลีกเลี่ยงไม่ได้บอกว่าการบีบอัด (compression) นั้น เมื่อสำเร็จแล้ว ไม่ให้ความรู้สึกว่าฉลาด แต่ให้ความรู้สึกว่าคุณต้อง — ราวกับสี่บรรทัดนั้นคือขนาดจริงของข้อเท็จจริงมาตลอด และทุกคนก่อน Euclid แคแบกมันไว้ในกล่องที่ใหญ่เกินตัว ข้อเสนอของบทนี้ตรงไปตรงมา: ความรู้สึกนี้ไม่ใช่เครื่องประดับที่วางทับความเข้าใจ มันคือสัญญาณจริง จากกลไกจริง ที่จุดขึ้นเมื่อรูปแบบการแทนปัญหา (representation) เพิ่งสั้นลงอย่างฮวบฮาบ บทนี้จะเถียงข้อเสนอนั้นผ่านคณิตศาสตร์ จิตวิทยา และประสาทวิทยา ผลที่ตามมาคือประสบการณ์ของความงามในคำตอบ — ในความหมายที่แม่นยำที่บทนี้จะทำให้

เป็นจริงตามตัวอักษร — ก็คือเส้นทางที่สั้นที่สุดกำลังมาถึง และถูกรับรู้ว่าเป็นเช่นนั้นจากข้างใน

The Book

คัมภีร์ (The Book)

EN

Paul Erdős, the itinerant Hungarian number theorist who published more papers than any mathematician in history and owned almost nothing, had a private name for this feeling. He called it "the Book" — an imaginary volume, kept by God, containing the single most elegant proof of every true theorem, however many inferior proofs mathematicians had settled for along the way. Erdős was an avowed atheist; the joke, such as it was, cut in both directions. In a 1985 lecture, Erdős put it to his audience directly: "You don't have to believe in God, but you should believe in The Book" — an account preserved by his collaborator Joel Spencer in the AMS memorial notices (Spencer, in "Paul Erdős (1913–1996)," *Notices of the AMS*, 1998). His highest compliment for a piece of mathematics was to say it was "straight from the Book."

TH

Paul Erdős มีชื่อเรียกส่วนตัวสำหรับความรู้สึกนี้ นักทฤษฎีจำนวนชาวฮังการีผู้ร่อนเร่คนนี้ตีพิมพ์บทความมากกว่านักคณิตศาสตร์คนใดในประวัติศาสตร์ และแทบไม่มีข้าวของเป็นของตัวเอง เขาเรียกมันว่า "คัมภีร์" (the Book) — หนังสือในจินตนาการที่พระเจ้าเก็บรักษาไว้ บรรจุบทพิสูจน์ที่สง่างามที่สุดเพียงหนึ่งเดียวของทุกทฤษฎีบทที่เป็นจริง ไม่ว่านักคณิตศาสตร์จะยอมใช้บทพิสูจน์ที่ด้อยกว่ากี่ฉบับระหว่างทางก็ตาม Erdős ประกาศตัวเป็นผู้ไม่เชื่อในพระเจ้า มุกนี้ — ถ้าจะเรียกว่ามุก — จึงคมทั้งสองทาง ในปาฐกถาปี 1985 Erdős บอกผู้ฟังตรงๆ ว่า "คุณไม่จำเป็นต้องเชื่อในพระเจ้า แต่คุณควรเชื่อใน The Book" ("You don't have to believe in God, but you should believe in The Book") Joel Spencer ผู้ร่วมงานของเขาเก็บคำบอกเล่านี้ไว้ในบันทึกอาลัยของ AMS (Spencer, ใน "Paul Erdős (1913–1996)," *Notices of the AMS*, 1998) คำชมสูงสุดของเขาต่องานคณิตศาสตร์ชิ้นหนึ่ง คือบอกว่ามัน "มาจาก The Book ตรงๆ"

E N

In 1994, two years before Erdős's death, the mathematicians Martin Aigner and Günter Ziegler began collecting candidate entries — with Erdős's own enthusiastic help — for an earthly attempt at the volume. The result, *Proofs from THE BOOK*, appeared in 1998, too late for Erdős to hold a copy (Wikipedia, "Proofs from THE BOOK"). What is striking about the collection is not that it contains hard mathematics — every proof in every mathematics library is hard mathematics — but that it contains, again and again, *multiple* proofs of the same theorem, filed side by side so the reader can watch the difference between "a proof" and "the Book proof" of the identical fact. The infinitude of primes gets four separate treatments in the opening chapter, Euclid's among them. The point is not that the other three are wrong. It is that some correct arguments compress the truth further than others, and a trained eye can rank them even when every one of them is, technically, a complete and valid proof.

T H

Proofs from THE BOOK ออกมาในปี 1998 ช้าเกินกว่าที่ Erdős จะได้ถือมันไว้ในมือนักคณิตศาสตร์ Martin Aigner กับ Günter Ziegler เริ่มรวบรวมรายการที่เข้าข่ายตั้งแต่ปี 1994 สองปีก่อน Erdős เสียชีวิต — โดยได้รับความช่วยเหลืออย่างกระตือรือร้นจาก Erdős เอง — เพื่อทำเล่มนั้นให้เกิดขึ้นจริงบนโลก (Wikipedia, "Proofs from THE BOOK") สิ่งที่น่าสนใจในเล่มนี้ไม่ใช่ว่ามันมีคณิตศาสตร์ยากๆ — บทพิสูจน์ทุกฉบับในห้องสมุดคณิตศาสตร์ทุกแห่งก็คือคณิตศาสตร์ยากๆ ทั้งนั้น — แต่คือมันวางบทพิสูจน์ หลายฉบับ ของทฤษฎีบทเดียวกันเรียงเคียงกัน ครึ่งแล้วครึ่งเล่า ให้ผู้อ่านดูความต่างระหว่าง "บทพิสูจน์ฉบับหนึ่ง" กับ "บทพิสูจน์ฉบับ The Book" ของข้อเท็จจริงเดียวกัน บทเปิดพิสูจน์ความไม่รู้จบของจำนวนเฉพาะไว้อย่างแยบยล และของ Euclid เป็นหนึ่งในนั้น อีกสามแบบไม่ได้ผิด ประเด็นคือข้อโต้แย้งที่ถูกต้องบางอันบีบความจริงได้แน่นกว่าอันอื่น และตาที่ฝึกมาแล้วจัดอันดับพวกมันได้ แม้ทุกอันจะเป็นบทพิสูจน์ที่สมบูรณ์และใช้ได้จริงในทางเทคนิค

E N

At the extreme end of this compression sits an entire genre: the **proof without words**, a diagram that establishes a mathematical fact with no accompanying argument at all, because the picture is the argument (Wikipedia,

dia, "Proof without words"). The oldest and most reproduced example, a dissection proof of the Pythagorean theorem traceable to the ancient Chinese text *Zhou Bi Suan Jing*, needs no symbols and no sentence: a square is cut and rearranged, and the eye supplies the proof that formal notation would otherwise spend a paragraph laboring to state. A proof without words is Hardy's economy pushed to its logical limit — the shortest possible path is one you do not even have to walk, only glance at.

TH

ปลายสุดของการบีบอัดนี้มีแนวทางหนึ่งตั้งอยู่ทั้งแนว: **บทพิสูจน์ที่ไม่ใช้คำ** (proof without words) แผนภาพที่สถาปนาข้อเท็จจริงทางคณิตศาสตร์โดยไม่มีข้อโต้แย้งประกอบเลยแม้แต่หน่อย เพราะภาพคือข้อโต้แย้ง (Wikipedia, "Proof without words") ตัวอย่างที่เก่าที่สุดและถูกพิมพ์ซ้ำมากที่สุดคือบทพิสูจน์ทฤษฎีบทพีทาโกรัสด้วยการตัดแบ่ง สวกลับไปถึงตำราจีนโบราณ *Zhou Bi Suan Jing* ได้ มันไม่ต้องใช้สัญลักษณ์ ไม่ต้องใช้ประโยคสักประโยค: สีเหลี่ยมจัตุรัสถูกตัดและจัดเรียงใหม่ แล้วสายตาก็ให้บทพิสูจน์ที่สัญกรณ์รูปนัยต้องใช้ทั้งย่อหน้าเพื่อกล่าวออกมา นี่คือความประหยัดของ Hardy ที่ถูกดันไปถึงขีดจำกัดทางตรรกะ — เส้นทางที่สั้นที่สุดเท่าที่เป็นไปได้คือเส้นทางที่ไม่ต้องเดินด้วยซ้ำ แค่ชำเลื่องมองก็พอ

Occam's Razor, Rebuilt in Bits

มีดโกนอ็อกคัม สร้างใหม่ด้วยบิต

EN

Long before information theory existed, medieval logicians had already guessed that shortness tracks truth. William of Ockham's razor — prefer the explanation that requires the fewest new assumptions — is usually stated as a rule of thumb about parsimony. In the twentieth century it acquired a mathematical skeleton. Andrey Kolmogorov and, independently, Ray Solomonoff and Gregory Chaitin, defined the **Kolmogorov complexity** of a string of data as the length of the shortest computer program that outputs it and then halts. A string with real structure — the digits of pi, a repeating pattern, a grammatical sentence — can be produced by a short program, because the structure itself is a compression instruction. A string of pure noise cannot be shortened at all; the shortest program that reproduces it is,

essentially, the string itself, verbatim. Complexity, in this technical sense, is incompressibility, and simplicity is the opposite: the gap between a thing's raw size and the size of its shortest true description (overview at Wikipedia, "Kolmogorov complexity").

TH

ความสั้นเดินคู่กับความจริง — นักตรรกวิทยาในยุคกลางเดาข้อนี้ไว้นานก่อนที่ทฤษฎีสารสนเทศจะมีอยู่ มีดโกนอีกคัมของ William of Ockham — เลือกคำอธิบายที่ต้องการสมมติฐานใหม่น้อยที่สุด — มักถูกเล่าเป็นกฎสามัญสำนึกว่าด้วยความมัธยัสถ์ในศตวรรษที่ยี่สิบ มันได้โครงกระดูกทางคณิตศาสตร์ Andrey Kolmogorov และ Ray Solomonoff กับ Gregory Chaitin ซึ่งทำงานอย่างเป็นอิสระต่อกัน นิยาม **ความซับซ้อนโคลโมโกรอฟ** ของสายข้อมูลหนึ่งว่าคือความยาวของโปรแกรมคอมพิวเตอร์ที่สั้นที่สุดที่พิมพ์มันออกมาแล้วหยุด สายที่มีโครงสร้างจริง — เลขหลักของพาย ลวดลายที่ซ้ำ ประโยคที่ถูกไวยากรณ์ — ผลิตได้ด้วยโปรแกรมสั้นๆ เพราะโครงสร้างนั่นเองคือคำสั่งบีบอัด สายที่เป็นสัญญาณรบกวนล้วนๆ ทำให้สั้นลงไม่ได้เลย โปรแกรมที่สั้นที่สุดที่สร้างมันขึ้นมาใหม่ได้ก็คือตัวสายนั่นเอง คำต่อคำ ความซับซ้อนในความหมายทางเทคนิคนี้จึงคือความบีบไม่ลง และความเรียบง่ายคือด้านตรงข้าม: ช่องว่างระหว่างขนาดดิบของสิ่งหนึ่ง กับขนาดของคำอธิบายที่จริงและสั้นที่สุดของมัน (ภาพรวมที่ Wikipedia, "Kolmogorov complexity")

EN

Because Kolmogorov complexity is famously uncomputable in general — there is no algorithm that can certify a given program is truly the shortest one possible — Jorma Rissanen proposed in 1978 a practical cousin: the **Minimum Description Length** principle, which picks, among the models actually available to you, the one that lets you encode both the model and the data it explains in the fewest total bits (Wikipedia, "Minimum description length"; Grünwald, "A Tutorial Introduction to the Minimum Description Length Principle," 2004). MDL is Occam's razor with a unit attached: an explanation is better not merely because it is "simpler" in some intuitive sense, but because it demonstrably takes fewer bits to state, and a briefer statement that still accounts for everything observed is, formally, a superior compression of the same truth. This is the machinery that turns Hardy's word "economy" from a literary compliment into a quantity that

could, in principle, be measured — and it hands this chapter its central claim in blunt form: an elegant proof and a short program are the same kind of object, judged by the same criterion, in two different notations.

TH

ความซับซ้อนโคลโมโกรอฟขึ้นชื่อว่าคำนวณไม่ได้ในกรณีทั่วไป — ไม่มีอัลกอริทึมใดรับรองได้ว่าโปรแกรมที่ให้นั้นสั้นที่สุดเท่าที่เป็นไปได้จริง — Jorma Rissanen จึงเสนอญาติที่ใช้งานได้จริงไว้ในปี 1978: หลักการ **ความยาวคำอธิบายน้อยที่สุด (MDL)** มันเลือกจากบรรดาแบบจำลองที่คุณมีให้ใช้จริง ตัวที่เข้ารหัสทั้งแบบจำลองเองและข้อมูลที่มันอธิบายด้วยจำนวนบิตรวมน้อยที่สุด (Wikipedia, "Minimum description length"; Grünwald, "A Tutorial Introduction to the Minimum Description Length Principle," 2004) MDL คือมีดโกนอีกคมที่ติดหน่วยวัดมาด้วย: คำอธิบายหนึ่งดีกว่า ไม่ใช่เพียงเพราะมัน "เรียบง่ายกว่า" ในความรู้สึกกลางๆ แต่เพราะแสดงให้เห็นได้ว่ามันใช้บิตน้อยกว่าในการกล่าวออกมา และข้อความที่สั้นกว่าที่ยังอธิบายทุกอย่างที่สังเกตได้ครบ ก็คือการบีบอัดความจริงชุดเดียวกันที่เหนือกว่าในเชิงรูปนัย นี่คือการโกนที่เปลี่ยนคำว่า "ความประหยัด" ของ Hardy จากคำชมเชิงวรรณกรรมเป็นปริมาณที่โดยหลักการแล้ววัดได้ และมันยื่นข้อเสนอหลักของบทนี้ออกมาที่พอๆ บทพิสูจน์ที่สง่างามกับโปรแกรมสั้นๆ คือวัฏخذชุดเดียวกัน ตัดสินด้วยเกณฑ์เดียวกัน ในสัญกรณ์สองแบบ

The Compression Drive

แรงขับให้บีบอัด

EN

If shortness of description is what mathematicians are responding to when they call a proof beautiful, the natural next question is why the human nervous system would bother tracking such a thing at all. The computer scientist Jürgen Schmidhuber has spent much of his career arguing that it does so constantly, and that beauty is only one of several everyday experiences built on the same underlying signal. In a series of papers beginning in the mid-2000s, Schmidhuber proposed that an observer with limited computational resources is, at every moment, running an internal compressor on its sensory stream — trying to predict what comes next using as little

internal machinery as possible — and that subjective **beauty** is the experience of that compressor suddenly getting *better*: a pattern the observer had not yet noticed collapses, for the first time, into a shorter internal description (Schmidhuber, "Simple Algorithmic Principles of Discovery, Subjective Beauty, Selective Attention, Curiosity and Creativity," 2007; Schmidhuber, "Driven by Compression Progress," 2008).

TH

ระบบประสาทของมนุษย์ติดตามความสั้นของคำอธิบายอยู่ตลอดเวลา — นั่นคือข้อเสนอที่นักวิทยาการคอมพิวเตอร์ Jürgen Schmidhuber ใช้เวลาส่วนใหญ่ในอาชีพเถียง คำถามที่นำมาถึงตรงนี้ตรงไปตรงมา: ถ้าความสั้นของคำอธิบายคือสิ่งที่นักคณิตศาสตร์ตอบสนองเวลาเรียกบทพิสูจน์ว่างาม ทำไมระบบประสาทถึงต้องเสียแรงติดตามสิ่งนี้ตั้งแต่แรก Schmidhuber ตอบว่าความงามเป็นเพียงหนึ่งในประสบการณ์ประจำวันหลายอย่างที่สร้างขึ้นบนสัญญาณพื้นฐานอันเดียวกัน ในบทความชุดหนึ่งที่เริ่มกลางทศวรรษ 2000 เขาเสนอว่าผู้สังเกตที่มีทรัพยากรการคำนวณจำกัดกำลังรันเครื่องบีบอัดภายในกับกระแสประสาทสัมผัสของตัวเองอยู่ทุกขณะ — พยายามทำนายว่าจะไรจะมาต่อโดยใช้กลไกภายในให้น้อยที่สุด — และความงามเชิงอัตวิสัยคือประสบการณ์ที่เครื่องบีบอัดนั้นจู่ๆ ก็ดีขึ้น: ลวดลายที่ผู้สังเกตยังไม่เคยสังเกตเห็นยุบตัวลงเป็นคำอธิบายภายในที่สั้นกว่าเป็นครั้งแรก (Schmidhuber, "Simple Algorithmic Principles of Discovery, Subjective Beauty, Selective Attention, Curiosity and Creativity," 2007; Schmidhuber, "Driven by Compression Progress," 2008)

EN

On this account, **curiosity** is the drive to seek out exactly the data most likely to produce further compression progress — not data that is already understood, and not data so chaotic it will never compress, but data sitting right at the frontier of the observer's current model. **Creativity** is the active production of new data — a proof, an artwork, a joke — engineered specifically to hand another mind that same jolt of compression. Schmidhuber's theory is deliberately unified: it is offered as one mechanism explaining why infants stare longer at moderately surprising events than at fully predictable or fully random ones, why a punchline lands only once the setup's redundant framing snaps into a shorter reading, and why a mathematician's pulse

can visibly quicken over a four-line proof of an old theorem. The theory is a hypothesis, not a settled consensus, and it has not displaced older, purely social or purely sensory accounts of aesthetic pleasure. But it makes a sharp, falsifiable prediction that fits this book's spine exactly: what feels beautiful is what got *shorter*, not what is objectively simple in some absolute sense — the beauty is in the drop, not the destination.

TH

ตามคำอธิบายนี้ **ความอยากรู้** คือแรงขับให้ออกไปหาข้อมูลที่มีโอกาสสูงสุดที่จะทำให้การบีบอัดคืบหน้าต่อ — ไม่ใช่ข้อมูลที่เราเข้าใจไปแล้ว ไม่ใช่ข้อมูลที่วุ่นวายจนไม่มีวันบีบลงได้ แต่คือข้อมูลที่อยู่ตรงแนวหน้าของแบบจำลองปัจจุบันของผู้สังเกตพอดี **การสร้างสรรค** คือการผลิตข้อมูลใหม่ขึ้นมาอย่างจงใจ — บทพิสูจน์ งานศิลปะ มุกตลก — ที่ออกแบบมาเพื่อยืนแรงกระตุกจากการบีบอัดแบบเดียวกันนั้นให้อีกจิตหนึ่งโดยเฉพาะ ทฤษฎีของ Schmidhuber ตั้งใจให้รวมเป็นหนึ่ง: กลไกเดียวอธิบายว่าทำไมทารกจึงจ้องเหตุการณ์ที่น่าประหลาดใจปานกลางนานกว่าเหตุการณ์ที่หายได้หมดหรือสุมไปทั้งหมด ทำไมมุกจบถึงตลกก็ต่อเมื่อกรอบส่วนเกินของท่อนตั้งติดเข้าที่เป็นการอ่านที่สั้นกว่า และทำไมซีพอร์ของนักคณิตศาสตร์เด่นเร็วขึ้นจนเห็นได้เมื่อเจอบทพิสูจน์ที่บรรทัดของทฤษฎีบทเก่า ทฤษฎีนี้เป็นสมมติฐาน ไม่ใช่ฉันทามติที่ลงตัวแล้ว และมันยังไม่ได้ล้มคำอธิบายเก่าที่มองความสุขทางสุนทรียะเป็นเรื่องทางสังคมล้วนหรือทางประสาทสัมผัสล้วน แต่มันให้คำทำนายที่คมและพิสูจน์ผิดได้ และเข้ากับแกนของหนังสือเล่มนี้พอดี: สิ่งที่รู้สึกว่างามคือสิ่งที่สั้นลง ไม่ใช่สิ่งที่เรียบง่ายขึ้นอย่างเป็นภาววิสัยในความหมายสมบูรณ์ใดๆ ความงามอยู่ที่การหล่นลง ไม่ใช่ที่ปลายทาง

Chess Masters Don't See Pieces

ปรมาจารย์หมากรุกไม่เห็นตัวหมากรุก

EN

Compression is not only something that happens to a proof on paper. It is something a trained mind builds inside itself, and one of the cleanest demonstrations of that comes from a chessboard rather than a blackboard.

TH

การบีบอัดเกิดในหัวคนด้วย ไม่ใช่แค่กับบทพิสูจน์บนกระดาน จิตที่ผ่านการฝึกสร้างมันขึ้นภายในตัวเอง และการสาธิตที่สะอาดที่สุดอันหนึ่งของเรื่องนี้มาจากกระดานหมากรุก ไม่ใช่กระดานดำ

EN

In the 1940s, the Dutch psychologist and chess master Adriaan de Groot showed strong players a real mid-game position for a few seconds and then asked them to reconstruct it from memory. Grandmasters did this almost perfectly; club players did much worse — the standard, unsurprising story of expertise. But de Groot's next step was the important one: when the position on the board was not a plausible game at all, but pieces scattered at random, the grandmasters' advantage nearly vanished. They reconstructed random boards barely better than novices did (summarized in Wikipedia, "Adriaan de Groot"). Whatever the masters had that the amateurs lacked, it was not a bigger memory for squares and pieces in general. It was something that only worked on *real* chess.

TH

แกรนด์มาสเตอร์จำตำแหน่งหมากรุกได้เกือบสมบูรณ์แบบ — แต่เฉพาะเมื่อตำแหน่งนั้นมาจากเกมจริง ในทศวรรษ 1940 Adriaan de Groot นักจิตวิทยาและปรมาจารย์หมากรุกชาวดัตช์ให้ผู้เล่นฝีมือดีดูตำแหน่งกลางเกมจริงไม่กี่วินาที แล้วขอให้สร้างมันขึ้นใหม่จากความจำ แกรนด์มาสเตอร์ทำได้เกือบสมบูรณ์แบบ ผู้เล่นระดับสโมสรทำได้แย่มาก — เรื่องเล่ามาตรฐานเรื่องความเชี่ยวชาญไม่มีอะไรน่าแปลกใจ ก้าวถัดไปของ de Groot ต่างหากที่สำคัญ: เมื่อตำแหน่งบนกระดานไม่ใช่เกมที่เป็นไปได้เลย แต่เป็นตัวหมากรุกที่วางกระจายแบบสุ่ม ความได้เปรียบของแกรนด์มาสเตอร์แทบหายไปหมด พวกเขาสร้างกระดานสุ่มขึ้นใหม่ได้ดีกว่ามือใหม่เพียงนิดเดียว (สรุปไว้ใน

Wikipedia, "Adriaan de Groot") สิ่งที่มาจารย์มีและมือสมัครเล่นไม่มี จึงไม่ใช่ความจำที่ใหญ่กว่าสำหรับช่องและตัวหมากโดยทั่วไป มันคือบางอย่างที่ทำงานได้เฉพาะกับหมากกรุกจริงเท่านั้น

EN

William Chase and Herbert Simon — the same Simon whose problem-space formalism runs through this entire book — reran de Groot's experiment with modern controls in 1973 and gave the phenomenon its name: **chunking**. A chunk, operationally, is a cluster of pieces a player's eye groups and recalls as a single unit rather than as separate items, and Chase and Simon estimated that a chess master carries somewhere between fifty and a hundred thousand such chunks in long-term memory, each one a compressed, pre-labeled pattern — "this is a fianchettoed bishop formation," "this is a standard rook-lift" — built from years of exposure to real games (Chase & Simon, "Perception in Chess," *Cognitive Psychology*, 1973). A master does not perceive thirty-two individual pieces on a real board; the master perceives perhaps five or six *chunks*, each one already carrying a compressed history of what tends to happen next. Strip away the real-game statistics — randomize the position — and every one of those chunks becomes useless, because chunks are compressions of *this specific, structured domain*, not of chessboards in the abstract. Mastery, on this evidence, is not superior raw processing. It is a private library of shortest descriptions, built the slow way, that lets an expert's mind do to a familiar problem exactly what Euclid's proof did to the infinitude of primes: replace a long, effortful search with a nearly instantaneous, already-compressed read.

TH

ปรากฏการณ์นี้ได้ชื่อในปี 1973: การรวมเป็นก้อน (**chunking**) William Chase กับ Herbert Simon — Simon คนเดียวกับที่รูปถ่ายด้วยพื้นที่ปัญหา (problem space) ของเขาพาดผ่านหนังสือเล่มนี้ทั้งเล่ม — รันการทดลองของ de Groot เข้าด้วยตัวควบคุมสมัยใหม่ และตั้งชื่อนั้นให้มัน ในทางปฏิบัติ chunk คือกลุ่มตัวหมากที่สายตาผู้เล่นจับรวมและจำเป็นหน่วยเดียว แทนที่จะเป็นชิ้นแยกๆ Chase กับ Simon ประเมินว่าปรมาจารย์หมากกรุกคนหนึ่งพก chunk แบบนี้ไว้ในความจำระยะยาวราวห้าหมื่นถึงหนึ่งแสนก้อน แต่ละก้อนคือลวดลายที่บีบอัดและติดป้ายไว้ล่วงหน้า — "นี่

คือรูปปิชอบแบบ *fianchetto* "นี่คือ rook-lift มาตรฐาน" — สร้างจากการคลุกคลีกับเกมจริงหลายปี (Chase & Simon, "Perception in Chess," *Cognitive Psychology*, 1973) ประมาจารย์ไม่ได้รับรู้ตัวหมากสามสิบสองตัวแยกกันบนกระดานจริง เขารับรู้ *chunk* รวาท้าหรือหกก่อน แต่ละก้อนพ่วงประวัติที่บีบอัดไว้แล้วว่าอะไรมักเกิดขึ้นต่อจากนี้ ลอกสถิติของเกมจริงออก — สุ่มตำแหน่งเสีย — *chunk* ทุกก้อนก็ไร้ประโยชน์ทันที เพราะ *chunk* คือการบีบอัดของโดเมนเฉพาะที่มีโครงสร้างนี้ไม่ใช่ของกระดานหมากรุกในเชิงนามธรรม ความเชี่ยวชาญ ตามหลักฐานชุดนี้ จึงไม่ใช่กำลังประมวลผลดิบที่เหนือกว่า มันคือห้องสมุดส่วนตัวของคำอธิบายที่สั้นที่สุด สร้างขึ้นด้วยวิธีซ้ำๆ และทำให้จิตของผู้เชี่ยวชาญทำกับปัญหาที่คุ้นเคยได้แบบเดียวกับที่บทพิสูจน์ของ Euclid ทำกับความไม่รู้จบของจำนวนเฉพาะเป๊ะๆ: แทนการค้นหา (search) ที่ยาวและเปลืองแรง ด้วยการอ่านที่บีบอัดมาแล้วและเกิดขึ้นแทบทันที

The Aha in the Scanner

จังหวะ "อ๋อ" ในเครื่องสแกน

EN

If beauty is compression progress, and mastery is compressed perception, the next question is whether the moment the compression actually *happens* — the instant a representation snaps into its shorter form — leaves any trace a machine can detect. In 2004, the cognitive neuroscientists Mark Jung-Beeman and John Kounios, together with several collaborators, set out to find one.

TH

ขณะที่ representation ดัดเข้าสู่รูปที่สั้นกว่า มันทิ้งร่องรอยให้เครื่องจักรตรวจจับได้หรือเปล่า — นี่คือการถามถัดไป ถ้าความงามคือความคืบหน้าของการบีบอัด และความเชี่ยวชาญคือการรับรู้ที่ถูกบีบอัดแล้ว ช่วงเวลาที่มีการบีบอัดเกิดขึ้นจริงๆ ก็ควรมีร่องรอย ปี 2004 นักประสาทวิทยาศาสตร์เชิงปัญญา Mark Jung-Beeman กับ John Kounios พร้อมผู้ร่วมงานอีกหลายคน ออกไปตามหาร่องรอยนั้น

EN

Their subjects solved compound remote-associates puzzles — three words (say, *pine*, *crab*, *sauce*) that share a fourth, connecting word (*apple*) — while

wired to either an EEG cap or an fMRI scanner. After each solution, subjects reported whether the answer had arrived analytically, step by step, or as a sudden insight — an "aha." The fMRI data showed a burst of activity, specific to insight solutions, in the right hemisphere's anterior superior temporal gyrus, a region already known for making distant, non-obvious connections across concepts — the same territory that lights up when someone parses a metaphor or gets a joke's twist. The EEG data, with its far finer time resolution, caught something the scanner could not: a sudden spike of high-frequency gamma-band activity in that same right-hemisphere region, beginning roughly 0.3 seconds *before* subjects consciously registered the answer — and, in a further wrinkle, roughly 1.5 seconds before that gamma burst, a jump in alpha-wave activity over the visual cortex, as though the brain briefly dimmed its own incoming visual feed to let the insight surface undisturbed (Jung-Beeman, Bowden, Haberman, Frymiare, Arambel-Liu, Greenblatt, Reber & Kounios, "Neural Activity When People Solve Verbal Problems with Insight," *PLoS Biology*, 2004).

TH

สมองรู้คำตอบก่อนเจ้าของมัน ผู้เข้าร่วมการทดลองของพวกเขาแก้โจทย์ compound remote associates — คำสามคำ (เช่น pine, crab, sauce) ที่มีคำที่สี่ เชื่อมรวมกันอยู่ (apple) — ขณะติดหมวก EEG หรืออยู่ในเครื่องสแกน fMRI หลังแก้ได้แต่ละข้อ ผู้เข้าร่วมรายงานว่าคำตอบมาแบบวิเศษที่ละชั้น หรือมาเป็นการหยั่งเห็น (insight) อย่างฉับพลัน — แบบ "อ้อ" ข้อมูล fMRI แสดงกิจกรรมที่พุ่งขึ้น เฉพาะกับคำตอบแบบ insight ที่ anterior superior temporal gyrus ของสมองซีกขวา บริเวณที่รู้จักอยู่แล้วว่าเชื่อมโยงโน้ตส่นที่อยู่ห่างกันและไม่ชัดเจน — พื้นที่เดียวกับที่สว่างขึ้นเวลาคนตีความอุปลักษณ์หรือจับมุกได้ ข้อมูล EEG มีความละเอียดทางเวลาสูงกว่ามาก จึงจับสิ่งที่เครื่องสแกนจับไม่ได้: คลื่นย่านแกมมาความถี่สูงพุ่งขึ้นฉับพลันในบริเวณสมองซีกขวาเดียวกันนั้น เริ่มราว 0.3 วินาทีก่อนที่ผู้เข้าร่วมจะรู้ว่าได้คำตอบ และในรอยพับอีกชั้นหนึ่ง ราว 1.5 วินาทีก่อนการพุ่งของแกมมานั้น คลื่นอัลฟาเหนือคอร์เทกซ์การมองเห็นกระโดดขึ้น ราวกับสมองหรี่ภาพที่ไหลเข้ามาของตัวเองลงชั่วคราว เพื่อให้ insight โผล่ขึ้นมาโดยไม่ถูกรบกวน (Jung-Beeman, Bowden, Haberman, Frymiare, Arambel-Liu, Greenblatt, Reber & Kounios,

"Neural Activity When People Solve Verbal Problems with Insight," PLoS Biology, 2004)

EN

Kounios and Beeman spent the following decade replicating and extending the finding, and in 2015 published a popular synthesis, *The Eureka Factor*, arguing that this neural signature is a genuine, measurable correlate of the Gestalt insight this book met in chapter 03 — Köhler's chimpanzees, Duncker's candle — now visible on an instrument rather than inferred from behavior alone (Kounios & Beeman, *The Eureka Factor: Aha Moments, Creative Insight, and the Brain*, 2015). Read alongside Hardy, Erdős, and Schmidhuber, the picture completes itself: the felt sense of "aha," the sudden drop in a solution's apparent length, and a specific, localized, time-stamped burst of brain activity all appear to be describing the same event from three different vantage points — the introspective, the informational, and the physiological. The shortest path, when it arrives, is not merely computed. It is *announced*, by a mechanism that evolved to reward the mind for finding one.

TH

Kounios กับ Beeman ใช้ทศวรรษถัดมาทำซ้ำและต่อยอดผลการค้นพบนี้ และในปี 2015 ตีพิมพ์บทสังเคราะห์ฉบับตลาด *The Eureka Factor* ข้อเสนอของเล่มนี้: ลายเซ็นทางประสาทนี้คือสหสัมพันธ์ที่แท้จริงและวัดได้ของ insight แบบเกสตัลท์ (Gestalt) ที่หนังสือเล่มนี้พบในบทที่ 03 — ลิงชิมแปนซีของ Köhler เทียนของ Duncker — ตอนนี้องเห็นได้บนเครื่องมือ ไม่ใช่อนุมานจากพฤติกรรมอย่างเดียว (Kounios & Beeman, *The Eureka Factor: Aha Moments, Creative Insight, and the Brain*, 2015) อ่านคู่กับ Hardy, Erdős และ Schmidhuber ภาพก็เต็มขึ้นมาเอง ความรู้สึก "อ้อ" การหล่นลงฉับพลันของความยาวที่ปรากฏของคำตอบ และการพลุ่งของกิจกรรมสมองที่เฉพาะเจาะจง ระบุตำแหน่งได้ ประทับเวลาได้ — ทั้งสามดูเหมือนบรรยายเหตุการณ์เดียวกันจากสามมุมมอง: มุมสำรวจภายใน มุมสารสนเทศ และมุมสรีรวิทยา เส้นทางที่สั้นที่สุด เมื่อมาถึง ไม่ได้ถูกคำนวณเฉยๆ มันถูกประกาศด้วยกลไกที่วิวัฒนาการขึ้นมาเพื่อให้รางวัลจิตที่หาเส้นทางนั้นเจอ

When Beauty Lies

เมื่อความงามโกหก

EN

An instrument that fires reliably is not the same thing as an instrument that fires only when it should, and the honest half of this chapter is that the beauty-detector can be tripped by a representation that is short, symmetric, and wrong.

TH

ตัวตรวจจับความงามถูกหลอกได้ ด้วย representation ที่สั้น สมมาตร และผิด
เครื่องมือที่จุดขึ้นอย่างน่าเชื่อถือไม่ใช่สิ่งเดียวกับเครื่องมือที่จุดขึ้นเฉพาะตอนที่ควร
จุด และนี่คือครั้งที่ข้อสัจของบทนี้

EN

The clearest documented case sits not in mathematics but in physics. For much of the late twentieth century, theoretical particle physicists built and favored models — supersymmetry, grand unified theories, various proposed extensions to the Standard Model — substantially on the strength of their mathematical elegance: extra symmetries, "naturalness" (the absence of implausible fine-tuning between unrelated numbers), and a pleasing unification of forces that otherwise look unrelated. The physicist Sabine Hossenfelder's 2018 book *Lost in Math: How Beauty Leads Physics Astray* argues, from extended interviews with senior researchers in the field, that this aesthetic filter had by then hardened into something close to a substitute for evidence: theories were pursued and funded for decades in the absence of any experimental confirmation, propped up mainly by their formal beauty, while the field went through what she and others describe as a multi-decade drought of confirmed new fundamental physics (Hossenfelder, *Lost in Math: How Beauty Leads Physics Astray*, 2018; reviewed in Butterfield, "Lost in Math? A Review," 2019). Hossenfelder does not claim every elegant theory is false, and this book will not overstate her case into a verdict on string theory or supersymmetry as such — the argument inside physics remains genuinely contested among working researchers, which is itself worth stating plainly rather than smoothing over. What her book documents, carefully, is a *process risk*: a community that has trained its

members, generation after generation, to treat a certain kind of compression — symmetry, naturalness, unification — as evidence in its own right, independent of any observation confirming it.

TH

กรณีที่มีบันทึกขัดที่สุดอยู่ในฟิสิกส์ ไม่ใช่คณิตศาสตร์ ตลอดช่วงหลังของศตวรรษที่ยี่สิบ นักฟิสิกส์อนุภาคเชิงทฤษฎีสร้างและนิยมแบบจำลองต่างๆ — สมมาตรยิ่งยวด (supersymmetry) ทฤษฎีรวมแรงใหญ่ ส่วนขยายของ Standard Model ที่เสนอกันหลายแบบ — โดยใช้ความสง่างามทางคณิตศาสตร์ของมันเป็นน้ำหนักหลัก: สมมาตรที่เพิ่มเข้ามา "ความเป็นธรรมชาติ" (การไม่มีการปรับจูนละเอียดอย่างไม่น่าเชื่อระหว่างตัวเลขที่ไม่เกี่ยวข้องกัน) และการรวมแรงที่ดูไม่เกี่ยวข้องกันเข้าด้วยกันอย่างน่าพอใจ นักฟิสิกส์ Sabine Hossenfelder เขียนในหนังสือปี 2018 ของเธอ Lost in Math: How Beauty Leads Physics Astray โดยอิงการสัมภาษณ์นักวิจัยอาวุโสในสาขาอย่างยืดยาว ว่าตัวกรองทางสุนทรียะนี้แข็งตัวจนเกือบเป็นของแทนหลักฐานไปแล้ว ในตอนนั้น ทฤษฎีถูกไล่ตามและได้ทุนต่อเนื่องหลายทศวรรษโดยไม่มีการยืนยันจากการทดลองใดๆ คำไว้วางใจความงามเชิงรูปแบบของมันเป็นหลัก ขณะที่สาขาผ่านสิ่งที่เธอและคนอื่นบรรยายว่าเป็นภาวะแล้งฟิสิกส์มูลฐานใหม่ที่ได้รับการยืนยัน ยาวนานหลายทศวรรษ (Hossenfelder, Lost in Math: How Beauty Leads Physics Astray, 2018; บทวิจารณ์ใน Butterfield, "Lost in Math? A Review," 2019) Hossenfelder ไม่ได้อ้างว่าทฤษฎีที่สง่างามทุกทฤษฎีเป็นเท็จ และหนังสือเล่มนี้จะไม่ขยายข้อเสนอของเธอเป็นคำตัดสินต่อ string theory หรือ supersymmetry ข้อถกเถียงภายในฟิสิกส์ยังเป็นข้อพิพาทจริงในหมู่นักวิจัยที่ทำงานอยู่ และเรื่องนั้นควรพูดออกมาตรงๆ มากกว่ากลบให้เรียบ สิ่งที่หนังสือของเธอบันทึกไว้อย่างระมัดระวังคือความเสี่ยงของกระบวนการ: ชุมชนที่ฝึกสมาชิกของตัวเองรุ่นแล้วรุ่นเล่าให้ถือว่าการปิดกั้นชนิดหนึ่ง — สมมาตร ความเป็นธรรมชาติ การรวมแรง — เป็นหลักฐานในตัวมันเอง โดยไม่ขึ้นกับการสังเกตใดที่มายืนยัน

EN

This is the shadow side of everything else in this chapter. Chunking works because chess is a structured, stable domain where past compressions reliably predict future ones; it fails instantly, as de Groot showed, the moment the domain's structure is removed. The same trap catches minds, not just

perception: a solver who has been rewarded, correctly, a thousand times for noticing when a problem gets shorter can be fooled the thousand-and-first time by a representation that merely looks shorter — an elegant story standing in for a true one, a tidy equation standing in for a confirmed result. Premature closure — the sense of "this must be it" arriving before the checking has actually been done — is Wallas's verification stage being skipped because illumination felt too convincing to need it. The gamma burst in Jung-Beeman and Kounios's subjects fired for the *feeling* of insight; it said nothing about whether the compound word those subjects landed on was actually correct. Beauty, in every case this chapter has examined, is a detector for compression — real, measurable, and worth building instruments to study. It was never built to also detect truth, and the two only usually travel together.

TH

นี่คือด้านเงาของทุกอย่างที่เหลือในบทนี้ chunking ทำงานได้เพราะหมากรุกเป็นโดเมนที่มีโครงสร้างและเสถียร การบีบอัดในอดีตทำนายการบีบอัดในอนาคตได้อย่างเชื่อถือได้ พอโครงสร้างของโดเมนถูกถอดออก มันล้มทันที อย่างที่ de Groot แสดงให้เห็น กับดักเดียวกันนี้จับจิต ไม่ใช่แค่การรับรู้: นักแก้ปัญหาก็ได้รางวัลอย่างถูกต้องมาแล้วพันครั้งจากการสังเกตว่าปัญหาสั้นลง อาจถูกหลอกในครั้งที่พันหนึ่งด้วย representation ที่แค่ดูสั้นกว่า — เรื่องเล่าที่ส่งงมยืนยันแทนเรื่องที่จริง สมการที่เรียบร้อยยืนยันแทนผลที่ได้รับการยืนยัน การปิดก่อนกำหนด — ความรู้สึกที่ว่า "ต้องใช้แน่ๆ" มาถึงก่อนที่จะได้ตรวจสอบจริง — คือขั้นการตรวจสอบของ Wallas ที่ถูกข้ามไป เพราะการสว่างวาบ (illumination) รู้สึกน่าเชื่อเกินกว่าจะต้องการมัน คลื่นแถมมาที่พุ่งในผู้เข้าร่วมการทดลองของ Jung-Beeman กับ Kounios จุดขึ้นเพื่อความรู้สึกของ insight มันไม่ได้บอกอะไรเลยว่าคำประสมที่ผู้เข้าร่วมเหล่านั้นลงเอยด้วยถูกต้องหรือไม่ ความงาม ในทุกกรณีที่บทนี้ตรวจสอบ คือตัวตรวจจับการบีบอัด — จริง วัดได้ และคุ้มค่าที่จะสร้างเครื่องมือขึ้นมาศึกษา มันไม่เคยถูกสร้างมาให้ตรวจจับความจริงด้วย และสองอย่างนี้แค่เดินทางไปด้วยกันโดยส่วนใหญ่เท่านั้น

The Instrument, Calibrated

เครื่องมือวัด ที่ปรับเทียบแล้ว

EN

Pull the chapter's threads into one strand. Hardy's economy, Erdős's Book, proofs without words, Kolmogorov complexity, Minimum Description Length, Schmidhuber's compression progress, de Groot's chunked grandmasters, and the gamma burst in Jung-Beeman and Kounios's scanner are eight independent lines of evidence converging on a single mechanism: the felt sense of elegance is what it feels like, from the inside, when a problem's representation has just gotten dramatically shorter. That is not a metaphor borrowed from aesthetics to decorate a chapter on mathematics. It is closer to the reverse — aesthetics, in the register this book cares about, is a technical instrument the brain evolved for tracking exactly the quantity this book has been arguing about since chapter 00: not whether a path exists, but how short it can be made. Hossenfelder's physicists are proof that the instrument, like any instrument, drifts and needs calibration against the world outside itself.

TH

หลักฐานแปดสายที่เป็นอิสระต่อกันบรรจบที่กลไกเดียว: ความรู้สึกถึงความสง่างามคือสิ่งที่รู้สึกได้จากข้างใน เมื่อ representation ของปัญหาเพ่งสั้นลงอย่างฮวบฮาบ ดึงเส้นด้ายของบทนี้มารวมเป็นเกลียวเดียวแล้วจะเห็นทั้งแปด: ความประหยัดของ Hardy, The Book ของ Erdős, บทพิสูจน์ที่ไม่ใช้คำ, ความซับซ้อนโคลโมโกรอฟ, MDL, ความคืบหน้าของการบีบอัดของ Schmidhuber, แกรนด์มาสเตอร์ที่รวบเป็นก้อนของ de Groot และคลื่นเกมมาที่พุ่งในเครื่องสแกนของ Jung-Beeman กับ Kounios นั้นไม่ใช่อุปลักษณ์ที่ยืมจากสุนทรียศาสตร์มาประดับบทที่ว่าด้วยคณิตศาสตร์ มันใกล้เคียงกับด้านกลับมากกว่า — สุนทรียศาสตร์ในระดับที่หนังสือเล่มนี้สนใจ คือเครื่องมือทางเทคนิคที่สมองวิวัฒนาการขึ้นมาเพื่อติดตามปริมาณเดียวกับที่หนังสือเล่มนี้เถียงมาตั้งแต่บทที่ 00 เป๊ะๆ: ไม่ใช่ว่าเส้นทางมีอยู่หรือไม่ แต่วามันทำให้สั้นได้แค่ไหน นักฟิสิกส์ของ Hossenfelder คือหลักฐานว่าเครื่องมือนี้ก็เหมือนเครื่องมือทุกชิ้น มันเคลื่อน และต้องปรับเทียบกับโลกภายนอกตัวมันเอง

EN

That calibration is not a call to distrust the feeling. It is a call to use it the way any working scientist uses a sensitive but imperfect instrument: as the start of an investigation, never as its conclusion.

TH

การปรับเทียบนั้นไม่ใช่คำเรียกร้องให้เลิกใส่ใจความรู้สึก มันเรียกร้องให้ใช้ความรู้สึกแบบที่นักวิทยาศาสตร์ทำงานคนไหนก็ใช้เครื่องมือที่ไวแต่ไม่สมบูรณ์แบบ: เป็นจุดเริ่มของการสืบสวน ไม่ใช่ข้อสรุปของมัน

The Move

ท่าประจำบท

EN

Move 09 — Treat beauty as evidence, then verify.

TH

Move 09 — ถือความงามเป็นหลักฐาน แล้วตรวจสอบ

EN

A sudden sense that a problem has simplified is data, not a diploma. It tells you your internal compressor just found something — it does not yet tell you the something is true.

TH

ความรู้สึกฉับพลันว่าปัญหาย่อยลงคือข้อมูล ไม่ใช่ประกาศนียบัตร มันบอกว่าเครื่องบีบอัดภายในเพิ่งเจอบางอย่าง — ยังไม่ได้บอกว่าบางอย่างนั้นจริง

EN

- **Log the moment, don't just enjoy it.** When a representation snaps shorter — an analogy clicks, a proof collapses to three lines, a messy system suddenly looks symmetric — write down what changed before the feeling fades. Hadamard's mathematicians (chapter 03) could describe their incubations in detail decades later only because they had made a habit of capturing them.
- **Ask what got compressed, specifically.** Schmidhuber's account predicts the beauty is proportional to the *drop* in description length, not to the

absolute elegance of the final form. Name the drop — what long story did this short one replace? — and the claim becomes checkable instead of merely felt.

- **Run verification as a separate, adversarial step.** Wallas's fourth stage and Hossenfelder's caution are the same instruction from two centuries and two disciplines: illumination and verification are different acts, done with different mental postures, and skipping straight from the first to belief is how elegant wrong answers survive.
- **Distrust elegance fastest when it is unearned.** A grandmaster's chunk is compression bought by thousands of real games; a beautiful theory untested against a single new observation has paid no such price. The less verification a piece of elegance has survived, the more scrutiny it is owed before it is trusted.

TH

- **บันทึกช่วงเวลานั้น อย่าแค่เสกมัน** เมื่อ representation ดิดิ้นลง — การเทียบเคียงคลิกเข้าที่ บทพิสูจน์ยุบเหลือสามบรรทัด ระบบที่กรงรั้งๆ ดูสมมาตร — จดสิ่งที่เปลี่ยนไปก่อนความรู้สึก็จะจางหาย นักคณิตศาสตร์ของ Hadamard (บทที่ 03) บรรยายการบ่มความคิด (incubation) ของตัวเองได้ละเอียดหลายสิบปี ให้หลัง ก็เพราะพวกเขาติดนิสัยจุ่มมันไว้เท่านั้น
- **ถามให้เจาะจงว่าจะอะไรถูกบีบอัด** คำอธิบายของ Schmidhuber ทำนายว่าความงามแปรผันตามการหล่นลงของความยาวคำอธิบาย ไม่ใช่ตามความสง่างามสัมบูรณ์ของรูปสุดท้าย เรียกชื่อการหล่นลงนั้นให้ได้ — เรื่องยาวเรื่องไหนที่เรื่องสั้นนี้เข้าไปแทนที่ — แล้วขอกล่าวอ้างจะกลายเป็นสิ่งที่ตรวจได้ แทนที่จะเป็นแค่สิ่งที่รู้สึก
- **รับการตรวจสอบเป็นขั้นแยกต่างหาก และเป็นปฏิปักษ์** ขั้นที่สี่ของ Wallas กับคำเตือนของ Hossenfelder คือคำสั่งเดียวกันจากสองศตวรรษและสองสาขา: การสว่างวากับการตรวจสอบเป็นการกระทำคนละอย่าง ทำด้วยท่าทีทางความคิดคนละแบบ และการกระโดดจากอย่างแรกไปสู่ความเชื่อเลย คือวิธีที่คำตอบผิดอันสง่างามรอดชีวิตมาได้
- **ไม่ไวใจความสง่างามให้เร็วที่สุดเมื่อมันได้มาโดยไม่ได้ง่าย** chunk ของแกรนด์มาสเตอร์คือการบีบอัดที่ซื้อมาด้วยเกมจริงหลายพันเกม; ทัศนียที่งามซึ่งไม่เคยถูก

ทดสอบกับการสังเกตใหม่แม้แต่ครั้งเดียวยังไม่ได้จ่ายราคานั้น ยิ่งความสง่างามขึ้น
หนึ่งผ่านการตรวจสอบมาน้อยเท่าไร มันยิ่งเป็นหน้การตรวจสอบมากขึ้นเท่านั้น
ก่อนจะได้รับความไว้วางใจ

EN

Next — Chapter 10 · The Future Solver: the compression instrument this chapter measured in a human skull is now being built into machines that search spaces no human mind ever evolved to compress. What is left for the aesthetic sense to do when the search itself no longer needs it?

TH

ต่อไป — บทที่ 10 · นักแก้ปัญหาแห่งอนาคต: เครื่องมือวัดการบีบอัดที่บันทึกไว้ใน
กะโหลกมนุษย์ กำลังถูกสร้างเข้าไปในเครื่องจักรที่ค้นหาพื้นที่ซึ่งไม่มีจิตมนุษย์ใด
วิวัฒน์มาเพื่อบีบอัดมัน อะไรเหลือไว้ให้สัมผัสทางสุนทรียะท่า เมื่อการค้นหาเองไม่
ต้องการมันอีกต่อไป

CHAPTER 10

The Future Solver

นักแก้ปัญหาแห่งอนาคต

Seoul, March 2016

โซล มีนาคม 2016

EN

At the Four Seasons Hotel, on the tenth of March, Lee Sedol sat down for the second game of a five-game match against a piece of software he had predicted he would sweep 5-0 (AlphaGo versus Lee Sedol, Wikipedia). He had already lost game one. On move 37 of game two, playing Black, AlphaGo placed a stone on the fifth line, deep on the right side of the board, in a spot no strong human player would consider. Commentators watching the live feed assumed it was a bug. Fan Hui, the European champion AlphaGo had already beaten in a private match the previous October, said simply: "It's not a human move. I've never seen a human play this move" (DeepMind, AlphaGo). Lee Sedol left the table for over fifteen minutes. He resigned four and a half hours later, and AlphaGo led the match 2-0 (AlphaGo versus Lee Sedol, Wikipedia).

TH

Lee Sedol แพ้เกมแรกไปแล้ว ตอนที่เขานั่งลงเล่นเกมที่สองที่โรงแรม Four Seasons วันที่สิบมีนาคม คู่แข่งคือซอฟต์แวร์ชิ้นหนึ่งในแมตช์ห้าเกม ที่เขาเคยทำนายว่าตัวเองจะกวาด 5-0 (AlphaGo versus Lee Sedol, Wikipedia) หมากที่ 37 ของเกมสอง AlphaGo เล่นฝ่ายดำ วางหมากลงบนเส้นที่ห้า ลึกเข้าไปทางขวาของกระดาน ในจุดที่ไม่มีผู้เล่นมนุษย์ฝีมือดีคนไหนคิดจะวาง นักวิจารณ์ที่ดูถ่ายทอดสดคิดว่าเป็นบั๊ก Fan Hui แชมป์ยุโรปที่ AlphaGo เอาชนะไปแล้วในแมตช์ปิดเมื่อเดือนตุลาคมปีก่อน พูดสั้นๆ ว่า "มันไม่ใช่หมากของมนุษย์ ผมไม่เคยเห็นมนุษย์เดินหมากนี้" ("It's not a human move. I've never seen a human play this move")

(DeepMind, AlphaGo) Lee Sedol ลูกออกจากโต๊ะไปนานกว่าสิบห้านาที สี่ชั่วโมง

ครั้งให้หลังเขายอมแพ้ AlphaGo นำแมตช์ 2–0 (AlphaGo versus Lee Sedol, Wikipedia)

EN

The figure that stuck was a probability: AlphaGo's own policy network had estimated the odds of a human professional playing that stone at roughly one in ten thousand (DeepMind, AlphaGo). Go has been played for something like 2,500 years. Move 37 was not a memorized trick or a database lookup — Go's search space is larger than the number of atoms in the observable universe, too large for any lookup table — it was a stone nobody had ever recorded a human placing there, arrived at by a machine that had spent months playing games against itself. This book has spent nine chapters arguing that great solving is not deeper search through a fixed space but a change of space — new states, new operators, a shorter path. Move 37 is the moment that claim stopped being a description of human genius and became a description of something else: a system finding a shorter path through a space every human master believed they had already mapped.

TH

ตัวเลขที่ค้างอยู่ในความทรงจำคือความน่าจะเป็นตัวเดียว: โครงข่ายนโยบาย (policy network) ของ AlphaGo เองประเมินว่าโอกาสที่มีอาชีพมนุษย์จะวางหมากตรงนั้น อยู่ที่ราวหนึ่งในหมื่น (DeepMind, AlphaGo) โกะถูกเล่นมาราวสองพันห้าร้อยปี Move 37 ไม่ใช่กลที่ท่องจำมา ไม่ใช่การเปิดดูฐานข้อมูล — พื้นที่ค้นหาของโกะใหญ่กว่าจำนวนอะตอมในเอกภพที่สังเกตได้ ตารางค้นหาได้ก็รับไม่ไหว — มันคือหมากที่ไม่เคยมีบันทึกว่ามนุษย์คนไหนวางไว้ตรงนั้น และเครื่องจักรที่ใช้เวลาหลายเดือนเล่นกับตัวเองเดินทางไปถึงเอง หนังสือเล่มนี้ใช้เก๋าบเทียบว่าการแก้ปัญหาที่ยิ่งใหญ่ไม่ใช่การค้นหา (search) ที่ลึกลงไปในพื้นที่ตายตัว แต่คือการเปลี่ยนพื้นที่ — สถานะ (state) ชุดใหม่ ตัวดำเนินการ (operator) ชุดใหม่ เส้นทางที่สั้นลง Move 37 คือช่วงเวลาที่ยกกล่าวอ้างนั้นเลิกเป็นคำบรรยายอัจฉริยภาพของมนุษย์ แล้วกลายเป็นคำบรรยายของอย่างอื่น: ระบบหนึ่งหาเส้นทางที่สั้นกว่าได้ ในพื้นที่ที่ปรมาจารย์มนุษย์ทุกคนเชื่อว่าตัวเองทำแผนที่ไว้หมดแล้ว

EN

The generalization came fast. Within twenty months, DeepMind's successor system, AlphaZero, was given nothing but the rules of chess, shogi, and Go — no opening books, no grandmaster games, no domain-specific tuning — and told to play itself. Using one algorithm and one set of hyperparameters across all three games, it reached superhuman strength in each within twenty-four hours of self-play and beat the reigning world-champion program at each one (Silver et al., "A general reinforcement learning algorithm that masters chess, shogi and Go through self-play," 2017). The lesson of Move 37 was not "Go is solved." It was "the machine that reframes its own problem space, unaided, generalizes to problems its designers never showed it." That is the sentence this chapter has to take seriously, because it is the sentence the rest of the book has been building toward.

TH

การขยายผลมาเร็ว ภายในยี่สิบเดือน AlphaZero ระบบรุ่นถัดมาของ DeepMind ได้รับแค่กติกาของหมากรุก โชกิ และโกะ — ไม่มีตำราเปิดเกม ไม่มีเกมของแกรนด์มาสเตอร์ ไม่มีการปรับจูนเฉพาะโดเมน — แล้วถูกสั่งให้เล่นกับตัวเอง อัลกอริทึมเดียวไฮเปอร์พารามิเตอร์ชุดเดียว ใช้ข้ามทั้งสามเกม มันไปถึงระดับเหนือมนุษย์ในทุกเกมภายในยี่สิบสี่ชั่วโมงของการเล่นกับตัวเอง และเอาชนะโปรแกรมแชมป์โลกที่ครองตำแหน่งอยู่ในแต่ละเกม (Silver et al., "A general reinforcement learning algorithm that masters chess, shogi and Go through self-play," 2017) บทเรียนของ Move 37 ไม่ใช่ "โกะถูกแก้แล้ว" แต่คือ "เครื่องจักรที่เปลี่ยนกรอบ (reframe) พื้นที่ปัญหา (problem space) ของตัวเองได้โดยไม่มีใครช่วย ขยายผลไปยังปัญหาที่ผู้ออกแบบไม่เคยแสดงให้มันดู" บทนี้ต้องเอาจริงกับประโยคนั้น เพราะมันคือประโยคที่หนังสือทั้งเล่มมุ่งมาหา

The proof frontier

แนวหน้าของบทพิสูจน์

EN

Long before Move 37, mathematics had already had its own argument about whether a machine's path counts as a path at all. In 1976, Kenneth Appel and Wolfgang Haken, working with algorithms built by John Koch, proved the

four-color theorem — that any map can be colored with four colors so no two adjacent regions share one — by reducing the infinite space of possible maps to 1,936 unavoidable configurations and having a computer check each one by brute force, a case analysis covering roughly a billion individual cases that no human could perform by hand (AMS Blogs, "The Four Color Theorem"; Quanta Magazine, "Only Computers Can Solve This Map-Coloring Problem From the 1800s," 2023). It was the first major theorem proved with essential computer assistance, and it split the field: to purists, a proof must be checkable line by line by a human mind, and Appel–Haken's is not, so trusting it means trusting that a program ran correctly rather than trusting a chain of reasoning — a controversy that, fifty years later, has never fully closed (Quanta Magazine, 2023).

TH

คณิตศาสตร์เถียงกันเรื่องนี้มาก่อน Move 37 นานแล้ว: เส้นทางของเครื่องจักรนับเป็นเส้นทางหรือเปล่า ปี 1976 Kenneth Appel กับ Wolfgang Haken ทำงานร่วมกับอัลกอริทึมที่ John Koch สร้างขึ้น พิสูจน์ทฤษฎีบทสี่สี — แผนที่ใดก็ตามระบายได้ด้วยสี่สี โดยไม่มีสองพื้นที่ที่ติดกันใช้สีเดียวกัน — วิธีคือลดพื้นที่อนันต์ของแผนที่ที่เป็นไปได้ให้เหลือ 1,936 โครงแบบที่เรียงไม่ได้ แล้วให้คอมพิวเตอร์ตรวจทีละอันด้วยการไล่ทุกทาง (brute force) การวิเคราะห์กรณีครอบคลุมราวหนึ่งพันล้านกรณีย่อย ไม่มีมนุษย์คนไหนทำได้ด้วยมือได้ (AMS Blogs, "The Four Color Theorem"; Quanta Magazine, "Only Computers Can Solve This Map-Coloring Problem From the 1800s," 2023) มันคือทฤษฎีบทสำคัญข้อแรกที่พิสูจน์ได้โดยไม่ต้องพึ่งคอมพิวเตอร์อย่างขาดไม่ได้ และมันแบ่งวงการออกเป็นสองฝ่าย ฝ่ายเคร่งจารีตถือว่าบทพิสูจน์ต้องตรวจได้ทีละบรรทัดด้วยจิตมนุษย์ ของ Appel–Haken ไม่ใช่แบบนั้น ดังนั้นการเชื่อมั่นคือการเชื่อว่าโปรแกรมรันถูกต้อง ไม่ใช่การเชื่อสายโซ่ของการให้เหตุผล — ข้อโต้เถียงนี้หาสิบบิให้หลังก็ยังไม่เคยปิดสนิท (Quanta Magazine, 2023)

EN

The 2020s answer to "how do we trust a proof no one can read end to end" was not to stop using machines but to make the machine itself the reader. Lean, a proof assistant, and its community-built library mathlib let mathematicians write proofs in a language a computer can check line by line with total rigor — no step accepted on faith, ever. By the mid-2020s mathlib

had grown to more than 115,000 definitions and 232,000 theorems across roughly 1.5 million lines (Bianchini et al., "A Comprehensive Survey of the Lean 4 Theorem Prover," 2025), built by a community of well over a thousand contributors. Its coming-of-age moment arrived in December 2020, when Fields Medalist Peter Scholze publicly challenged the formalization community to verify, in Lean, a hard new theorem of his and Dustin Clausen's about liquid vector spaces — a result Scholze himself was not fully certain was correct. Eighteen months later, at a workshop in Providence, Rhode Island, the "Liquid Tensor Experiment" was completed: the theorem checked out, verified not by trust in a human referee but by a machine incapable of accepting an invalid step (Lean community blog, "Completion of the Liquid Tensor Experiment," 2022). The same library is now being used, in a separate and much longer project, to re-verify Andrew Wiles's proof line by line (chapter 08 told that proof's human story); here the machine is not the discoverer but the incorruptible reader.

TH

ทศวรรษ 2020 ตอบคำถาม "เราจะเชื่อบทพิสูจน์ที่ไม่มีใครอ่านจบตั้งแต่ต้นจนจบได้อย่างไร" ด้วยการทำให้เครื่องจักรเองเป็นผู้อ่าน ไม่ใช่ด้วยการเลิกใช้เครื่องจักร Lean คือผู้ช่วยพิสูจน์ (proof assistant) และ mathlib คือไลบรารีที่ชุมชนสร้างขึ้น ทั้งสองทำให้นักคณิตศาสตร์เขียนบทพิสูจน์ในภาษาที่คอมพิวเตอร์ตรวจได้ทีละบรรทัดด้วยความเข้มงวดเต็มที่ — ไม่มีชั้นไหนถูกรับด้วยความเชื่อใจ ไม่มีเลย พอถึงกลางทศวรรษ 2020 mathlib โตขึ้นเป็นนิยามกว่า 115,000 รายการและทฤษฎีบทกว่า 232,000 ข้อ กระจายอยู่ในโค้ดราว 1.5 ล้านบรรทัด (Bianchini et al., "A Comprehensive Survey of the Lean 4 Theorem Prover," 2025) สร้างโดยชุมชนผู้ร่วมสมทบที่มีมากกว่าหนึ่งพันคนไปมาก จุดที่มันเติบโตเป็นผู้ใหญ่คือเดือนธันวาคม 2020 Peter Scholze เจ้าของเหรียญ Fields ทำชุมชนการทำให้เป็นรูปนัย (formalisation) อย่างเปิดเผย ให้ตรวจสอบทฤษฎีบทใหม่ที่ยากข้อหนึ่งของเขากับ Dustin Clausen เรื่อง liquid vector spaces ด้วย Lean — ผลลัพธ์ที่ Scholze เองยังไม่มั่นใจเต็มร้อยว่าถูกต้อง สิบแปดเดือนต่อมา ที่เวิร์กช็อปในเมือง Providence รัฐ Rhode Island "Liquid Tensor Experiment" เสร็จสมบูรณ์: ทฤษฎีบทผ่าน ผู้ตรวจไม่ใช่มนุษย์ที่ต้องไวใจ แต่เป็นเครื่องจักรที่รับขั้นที่ไม่สมเหตุสมผลไม่ได้เลย (Lean community blog, "Completion of the Liquid Tensor Experiment," 2022)

ไลบรารีเดียวกันนี้กำลังถูกใช้ในโครงการที่แยกออกไปและยาวกว่ามาก เพื่อตรวจสอบบทพิสูจน์ของ Andrew Wiles ใหม่ที่ละบรรทัด (บทที่ 08 เล่าเรื่องราวฝั่งมนุษย์ของบทพิสูจน์นั้นไว้แล้ว); ตรงนี้เครื่องจักรไม่ใช่ผู้ค้นพบ แต่เป็นผู้อ่านที่ติดสินบนไม่ได้

EN

Discovery, not just verification, arrived next. On 25 July 2024, Google DeepMind announced that a combined system — AlphaProof, a reinforcement-learning prover built on the same self-play machinery as AlphaZero, paired with the geometry specialist AlphaGeometry 2 — had solved four of the six problems at the 2024 International Mathematical Olympiad, scoring 28 out of 42 points, the standard for a silver medal, one point short of gold (DeepMind, "AI achieves silver-medal standard solving International Mathematical Olympiad problems," 2024). AlphaProof solved the hardest problem on the paper outright — a problem only five of over six hundred human contestants solved in full (DeepMind, 2024). The full method was published in *Nature* in November 2025 (*Nature*, "Olympiad-level formal mathematical reasoning with reinforcement learning," 2025).

TH

ลำดับถัดไปคือการค้นพบ ไม่ใช่แค่การตรวจสอบ วันที่ 25 กรกฎาคม 2024 Google DeepMind ประกาศว่าระบบผสมชุดหนึ่งแก้โจทย์ได้สี่ในหกข้อของ International Mathematical Olympiad ปี 2024 ระบบนั้นคือ AlphaProof ตัวพิสูจน์แบบเรียนรู้เสริมกำลังที่สร้างบนกลไกเล่นกับตัวเองชุดเดียวกับ AlphaZero จับคู่กับ AlphaGeometry 2 ผู้เชี่ยวชาญเรขาคณิต คะแนน 28 จาก 42 เท่ากับเกณฑ์เหรียญเงิน ห่างจากเหรียญทองหนึ่งคะแนน (DeepMind, "AI achieves silver-medal standard solving International Mathematical Olympiad problems," 2024) AlphaProof แก้โจทย์ที่ยากที่สุดในชุดข้อสอบได้อย่างสมบูรณ์ — โจทย์ที่ผู้เข้าแข่งขันมนุษย์กว่าหกร้อยคนแก้ได้เต็มเพียงห้าคน (DeepMind, 2024) วิธีการฉบับเต็มตีพิมพ์ใน *Nature* เดือนพฤศจิกายน 2025 (*Nature*, "Olympiad-level formal mathematical reasoning with reinforcement learning," 2025)

E N

Gold came the following year, and it came from two directions at once, with an important difference between them. At the 2025 IMO, an experimental version of Google's Gemini Deep Think solved five of six problems for 35 of 42 points — and, unlike the 2024 result, the proofs were graded and certified by the actual IMO coordinators using the same rubric applied to the human contestants, a first for any AI system (DeepMind, "Advanced version of Gemini with Deep Think officially achieves gold-medal standard," 2025). OpenAI, days earlier, announced that an experimental general-purpose reasoning model had also scored 35 out of 42 under contest conditions, with three former IMO medalists grading the proofs — a result OpenAI reported itself rather than one certified by the Olympiad's own coordinators (OpenAI, 19 July 2025; Simon Willison's summary, 2025). That distinction — officially graded versus self-reported — matters more than either headline, and it recurs everywhere in this chapter: the machine's score is only as trustworthy as whoever is willing to check it against a rule nobody controls. A few months later Gemini Deep Think repeated the pattern in computer science, reaching gold-medal level at the 2025 ICPC World Finals, one of programming's hardest competitions (DeepMind, "Gemini achieves gold-medal level at the International Collegiate Programming Contest World Finals," 2025).

T H

เหรียญทองมาถึงในปีถัดมา จากสองทิศทางพร้อมกัน และความต่างระหว่างสองทิศทางนั้นสำคัญ ที่ IMO ปี 2025 Gemini Deep Think ของ Google เวอร์ชันทดลอง แก้ได้ห้าในหกข้อ ได้ 35 จาก 42 คะแนน — และต่างจากผลปี 2024 ตรงที่บทพิสูจน์ ถูกตรวจและรับรองโดยผู้ประสานงาน IMO ตัวจริง ด้วยเกณฑ์เดียวกับที่ใช้กับผู้เข้าแข่งขันมนุษย์ เป็นครั้งแรกสำหรับระบบ AI ใดๆ (DeepMind, "Advanced version of Gemini with Deep Think officially achieves gold-medal standard," 2025) ก่อนหน้านั้นไม่กี่วัน OpenAI ประกาศว่าโมเดลการให้เหตุผลเนกประสงค์รุ่นทดลอง ก็ทำได้ 35 จาก 42 เช่นกันภายใต้เงื่อนไขการแข่งขัน โดยมีอดีตผู้ได้เหรียญ IMO สามคนตรวจบทพิสูจน์ — OpenAI รายงานผลนี้เอง ผู้ประสานงานของโอลิมปิกไม่ได้รับรอง (OpenAI, 19 July 2025; บทสรุปของ Simon Willison, 2025) ความต่างตรงนั้น — ตรวจอย่างเป็นทางการ กับรายงานตัวเอง — สำคัญกว่าพาดหัวข่าวทั้งสองอัน

และมันวนกลับมาทุกทีในบทนี้: คะแนนของเครื่องจักรน่าเชื่อถือได้แค่เท่าที่มีใครสักคนยินดีตรวจมันกับกฎที่ไม่มีใครควบคุมได้ไม่กี่เดือนต่อมา Gemini Deep Think ทำแบบเดียวกันซ้ำในวิทยาการคอมพิวเตอร์ ไปถึงระดับเหรียญทองที่ ICPC World Finals ปี 2025 หนึ่งในการแข่งขันเขียนโปรแกรมที่ยากที่สุด (DeepMind, "Gemini achieves gold-medal level at the International Collegiate Programming Contest World Finals," 2025)

E N

By early 2026 the frontier had moved from competition problems, which have known answers, to genuinely open ones. Mathematician Terence Tao maintains a running public tally of AI contributions to the unsolved Erdős problems — the hundreds of open questions Paul Erdős left behind, catalogued at erdosproblems.com — and in January 2026 he documented a threshold crossing: Erdős problem #728 was solved "more or less autonomously" by OpenAI's GPT-5.2 Pro, staying faithful to the problem's intended spirit (which had itself only recently been correctly reconstructed by the erdosproblems.com community) and producing a result not found anywhere in the existing literature; a second tool, Aristotle, translated the proof into Lean, which caught and helped fix a few minor errors along the way (Tao, Mathstodon, 7 January 2026; The Decoder, 16 January 2026). Tao's own framing is the responsible one to borrow. Erdős problems vary in difficulty "by several orders of magnitude," and many of the easier ones were simply never tried by a serious mathematician, so a problem sitting open for fifty years does not mean it resisted fifty years of human effort (Tao, GitHub wiki, "AI contributions to Erdős problems," 2026). His estimate, as of that writing, is that only one to two percent of currently open Erdős problems are within reach of today's tools with minimal human guidance (Tao, GitHub wiki, 2026) — real progress, honestly bounded, on the lowest-hanging branch of a very tall tree.

T H

พอถึงต้นปี 2026 แนวหน้าเคลื่อนจากโจทย์แข่งขันที่มีคำตอบอยู่แล้ว ไปสู่โจทย์ที่เปิดอยู่จริง นักคณิตศาสตร์ Terence Tao ทำบัญชีสาธารณะที่เดินต่อเนื่องว่า AI มีส่วนแก้โจทย์ Erdős ที่ยังไม่ถูกแก้ไปแค่ไหน — คำถามเปิดหลายร้อยข้อที่ Paul

Erdős ที่ไว้รวบรวมไว้ที่ erdosproblems.com — และในเดือนมกราคม 2026 เขาบันทึกการข้ามกรณีประตุนี้: โจทย์ Erdős ข้อ #728 ถูกแก้ "โดยอัตโนมัติไม่มากนัก" ("more or less autonomously") โดย GPT-5.2 Pro ของ OpenAI คำตอบชื่อตรงต่อเจตนาที่โจทย์ตั้งใจไว้ (เจตนาที่ตัวเองก็เพิ่งถูกชุมชน erdosproblems.com ประกอบขึ้นใหม่ได้ถูกต้องเมื่อไม่นานมานี้) และให้ผลลัพธ์ที่ไม่ปรากฏอยู่ที่ไหนในวรรณกรรมที่มีอยู่; เครื่องมือตัวที่สองชื่อ Aristotle แปลบทพิสูจน์นั้นเป็น Lean ซึ่งจับและช่วยแก้ข้อผิดพลาดเล็กๆ ไม่ก็จุดระหว่างทาง (Tao, Mathstodon, 7 January 2026; The Decoder, 16 January 2026) กรอบที่ Tao วางไว้เองคือกรอบที่ควรยืมมาใช้ โจทย์ Erdős ยากต่างกัน "หลายอันดับของขนาด" และโจทย์ที่ง่ายกว่าหลายข้อก็แค่ไม่เคยมีนักคณิตศาสตร์จริงจังคนไหนลองทำ โจทย์ที่เปิดค้างมาห้าสิบปีจึงไม่ได้แปลว่ามันต้านความพยายามของมนุษย์มาห้าสิบปี (Tao, GitHub wiki, "AI contributions to Erdős problems," 2026) ประมาณการของเขา ณ เวลาที่เขียนนั้น: มีเพียงหนึ่งถึงสองเปอร์เซ็นต์ของโจทย์ Erdős ที่ยังเปิดอยู่ตอนนี้ ที่อยู่ในระยะเอื้อมของเครื่องมือปัจจุบันโดยมนุษย์ชี้แนะน้อยที่สุด (Tao, GitHub wiki, 2026) — ความคืบหน้าจริง ล้อมกรอบไว้อย่างซื่อสัตย์ บนกิ่งที่ต่ำที่สุดของต้นไม้ที่สูงมาก

Machines that invent algorithms

เครื่องจักรที่คิดค้นอัลกอริทึม

EN

Proving theorems is one kind of search. Inventing the algorithms other searches run on is a different, stranger kind — because now the "problem space" is the space of *possible algorithms*, and the object the machine hands back is itself a method, not an answer.

TH

การพิสูจน์ทฤษฎีบทคือการค้นหาแบบหนึ่ง การคิดค้นอัลกอริทึมที่การค้นหาอื่นวิ่งอยู่บนนั้นเป็นการค้นหาอีกแบบ และแปลกกว่า — เพราะตอนนี้ "พื้นที่ปัญหา" คือพื้นที่ของอัลกอริทึมที่เป็นไปได้ และสิ่งที่เครื่องจักรยื่นกลับมาคือวิธี ไม่ใช่คำตอบ

EN

In October 2022, DeepMind's AlphaTensor — built directly on AlphaZero's architecture, with matrix multiplication reframed as a single-player game of

decomposing a tensor — rediscovered the best known algorithms for multiplying matrices and, in several cases, beat them, finding new multiplication schemes 10 to 20 percent faster than the standard methods on real hardware such as Nvidia's V100 GPU (DeepMind/*Nature*, October 2022, via Wikipedia summary; MarkTechPost, October 2022). A year later, FunSearch — pairing a large language model with an automated checker and an evolutionary search loop — found a new construction for the cap set problem, a combinatorics question about how many points can be placed in a grid with no three in a line, that Terence Tao himself has called one of his favorite open questions; in eight dimensions, FunSearch found a cap set of 512 points, a result not present anywhere in its training data and unknown to the field (MIT Technology Review, "Google DeepMind used a large language model to solve an unsolved math problem," December 2023; DeepMind, "FunSearch").

T H

เดือนตุลาคม 2022 AlphaTensor ของ DeepMind ค้นพบอัลกอริทึมคูณเมทริกซ์ที่ดีที่สุดเท่าที่รู้จักซ้ำอีกครั้ง และในหลายกรณีก็เอาชนะมัน ระบบนี้สร้างบนสถาปัตยกรรมของ AlphaZero โดยตรง โดยเปลี่ยนกรอบการคูณเมทริกซ์ให้เป็นเกมผู้เล่นเดียวของการแยกส่วนเทนเซอร์ วิธีคูณแบบใหม่ที่มันพบเร็วกว่าวิธีมาตรฐาน 10 ถึง 20 เปอร์เซ็นต์บนฮาร์ดแวร์จริงอย่าง GPU V100 ของ Nvidia (DeepMind/*Nature*, October 2022, ผ่านบทสรุปของ Wikipedia; MarkTechPost, October 2022) หนึ่งปีให้หลัง FunSearch พบการสร้างแบบใหม่สำหรับปัญหา cap set ระบบนี้จับคู่โมเดลภาษาขนาดใหญ่เข้ากับตัวตรวจสอบอัตโนมัติและวงวนการค้นหาลำดับวิวัฒนาการ ปัญหา cap set เป็นคำถามเชิงการจัดหมู่ว่าจะวางจุดในตารางได้กี่จุดโดยไม่มีสามจุดใดอยู่ในแนวเดียวกัน Terence Tao เองเรียกมันว่าเป็นหนึ่งในคำถามเปิดที่เขาชอบที่สุด; ในแปดมิติ FunSearch พบ cap set ขนาด 512 จุด ผลลัพธ์นี้ไม่มีอยู่ที่ไหนในข้อมูลฝึกของมัน และวงวนไม่เคยรู้จัก (MIT Technology Review, "Google DeepMind used a large language model to solve an unsolved math problem," December 2023; DeepMind, "FunSearch")

E N

Then, in 2025, DeepMind generalized the trick into AlphaEvolve — a Gemini-powered coding agent that proposes candidate programs, has auto-

mated evaluators score them, and evolves the best candidates forward, applied to open problems across analysis, geometry, combinatorics, and number theory (DeepMind, "AlphaEvolve: A Gemini-powered coding agent for designing advanced algorithms," 2025; Novikov et al., "AlphaEvolve: A coding agent for scientific and algorithmic discovery," 2025). Across more than fifty such problems, AlphaEvolve matched the best known human solution roughly 75 percent of the time and improved on it about 20 percent of the time (DeepMind, 2025). Its single most striking result: a way to multiply two 4-by-4 matrices using 48 scalar multiplications instead of the 49 required by Volker Strassen's 1969 algorithm — the first improvement on that specific case in fifty-six years, and one that, unlike some earlier speedups, recurses cleanly to larger matrices (DeepMind, 2025). Chapter 06 described the mathematics of hardness — the wall that proves some problems cannot be compressed by anyone. AlphaTensor, FunSearch, and AlphaEvolve do not move that wall. They walk up very close to it, faster than any team of humans has walked, on problems the wall does not forbid.

TH

ปี 2025 DeepMind ขยายกลไกนี้ให้ทั่วไปขึ้นเป็น AlphaEvolve — เอเจนต์เขียนโค้ดที่ขับเคลื่อนด้วย Gemini มันเสนอโปรแกรมที่เป็นตัวเลือก ให้ตัวประเมินอัตโนมัติให้คะแนน แล้ววิวัฒนาการตัวเลือกที่ดีที่สุดต่อไปข้างหน้า ใช้กับปัญหาเปิดข้ามสาขาวิเคราะห์ เรขาคณิต การจัดหมู่ และทฤษฎีจำนวน (DeepMind, "AlphaEvolve: A Gemini-powered coding agent for designing advanced algorithms," 2025; Novikov et al., "AlphaEvolve: A coding agent for scientific and algorithmic discovery," 2025) จากปัญหาแบบนั้นกว่าห้าสิบล้อ AlphaEvolve เสมอกับคำตอบที่ดีที่สุดที่มนุษย์รู้จักราว 75 เปอร์เซ็นต์ของครั้ง และทำได้ดีกว่าราว 20 เปอร์เซ็นต์ของครั้ง (DeepMind, 2025) ผลที่สะดุดตาที่สุดชิ้นเดียวของมัน: วิธีคูณเมทริกซ์ 4 คูณ 4 สองตัวด้วยการคูณสเกลาร์ 48 ครั้ง แทนที่จะเป็น 49 ครั้งตามที่อัลกอริทึมปี 1969 ของ Volker Strassen ต้องใช้ — การปรับปรุงครั้งแรกในกรณีเฉพาะนั้นในรอบห้าสิบล้อปี และต่างจากการเร่งความเร็วบางอันก่อนหน้านี้ ตรงที่มันเริ่มที่อัลกอริทึมที่ใหญ่กว่าได้อย่างสะอาด (DeepMind, 2025) บทที่ 06 บรรยายคณิตศาสตร์ของความยาก (hardness) — กำแพงที่พิสูจน์ว่าบางปัญหาไม่มีใครบีบอัดได้ AlphaTensor,

FunSearch และ AlphaEvolve ไม่ได้ขยับกำแพงนั้น พวกมันเดินเข้าไปใกล้มันมากเร็วกว่าที่ทีมมนุษย์ทีมไหนเคยเดิน บนปัญหาที่กำแพงไม่ได้ห้าม

E N

All three systems are special-purpose: built, tuned, and scored against one narrow kind of problem. The more general shift, the one reshaping ordinary use of AI by mid-2026, is "test-time compute" or "test-time search" — letting a general-purpose language model spend extra computation *at the moment it is asked a question*, rather than only during training, generating, checking, and revising its own reasoning before committing to an answer. OpenAI introduced this publicly with the o1 model on 12 September 2024, describing a model trained to think longer, via reinforcement learning over chains of reasoning, before responding (OpenAI, "Learning to reason with LLMs," 2024). Three months later, o3 posted 87.5 percent on the ARC-AGI benchmark — a test built specifically to resist memorization — using a high-compute configuration, a leap the benchmark's own maintainers called unprecedented given that models had gone from roughly zero to five percent on the same test over the preceding four years (ARC Prize, "OpenAI o3 Breakthrough High Score on ARC-AGI-Pub," 2024). A Stanford team then showed, in January 2025, that the core trick could be reproduced cheaply: "budget forcing" — simply making a model keep reasoning by appending "Wait" when it tries to stop — reproduced OpenAI's scaling curves using a model fine-tuned on just a thousand examples (Muennighoff et al., "s1: Simple test-time scaling," 2025). By 2026 every major lab — OpenAI's o-series, Anthropic's extended-thinking Claude models, Google's Gemini Deep Think and Flash Thinking lines — ships some version of this: a model that, given a hard question, does not just recall an answer but runs an internal search over possible reasoning paths before committing to one. Chapter 05 traced search from Dijkstra's algorithm to A*; what "test-time search" means, as of this writing, is that the search chapter 05 described is now running silently, by default, inside the box that answers your question.

T H

ทั้งสามระบบเป็นระบบเฉพาะกิจ: สร้าง ปรับจูน และให้คะแนนกับปัญหาแคบชนิดเดียว การเปลี่ยนแปลงที่ทั่วไปกว่า และปรับปรุงการใช้ AI ในชีวิตประจำวันไปแล้ว

ภายในกลางปี 2026 คือ "test-time compute" หรือ "test-time search" — โมเดลภาษาอเนกประสงค์ได้ใช้การคำนวณเพิ่มในขณะที่ถูกถามคำถาม ไม่ใช่เฉพาะตอนฝึก มันสร้าง ตรวจสอบ และแก้การให้เหตุผลของตัวเองก่อนจะผูกมัดกับคำตอบ OpenAI เปิดตัวสิ่งนี้ต่อสาธารณะด้วยโมเดล o1 เมื่อวันที่ 12 กันยายน 2024 และอธิบายว่าเป็นโมเดลที่ถูกฝึกให้คิดนานขึ้นก่อนตอบ ผ่านการเรียนรู้เสริมกำลังบนสายการให้เหตุผล (OpenAI, "Learning to reason with LLMs," 2024) สามเดือนต่อมา o3 ทำคะแนน 87.5 เปอร์เซนต์บนเกณฑ์วัด (benchmark) ARC-AGI — ข้อสอบที่สร้างขึ้นมาเพื่อด้านการท่องจำโดยเฉพาะ — ด้วยการตั้งค่าแบบใช้การคำนวณสูง ผู้ดูแล benchmark เองเรียกการกระโดดนี้ว่าไม่เคยมีมาก่อน เพราะตลอดสี่ปีก่อนหน้า โมเดลไปได้จากราวศูนย์ถึงห้าเปอร์เซนต์บนข้อสอบเดียวกัน (ARC Prize, "OpenAI o3 Breakthrough High Score on ARC-AGI-Pub," 2024) เดือนมกราคม 2025 ทีมจาก Stanford แสดงให้เห็นว่ากลหลักนั้นทำซ้ำได้ในราคาถูก: "budget forcing" — แค่บังคับให้โมเดลให้เหตุผลต่อโดยเติมคำว่า "Wait" เข้าไปตอนมันพยายามจะหยุด — สร้างเส้นโค้งการขยายผลของ OpenAI ขึ้นมาใหม่ได้ ด้วยโมเดลที่ปรับละเอียดบนตัวอย่างเพียงหนึ่งพันชิ้น (Muennighoff et al., "s1: Simple test-time scaling," 2025) พอถึงปี 2026 ทุกแล็บใหญ่ — ซีรีส์ o ของ OpenAI, โมเดล Claude แบบ extended thinking ของ Anthropic, สาย Gemini Deep Think และ Flash Thinking ของ Google — ส่งเวอร์ชันหนึ่งของสิ่งนี้ออกมา: โมเดลที่เมื่อได้คำถามยาก ไม่ได้แค่ระลึกราคำตอบ แต่รันการค้นหภายในบนเส้นทางการให้เหตุผลที่เป็นไปได้ก่อนจะผูกมัดกับเส้นทางหนึ่ง บทที่ 05 ไล่การค้นหาจากอัลกอริทึมของ Dijkstra ไปถึง A*; "test-time search" ณ เวลาที่เขียนนี้หมายความว่า การค้นหาที่บทที่ 05 บรรยายไว้ ตอนนี้อยู่เยื้องๆ โดยปริยาย อยู่ในกล่องที่ตอบคำถามของคุณ

The centaur, and its expiry

เซนทอร์ และวันหมดอายุของมัน

EN

Chess told this story once already, a quarter-century before Move 37, and it is worth returning to precisely because it did not end the way its author expected.

TH

หมากรุกเล่าเรื่องนี้ไปแล้วครั้งหนึ่ง หนึ่งในสี่ศตวรรษก่อน Move 37 และคัมที่จะกลับไปดู เพราะมันไม่ได้จบแบบที่คนคิดมันขึ้นมาคาดไว้

EN

After losing his 1997 rematch to IBM's Deep Blue — the first defeat of a reigning world champion by a machine under tournament conditions — Garry Kasparov did not retreat from computers. He proposed "advanced chess": a human, assisted by an engine, playing another human similarly assisted, with the human making every final decision (Wikipedia, "Advanced chess"). The clearest proof of concept came in 2005, at the PAL/CSS Free-style tournament: two American amateurs, using three ordinary chess programs and no supercomputer, beat a field that included grandmasters running dedicated supercomputers and engines running unaided (Wikipedia, "Advanced chess"; AME3, "Centaur Chess"). Kasparov's summary of the result became one of the most quoted sentences in the whole literature on human-machine teams: "weak human + machine + better process was superior to a strong computer alone and, more remarkably, superior to a strong human + machine + inferior process" (Kasparov, "Weak Human, Strong Force," *War on the Rocks*, 2022). The lesson generalized far past chess: skill at directing a tool could outweigh the raw power of the tool, or of the person, alone.

TH

Garry Kasparov ไม่ได้ถอยหนีจากคอมพิวเตอร์ หลังแพ้แมตช์รีแมตช์ปี 1997 ให้ Deep Blue ของ IBM — ความพ่ายแพ้ครั้งแรกของแชมป์โลกที่ครองตำแหน่งอยู่ต่อเครื่องจักรภายใต้เงื่อนไขการแข่งขัน — เขาเสนอ "advanced chess": มนุษย์คนหนึ่งที่มีเอนจินช่วย เล่นกับมนุษย์อีกคนที่มีเอนจินช่วยเหมือนกัน โดยมนุษย์เป็นผู้ตัดสินใจ

ใจขึ้นสุดท้ายทุกครั้ง (Wikipedia, "Advanced chess") หลักฐานยืนยันแนวคิดที่ซัดที่สุดมาถึงในปี 2005 ที่การแข่งขัน PAL/CSS Freestyle: มือสมัครเล่นชาวอเมริกันสองคน ใช้โปรแกรมหมากรุกธรรมดาสามตัว ไม่มีซูเปอร์คอมพิวเตอร์ ชนะสนามที่มีทั้งแกรนด์มาสเตอร์ที่รันซูเปอร์คอมพิวเตอร์เฉพาะทาง และเอนจินที่รันตามลำพัง (Wikipedia, "Advanced chess"; AME3, "Centaur Chess") บทสรุปของ Kasparov ต่อผลนั้นกลายเป็นประโยคที่ถูกอ้างมากที่สุดประโยคหนึ่งในวรรณกรรมทั้งหมด ว่าด้วยทีมมนุษย์-เครื่องจักร: "มนุษย์อ่อน + เครื่องจักร + กระบวนการที่ดีกว่า เหนือกว่าคอมพิวเตอร์แข็งแกร่งที่อยู่ลำพัง และที่น่าทึ่งกว่านั้น เหนือกว่ามนุษย์แข็งแกร่ง + เครื่องจักร + กระบวนการที่ดีน้อยกว่า" ("weak human + machine + better process was superior to a strong computer alone and, more remarkably, superior to a strong human + machine + inferior process") (Kasparov, "Weak Human, Strong Force," *War on the Rocks*, 2022) บทเรียนนี้ขยายผลไปไกลเกินหมากรุก: ทักษะในการกำกับเครื่องมืออาจมีน้ำหนักมากกว่ากำลังดิบของเครื่องมือหรือของคนคนนั้น เพียงลำพัง

EN

That window did not stay open. As early as 2013, commentators tracking engine progress were already arguing it was getting hard to find any human contribution centaur teams still added over the best engine running by itself (Business Insider, "Humans Are On The Verge Of Losing One Of Their Last Big Advantages Over Computers," November 2013, cited in Wikipedia, "Advanced chess"). Kasparov himself pushed back as late as 2017, telling Tyler Cowen he remained confident a well-operated centaur team could still outperform a top supercomputer given the right human (Kasparov, conversation with Tyler Cowen, 2017). By 2023, a peer-reviewed study of centaur-chess players found something more specific: AI was not merely assisting human chess skill anymore, it was *substituting* for it, displacing the very judgment humans once contributed and replacing it with new, narrower complementary tasks (Krakowski, Luger & Raisch, "Artificial intelligence and the changing sources of competitive advantage," *Strategic Management Journal*, 2023). And by 2024, even the one decision advanced-chess players still made for themselves — which of several disagreeing engines to trust in a given position — had itself been automated: a DeepMind team

built a system that trains up to ten differently-specialized AlphaZero-style agents and has the group itself decide, through internal diversity and self-collaboration, which line of play to trust, outperforming any single engine, including on puzzles designed to defeat narrow engines (Zahavy et al., "Diversifying AI: Towards Creative Chess with AlphaZero," 2023; reported in Marginal Revolution, "'Centaur chess' is now run by computers," February 2024). The job title "centaur" still exists. The last task it used to describe — choosing which machine to believe — now belongs to a machine.

TH

หน้าต่างนั้นไม่ได้เปิดค้างไว้ ตั้งแต่ปี 2013 นักสังเกตการณ์ที่ติดตามความก้าวหน้าของเอนจินก็เถียงกันแล้วว่าเริ่มหาไม่เจอ ว่าทีมเซนทอร์ (centaur) ยังเพิ่มอะไรจากมนุษย์ให้เหนือกว่าเอนจินที่ดีที่สุดที่รันตามลำพัง (Business Insider, "Humans Are On The Verge Of Losing One Of Their Last Big Advantages Over Computers," November 2013, อ้างใน Wikipedia, "Advanced chess") Kasparov เองยังได้กลับเข้าถึงปี 2017 เขาบอก Tyler Cowen ว่ายังมั่นใจว่าทีมเซนทอร์ที่ควบคุมดีทำได้เหนือกว่าซูเปอร์คอมพิวเตอร์ระดับท็อป ถ้าได้มนุษย์ที่ใช้ (Kasparov, สนทนากับ Tyler Cowen, 2017) พอถึงปี 2023 งานวิจัยที่ผ่านการทบทวนโดยผู้เชี่ยวชาญเกี่ยวกับผู้เล่นหมากรุกเซนทอร์พบสิ่งที่เขาจะจกกว่านั้น: AI ไม่ได้แค่ช่วยเสริมทักษะหมากรุกของมนุษย์อีกต่อไป มันแทนที่ทักษะนั้น ไล่วิจารณ์ญาณที่มนุษย์เคยเป็นฝ่ายให้ออกไปแล้วแทนด้วยงานเสริมชนิดใหม่ที่แคบลง (Krakowski, Luger & Raisch, "Artificial intelligence and the changing sources of competitive advantage," Strategic Management Journal, 2023) พอถึงปี 2024 แม้แต่การตัดสินใจข้อเดียวที่ผู้เล่น advanced chess ยังทำเองอยู่ — จะเชื่อเอนจินตัวไหน ในบรรดาเอนจินที่เห็นไม่ตรงกันในตำแหน่งหนึ่ง — ก็ถูกทำให้อัตโนมัติไปด้วย: ทีม DeepMind สร้างระบบที่ฝึกเอเจนต์แบบ AlphaZero ที่เชี่ยวชาญต่างกันได้ถึงสิบตัว แล้วให้ตัวกลุ่มเองตัดสินใจว่าจะเชื่อแนวเล่นไหน ผ่านความหลากหลายภายในและการร่วมมือกันเอง ผลดีกว่าเอนจินเดี่ยวตัวใดๆ รวมถึงบนโจทย์ที่ออกแบบมาเพื่อเอาชนะเอนจินแคบๆ (Zahavy et al., "Diversifying AI: Towards Creative Chess with AlphaZero," 2023; รายงานใน Marginal Revolution, "'Centaur chess' is now run by computers," February 2024) ตำแหน่งงานชื่อ "เซนทอร์" ยังมีอยู่ งานสุดท้ายที่มันเคยหมายถึง — การเลือกว่าจะเชื่อเครื่องไหน — ตอนนี้เป็นของเครื่องแล้ว

E N

Nobody has published a single clean controlled experiment proving "engine alone beats centaur" across the board in 2026, and this chapter will not pretend one exists. What exists instead is the shape of a fifteen-year retreat: 2005, humans plus modest software beat grandmasters and supercomputers; 2013, observers start doubting the human adds anything; 2017, Kasparov still disagrees; 2023, a management-science study documents substitution, not augmentation; 2024, the residual human judgment call gets automated too. Chess is a closed, fully-observable, zero-sum game with an objectively checkable winner — the conditions most favorable to a machine closing a gap completely. The open question this book cannot yet answer with evidence, only pose honestly, is which other domains share enough of chess's structure to close the same way, and on what timeline. Advanced chess took about twenty years from Kasparov's proposal to its own obsolescence. Nothing says every field runs on that clock, and nothing says none will.

T H

ไม่มีใครตีพิมพ์การทดลองแบบมีกลุ่มควบคุมที่สะอาดขึ้นเดียวที่พิสูจน์ว่า "เอนจินตามลำพังชนะเซนทอร์" ในทุกกรณีในปี 2026 และบทนี้จะไม่สร้างว่ามีอยู่ สิ่งที่มีอยู่แทนคือรูปทรงของการถอยร่นสิบห้าปี: ปี 2005 มนุษย์บวกซอฟต์แวร์พอประมาณชนะแกรนด์มาสเตอร์และซูเปอร์คอมพิวเตอร์; ปี 2013 ผู้สังเกตการณ์เริ่มสงสัยว่ามนุษย์เพิ่มอะไรให้หรือเปล่า; ปี 2017 Kasparov ยังไม่เห็นด้วย; ปี 2023 งานวิจัยทางวิทยาการจัดการบันทึกการแทนที่ ไม่ใช่การเสริม; ปี 2024 การตัดสินใจที่เหลืออยู่ของมนุษย์ก็ถูกทำให้อัตโนมัติไปด้วย หากกรุกเป็นเกมปิด สังเกตได้ทั้งหมด ผลรวมเป็นศูนย์ และมีผู้ชนะที่ตรวจได้อย่างเป็นภาววิสัย — เงื่อนไขที่เอื้อที่สุดให้เครื่องจักรปิดช่องว่างได้สนิท คำถามที่ยังเปิดอยู่ และหนังสือเล่มนี้ตอบด้วยหลักฐานไม่ได้ได้แต่ตั้งขึ้นอย่างซื่อสัตย์ คือโดเมนอื่นใดบ้างที่มีโครงสร้างร่วมกับหมากรุกมากพอจะปิดแบบเดียวกัน และบนไทม์ไลน์ไหน advanced chess ใช้เวลาราวี่สิบปีจากข้อเสนอของ Kasparov จนถึงความล้าสมัยของตัวเอง ไม่มีอะไรบอกว่าทุกสาขาเดินตามนาฬิกาเรือนนั้น และไม่มีอะไรบอกว่าจะไม่มีสาขาไหนเดินตามเลย

What is still handed to someone

สิ่งที่ยังถูกส่งให้ใครสักคน

EN

So what, if anything, is left for a human solver once the search itself — proof search, algorithm search, move search, reasoning search — is something a machine does better and faster? The honest answer is not sentimental. It is structural, and it starts with a fact this book has repeated in every chapter: search only happens *inside a space*, and something, or someone, has to specify the space before the first move is searched.

TH

แล้วอะไรเหลืออยู่ให้นักแก้ปัญหาที่เป็นมนุษย์ ถ้ามีอะไรเหลืออยู่จริง เมื่อการค้นหาเอง — ค้นหาทฤษฎีบท ค้นหาอัลกอริทึม ค้นหาหมาก ค้นหาการให้เหตุผล — เป็นสิ่งที่เครื่องจักรทำได้ดีกว่าและเร็วกว่า คำตอบที่ชั่สตัยไม่ใช่คำตอบเชิงอารมณ์ มันเป็นเชิงโครงสร้าง และเริ่มจากข้อเท็จจริงที่หนังสือเล่มนี้พุดซ้ำในทุกบท: การค้นหาเกิดขึ้นภายในพื้นที่เท่านั้น และบางสิ่งหรือบางคนต้องกำหนดพื้นที่นั้นก่อนที่หมากแรกจะถูกค้นหา

EN

AlphaGo did not decide to play Go. AlphaProof did not decide the goal predicate was "solve six IMO problems in four and a half hours under contest rules." AlphaEvolve did not decide that 4-by-4 matrix multiplication was worth fifty-six years of nobody trying to beat Strassen. FunSearch did not decide the cap set problem mattered more than a million other combinatorics questions it could equally well have searched. In every documented case in this chapter, a human being — Demis Hassabis's team choosing which competitions to enter, Terence Tao curating which Erdős problems were even worth listing, the IMO's own committee writing six problems nobody had seen before — set the goal predicate and the space before any machine searched a single node. Chapter 03 called this Pólya's oldest heuristic: understand the problem before attempting to solve it. What has changed is that understanding-the-problem is now nearly the whole job, because attempting-to-solve-it has been delegated.

TH

AlphaGo ไม่ได้ตัดสินใจว่าจะเล่นโกะ AlphaProof ไม่ได้ตัดสินใจว่าเงื่อนไขเป้าหมายคือ "แก้โจทย์ IMO หกข้อในสี่ชั่วโมงครึ่งภายใต้กติกาการแข่งขัน" AlphaEvolve ไม่ได้ตัดสินใจว่าคุณสมบัติ 4 คุณ 4 คัมกับการที่ไม่มีใครพยายามเอาชนะ Strassen มาห้าสิบหกปี FunSearch ไม่ได้ตัดสินใจว่าปัญหา cap set สำคัญกว่าคำถามเชิงการจัดหมู่อีกล้านข้อที่มันค้นหาได้ดีพอๆกัน ในทุกกรณีที่มีบันทึกในบทนี้ มนุษย์เป็นผู้ตั้งเงื่อนไขเป้าหมายและพื้นที่ ก่อนที่เครื่องจักรใดจะค้นหาไหนดแรก — ทีมของ Demis Hassabis เลือกว่าจะลงแข่งรายการไหน Terence Tao คัดว่าโจทย์ Erdős ข้อไหนควรค่าแม้ว่าจะถูกลงรายการ คณะกรรมการของ IMO เองเขียนโจทย์หกข้อที่ไม่มีใครเคยเห็น บทที่ 03 เรียกสิ่งนี้ว่าวิธีเดาอย่างมีหลัก (heuristic) ที่เก่าแก่ที่สุดของ Pólya: เข้าใจปัญหาก่อนพยายามแก้มัน สิ่ง que เปลี่ยนไปคือการเข้าใจปัญหานี้แบบเป็นงานทั้งหมด เพราะการพยายามแก้มันถูกมอบต่อไปแล้ว

EN

This produces four things machines in this chapter still do not do, evidenced rather than assumed. First, *problem-finding*: nothing in the AlphaProof or AlphaEvolve pipeline decides which unsolved conjecture is worth an afternoon of compute; a mathematician or a curated public list does that upstream, every time. Second, taste in what counts as a good result: Tao's own commentary on the Erdős milestone spends more words distinguishing a genuine advance from a rediscovery than celebrating the win itself (Tao, GitHub wiki, 2026) — a judgment about significance, not correctness, that the tools cannot make about their own output. Third, setting the goal predicate: "score points under IMO rubric," "beat Strassen's multiplication count," "find a cap set bigger than 496" are all human-authored stopping conditions; change the predicate and the same search machinery hunts an entirely different, and possibly worthless, target. Fourth, verification and responsibility in the sense that matters legally and professionally: DeepMind needed actual IMO coordinators to certify Gemini's proofs before the "gold medal" claim meant anything, precisely because OpenAI's self-graded gold, days earlier, could not settle the question by itself (DeepMind, 2025; OpenAI, 2025). Someone external, accountable, and outside the system that produced the answer still has to be willing to sign it.

T H

สื่อย่างจึงยังเป็นสิ่งที่เครื่องจักรในบทนี้ทำไม่ได้ และแต่ละอย่างมีหลักฐานรองรับ ไม่ใช่สมมติเอา หนึ่ง การหาปัญหา (problem-finding): ไม่มีอะไรในสายงานของ AlphaProof หรือ AlphaEvolve ที่ตัดสินว่าข้อความคาดการณ์ที่ยังไม่ถูกพิสูจน์ข้อไหนคุ้มกับการคำนวณหนึ่งบ้าย; นักคณิตศาสตร์คนหนึ่งหรือรายการสาธารณะที่ผ่านการคัดสรรทำหน้าที้นั้นที่ต้นน้ำ ทุกครั้งไป สอง รสนิยมว่าอะไรนับเป็นผลลัพธ์ที่ดี: คำอธิบายของ Tao เองต่อหมุดหมาย Erdős ใช้คำแยกความกำกวมหน้าที่แท้จริงออกจากการค้นพบซ้ำ มากกว่าที่ใช้อยู่ชยชยชยนั้นเสียอีก (Tao, GitHub wiki, 2026) — นี่คือการตัดสินเรื่องนัยสำคัญ ไม่ใช่เรื่องความถูกต้อง และเครื่องมือทำกับผลงานของตัวเองไม่ได้ สาม การตั้งเงื่อนไขเป้าหมาย: "ทำคะแนนตามเกณฑ์ IMO" "เอาชนะจำนวนครั้งการคูณของ Strassen" "หา cap set ที่ใหญ่กว่า 496" ล้วนเป็นเงื่อนไขหยุดที่มนุษย์เขียนขึ้น; เปลี่ยนเงื่อนไข แล้วกลไกการค้นหาชุดเดิมก็จะไล่ล่าเป้าหมายที่ต่างออกไปโดยสิ้นเชิง และอาจไร้ค่า สี่ การตรวจสอบและความรับผิดชอบในความหมายที่สำคัญทางกฎหมายและทางวิชาชีพ: DeepMind ต้องใช้ผู้ประสานงาน IMO ตัวจริงมารับรองบทพิสูจน์ของ Gemini ก่อนที่คำว่า "เหรียญทอง" จะมี ความหมาย เพราะเหรียญทองที่ OpenAI ตรวจสอบเมื่อไม่กี่วันก่อนหน้านั้น ตัดสินคำถามนี้ด้วยตัวเองไม่ได้ (DeepMind, 2025; OpenAI, 2025) ยังต้องมีใครสักคนจากภายนอก รับผิดชอบได้ และอยู่นอกระบบที่ผลิตคำตอบนั้น ยินดีเซ็นรับรองมัน

E N

Put together, these four are one thing wearing four names: framing. Chapter 09 argued that elegance is the felt experience of compression — a representation snapping into a form where the answer is nearly free. That felt experience still belongs to whoever chooses which representation to snap, because the machines in this chapter do not yet reach upstream and pick their own problems out of the world's infinite supply of unasked questions. This is the spine of the whole book, restated at its point of maximum stress: **great solvers do not search harder; they change the space until the path is short** — and as of mid-2026, the machines are the most powerful walkers of a fixed space that have ever existed, while the choice of *which* space to hand them is still, in every verified case in this chapter, made by a person. That handoff is not a consolation prize. It is leverage of a kind that did not exist twenty years ago: one well-framed question, posed to a system

that can now search a space of matrix algorithms or IMO-level proofs to exhaustion overnight, moves further than a century of unaided human search ever could. The leverage of framing well has never been higher, because the machine on the other end of a well-posed question has never been stronger.

TH

สิ่งอย่างนี้คือสิ่งเดียวที่สวมชื่อสี่ชื่อ: การวางกรอบ บทที่ 09 เลี่ยงว่าความสง่างามคือประสบการณ์ที่รู้สึกได้ของการบีบอัด — representation ดีเข้าสู่รูปที่คำตอบแทบจะได้มาฟรี ประสบการณ์นั้นยังเป็นของคน que เลือกว่าจะดีดี representation ไหน เพราะเครื่องจักรในบทนี้ยังเอื้อมขึ้นไปทีต้นน้ำ แล้วหยิบปัญหาของตัวเองออกมาจากคลังคำถามที่ยังไม่มีใครถามอันไม่รู้จบของโลกไม่ได้ นี่คือแกนของหนังสือทั้งเล่ม กล่าวซ้ำ ณ จุดที่มันรับแรงมากที่สุด: นักแก้ปัญหาที่ยิ่งใหญ่ไม่ค้นหาหนักขึ้น — เขาเปลี่ยนพื้นที่จนเส้นทางสั้นลง — และ ณ กลางปี 2026 เครื่องจักรคือผู้เดินพื้นที่ตายตัวที่ทรงพลังที่สุดเท่าที่เคยมีมา ขณะที่การเลือกจะยืนยันพื้นที่ไหนให้พวกมัน ยังเป็นสิ่งที่คนทำ ในทุกกรณีที่ตรวจสอบได้ในบทนี้ การส่งต่อนั้นไม่ใช่รางวัลปลอบใจ มันคือคานงัดชนิดที่ยังไม่มีอยู่เมื่อยี่สิบปีก่อน: คำถามที่วางกรอบมาดีข้อเดียว ตั้งกับระบบที่ตอนนี้ค้นหาพื้นที่ของอัลกอริทึมเมทริกซ์หรือบทพิสูจน์ระดับ IMO ได้จนหมดเกลี้ยงภายในคืนเดียว เดินไปได้ไกลกว่าการค้นหาของมนุษย์ที่ไม่มีตัวช่วยตลอดหนึ่งศตวรรษ คานงัดของการวางกรอบให้ดีขึ้นไม่เคยสูงเท่านี้ เพราะเครื่องจักรที่ปลายอีกด้านของคำถามที่ตั้งมาดีไม่เคยแข็งแกร่งเท่านี้

The chapter that will age

บทที่จะเก่า

E N

Every other chapter in this book describes events that finished: Alexander's knot, Euler's bridges, Snow's pump, four decades of protein folds closed in one. This chapter describes a slope, measured at a single moment on it, and slopes are exactly the kind of fact a book cannot future-proof. State plainly which parts are load-bearing and which are perishable, because a reader in 2030 deserves to know which sentences to distrust first.

T H

บทนี้บรรยายความชัน ไม่ใช่เหตุการณ์ที่จบแล้ว ทุกบทอื่นในหนังสือเล่มนี้บรรยายเหตุการณ์ที่จบไปแล้ว: ปมของ Alexander, สะพานของ Euler, ปีม้าของ Snow, สี่ทศวรรษของการพับโปรตีนที่ปิดลงในคราวเดียว ความชันในบทนี้วัดไว้ ณ จุดเดียวบนความชันนั้น และความชันคือชนิดของข้อเท็จจริงที่หนังสือกั่นอนาคตไม่ได้เลย จึงต้องบอกตรงๆ ว่าส่วนไหนรับน้ำหนักและส่วนไหนเน่าเสียได้ เพราะผู้อ่านในปี 2030 สมควรได้รู้ว่าควรไม่ไวใจประโยคไหนก่อน

E N

The perishable claims, ranked by how fast they will decay: the specific benchmark numbers — o3's 87.5 percent on ARC-AGI, AlphaEvolve's 75/20 percent split against state of the art, the 35-out-of-42 IMO scores — are snapshots of specific model versions in 2024 and 2025 and will be beaten, probably within a year or two of this writing, by systems not yet named. The claim that "one to two percent of open Erdős problems" are AI-tractable is Tao's estimate from January 2026 and is, by its author's own description, a moving number that should already be higher by the time this is read (Tao, GitHub wiki, 2026). The claim that chess's centaur advantage has fully closed is the best-supported historical judgment available at this writing, but it rests on chess's unusually clean, closed, checkable structure, and the chapter's honest admission — that no equivalent clean study yet exists for less closed domains — is itself a claim about 2026's evidence base, not about the future's. What should not decay, because it does not depend on any model version: the four-way split between problem-finding, taste, goal-setting, and accountable verification, and the observation that every machine

result in this chapter still ran inside a space someone else specified. If that division collapses — if a system starts reliably choosing its own problems, its own stopping conditions, and standing behind its own answers without a human institution backstopping it — that will be the moment this chapter's central argument needs a rewrite, and whoever is reading this later will know it happened long before any historian gets around to writing the next edition.

TH

ข้อกล่าวอ้างที่น่าเสียได้ เรียงตามความเร็วที่มันจะเสื่อม: ตัวเลขเกณฑ์วัดเฉพาะเจาะจงเสื่อมเร็วที่สุด — 87.5 เปอร์เซ็นต์ของ o3 บน ARC-AGI, สัดส่วน 75/20 เปอร์เซ็นต์ของ AlphaEvolve เทียบกับสถานะล่าสุด, คะแนน 35 จาก 42 ที่ IMO — คือภาพนิ่งของโมเดลเวอร์ชันเฉพาะในปี 2024 และ 2025 และจะถูกทำลายสถิติโดยระบบที่ยังไม่มีชื่อ อาจภายในหนึ่งถึงสองปีนับจากที่เขียนนี้ ข้อกล่าวอ้างว่า "หนึ่งถึงสองเปอร์เซ็นต์ของโจทย์ Erdős ที่เปิดอยู่" อยู่ในระยะที่ AI จัดการได้ คือประมาณการของ Tao จากเดือนมกราคม 2026 และตามคำบรรยายของผู้เขียนเอง มันคือตัวเลขที่เคลื่อนอยู่ น่าจะสูงกว่านั้นไปแล้วเมื่อข้อความนี้ถูกอ่าน (Tao, GitHub wiki, 2026) ข้อกล่าวอ้างว่าความได้เปรียบของเซนทอร์ในหมากรุกปิดสนิทไปแล้ว คือคำตัดสินเชิงประวัติศาสตร์ที่มีหลักฐานหนุ่ดีที่สุดเท่าที่มี ณ เวลาที่เขียนนี้ แต่มันตั้งอยู่บนโครงสร้างของหมากรุกที่สะอาด ปิด และตรวจได้อย่างผิดปกติ และคำสารภาพอย่างซื่อสัตย์ของบทนี้ — ว่ายังไม่มีการวิจัยที่สะอาดเทียบเท่าสำหรับโดเมนที่ปิดน้อยกว่า — ก็เป็นข้อกล่าวอ้างเกี่ยวกับฐานหลักฐานของปี 2026 เอง ไม่ใช่เกี่ยวกับอนาคต สิ่งที่ไม่ควรเสื่อม เพราะไม่ขึ้นกับโมเดลเวอร์ชันใด คือการแยกสีทางระหว่าง การหาปัญหา รสนิยม การตั้งเป้าหมาย และการตรวจสอบที่มีผู้รับผิดชอบ กับ ข้อสังเกตว่าผลลัพธ์จากเครื่องจักรทุกชิ้นในบทนี้ยังรันอยู่ในพื้นที่ที่คนอื่นกำหนดไว้ ถ้าการแบ่งนั้นพังลง — ถ้าระบบหนึ่งเริ่มเลือกปัญหาของตัวเอง เจื่อนไขหุุดของตัวเอง และยื่นหลังคำตอบของตัวเองได้อย่างเชื่อถือได้ โดยไม่มีสถาบันของมนุษย์คอยรองรับ — นั่นคือช่วงเวลาข้อเสนอลหลักของบทนี้ต้องถูกเขียนใหม่ ใครก็ตามที่อ่านสิ่งนี้ในภายหลังจะรู้ว่ามันเกิดขึ้นแล้ว นานก่อนที่นักประวัติศาสตร์คนไหนจะได้ลงมือเขียนฉบับถัดไป

The Move

ทำประจำบท

EN

Move 10 — Delegate the search, keep the frame.

TH

Move 10 — มอบการค้นหาก็เครื่อง เก็บกรอบไว้กับตัว

EN

The machines in this chapter search better than any human who has ever lived, in every domain rigorous enough to hand them a checkable goal. That is not a threat to the century and a half of craft this book has documented; it is the craft's next tool, arriving on schedule. What it changes is where a solver's effort belongs.

TH

เครื่องจักรในบทนี้ค้นหาได้ดีกว่ามนุษย์คนไหนที่เคยมีชีวิตอยู่ในทุกโดเมนที่เข้มงวดพอจะยื่นเป้าหมายที่ตรวจได้ให้พวกมัน นั่นไม่ใช่ภัยคุกคามต่อฝีมือหนึ่งศตวรรษครึ่งที่หนังสือเล่มนี้บันทึกไว้ มันคือเครื่องมือขึ้นถัดไปของฝีมือนั้น มาถึงตามกำหนด สิ่งที่มันเปลี่ยนคือแรงของนักแก้ปัญหาควรไปอยู่ตรงไหน

EN

- **Push the search downstream, on purpose.** Once a problem has a checkable goal predicate — a proof that can be verified, a matrix count that can be measured, a benchmark that can be scored — treat exhaustive human search on it as a poor use of a scarce mind. That work is now a commodity chapter 06 would recognize as tractable; spend your hours before that point, not during it.
- **Spend the saved time upstream, where machines still cannot go.** Problem-finding, taste about which result would matter if found, the choice of goal predicate, and accountable verification are the four things this chapter found no machine doing unassisted. That is where a solver's judgment now earns its keep.
- **Never let a self-graded result stand as the verdict.** The 2025 IMO gap between an officially certified gold medal and a self-reported one is the model for every future claim a machine — or a person leaning on one —

makes about its own output. Someone outside the system that produced the answer has to be willing to check it and sign it.

- **Treat the frame as the last thing that is still entirely yours.** Every system in this chapter was pointed at its target by a human decision made before any search began. That decision has not been automated yet, anywhere, in any case this chapter could verify. Guard it accordingly.

TH

- **ด้านการค้นหาไปปลายน้ำ อย่างตั้งใจ** พอปัญหาหนึ่งมีเงื่อนไขเป้าหมายที่ตรวจได้ — บทพิสูจน์ที่ยืนยันได้ จำนวนครั้งการคูณที่วัดได้ benchmark ที่ให้คะแนนได้ — ให้ถือว่าการค้นหาแบบละเอียดถี่ถ้วนด้วยแรงมนุษย์บนปัญหานั้นเป็นการใช้จิตที่หายากอย่างสิ้นเปลือง งานนั้นตอนนี้เป็นสินค้าโภคภัณฑ์ที่บทที่ 06 จะเรียกว่า จัดการไหว (tractable); ใช้เวลาของคุณก่อนถึงจุดนั้น ไม่ใช่ระหว่างนั้น
- **ใช้เวลาที่ประหยัดได้ไปที่ต้นน้ำ ที่เครื่องจักรยังไม่ถึง** การหาปัญหา รสนิยมว่าผลลัพธ์ไหนจะสำคัญถ้าหาเจอ การเลือกเงื่อนไขเป้าหมาย และการตรวจสอบที่มีผู้รับผิดชอบ คือสิ่งที่บทนี้ไม่พบว่าเครื่องจักรใดทำได้เองโดยไม่มีคนช่วย นั่นคือจุดที่วิจารณ์ญาณของนักแก้ปัญหาหาเลี้ยงตัวเองได้ในตอนนี้
- **อย่าปล่อยให้ผลที่ตรวจเองยืนเป็นคำตัดสิน** ช่องว่างที่ IMO ปี 2025 ระหว่างเหรียญทองที่ได้รับการรับรองอย่างเป็นทางการ กับเหรียญทองที่รายงานเอง คือแบบจำลองของทุกข้อกล่าวอ้างในอนาคตที่เครื่องจักร — หรือคนที่ฟังเครื่องจักรอยู่ — พูดถึงผลงานของตัวเอง ต้องมีใครสักคนนอกกรอบที่ผลิตคำตอบนั้น ยินดีตรวจสอบมันและเซ็นรับรอง
- **ถือว่ารอบคือสิ่งสุดท้ายที่ยังเป็นของคุณทั้งหมด** ทุกระบบในบทนี้ถูกเลี้ยงไปเป้าหมายด้วยการตัดสินใจของมนุษย์ที่เกิดขึ้นก่อนการค้นหาใดจะเริ่ม การตัดสินใจนั้นยังไม่ได้ถูกทำให้อัตโนมัติที่ไหนเลย ในกรณีใดก็ตามที่บทนี้ตรวจสอบได้ ระวังรักษามันไว้ให้สมกับนั้น

E N

The moves are gathered now, chapter by chapter, into a kit — everything from questioning the frame at Gordium to delegating the search to a machine that will outwalk you. What remains is the shortest path of all: turning eleven moves into a single practiced route through any new problem, which is the business of the closing chapter, 99 · The Leverage Route.

T H

ตอนนี้บรรดาทำถูกรวบรวมไว้แล้ว บทต่อบท เป็นชุดเครื่องมือชุดหนึ่ง — ตั้งแต่การตั้งคำถามกับกรอบที่กอร์เดียม ไปจนถึงการมอบการค้นหาให้เครื่องจักรที่จะเดินแข่งคุณ สิ่งที่เหลืออยู่คือเส้นทางที่สั้นที่สุดของทั้งหมด: การเปลี่ยนสิบเอ็ดทำให้เป็นเส้นทางเดียวที่ฝึกจนคล่อง ผ่านปัญหาใหม่ใดๆ ซึ่งคืองานของบทปิดท้าย 99 · เส้นทางคานจัด

CHAPTER 99

The Leverage Route

เส้นทางคานงัด

The kit, assembled

ชุดเครื่องมือ ประกอบเสร็จแล้ว

EN

This book made one claim and spent eleven chapters earning it: **great solvers do not search harder; they change the space until the path is short.** You watched the claim hold across four thousand years — a scribe writing down a route instead of an answer, Euler throwing away a city, a wartime economist discovering that a thousand problems were one, a machine playing a move with a one-in-ten-thousand chance of occurring to a human. Different centuries, one silhouette: the space changes, then the walk is short.

TH

หนังสือเล่มนี้อ้างเพียงข้อเดียว แล้วใช้สิบเอ็ดบทพิสูจน์มัน: นักแก้ปัญหาที่ยิ่งใหญ่ไม่ค้นหาหนักขึ้น — เขาเปลี่ยนพื้นที่จนเส้นทางสั้นลง ข้ออ้างนี้ยืนอยู่ได้ตลอดสี่พันปีที่ผ่านมา อาลักษณ์คนหนึ่งเขียนเส้นทางลงไปแทนคำตอบ Euler โยนเมืองทั้งเมืองทิ้ง นักเศรษฐศาสตร์ยามสงครามพบว่าปัญหาพันข้อคือข้อเดียว เครื่องจักรเดินหมากตาที่มนุษย์มีโอกาสนึกถึงเพียงหนึ่งในหมื่น ต่างศตวรรษ เเงาร่างเดียวกัน: พื้นที่เปลี่ยน แล้วทางเดินก็สั้น

EN

Along the way, every chapter left you one Move — a transferable act with a name. Here is the complete kit. The gloss column is not a summary; it is a trigger — the situation in which that Move should come to hand by itself.

TH

ทุกบททิ้งทำให้หนึ่งท่าระหว่างทาง — การกระทำที่มีชื่อเรียกและยกไปใช้กับงานอื่นได้ ข้างล่างคือชุดเครื่องมือครบชุด คอลัมน์คำอธิบายไม่ใช่บทสรุป มันคือตัวจุดชนวน — สถานการณ์ที่ทำนั้นควรมาถึงมือเองโดยไม่ต้องเรียก

EN

#	Move	Reach for it when...
00	Question the frame before touching the rope	any problem arrives pre-packaged — before you spend a single unit of effort inside the package
01	Turn the method into an object	you have just done something well and could not say how — or badly, twice
02	Change the representation until the difficulty dissolves	the work feels like wading; every step is legal but nothing gets lighter
03	Carry named tools	you are staring, and nothing is happening — a heuristic is a bet about where paths hide
04	Optimize the whole system, and measure before you optimize	you are about to improve a part, and no one has timed the whole
05	Order the search: exploit structure so good paths come first	the options are many but not equal, and you can rank before you walk
06	Name the wall, then trade: reduce, approximate, randomize, or repurpose the hardness	honest effort keeps failing — check whether the space itself resists compression
07	Shrink the experiment: find the cheap measurement that stands in for the expensive world	each test costs a season, a fortune, or a reputation

#	Move	Reach for it when...
08	Find the load-bearing constraint: the one condition that, honored or broken, decides everything	a situation has a hundred rules and you suspect only one is structural
09	Treat beauty as evidence, then verify	something suddenly simplifies — log the click, then test it before you trust it
10	Delegate the search, keep the frame	a machine can walk the space faster than you — your job moves to choosing the space and grading arrivals

T H

#	ทำ	หยิบมาใช้เมื่อ...
00	ตั้งคำถามกับกรอบ ก่อนแตะเชือก	ปัญหาไหนก็ตามมาถึงในกล่องที่ห่อไว้เรียบร้อยแล้ว — ก่อนจะลงแรงแม้แต่หน่วยเดียวข้างในกล่อง
01	เปลี่ยนวิธีให้เป็นวัตถุ	เพิ่งทำอะไรได้ดีแต่บอกไม่ได้ว่าทำอย่างไร — หรือทำพลาดแบบเดิมเป็นครั้งที่สอง
02	เปลี่ยน representation จนความยากละลาย	งานให้ความรู้สึกเหมือนลุยน้ำ ทุกก้าวถูกกตึกาแต่ไม่มีอะไรเบาขึ้นเลย
03	พกเครื่องมือที่มีชื่อ	นั่งจ้องอยู่ และไม่มีอะไรเกิดขึ้น — heuristic คือการเดาฟันว่าเส้นทางซ่อนอยู่ตรงไหน
04	ปรับทั้งระบบ และวัดก่อนปรับ	กำลังจะปรับปรุงชิ้นส่วนหนึ่ง ทั้งที่ยังไม่มีใครจับเวลาทั้งระบบ
05	จัดลำดับการค้นหา — ใช้โครงสร้างต้นเส้นทางที่ดีขึ้นมาก่อน	ทางเลือกมีมาก แต่ไม่เท่ากัน และจัดอันดับได้ก่อนจะเริ่มเดิน
06	เรียกชื่อกำแพง แล้วแลก: ลดรูปประมาณ สุ่ม หรือเอาความยากไปใช้ประโยชน์	ลงแรงอย่างซื่อสัตย์แล้วยังล้มเหลวไม่หยุด — ตรวจสอบว่าตัวพื้นที่เองด้านการบีบอัดหรือเปล่า

#	ทำ	หยิบมาใช้เมื่อ...
07	ย่อการทดลอง — หากการวัดราคาถูกที่ยืนแทนโลกราคาแพงได้	การทดสอบแต่ละครั้งกินเวลาหนึ่งฤดูเงินหนึ่งกอง หรือชื่อเสียงหนึ่งชีวิต
08	หาข้อจำกัดที่รับน้ำหนัก — เจื่อนไขเดียวที่ทำตามหรือหักแล้วชี้ขาดทุกอย่าง	สถานการณ์หนึ่งที่มีกฎอยู่ร่อยร้อย และสงสัยว่ามีข้อเดียวที่เป็นโครงสร้างจริง
09	ถือความงามเป็นหลักฐาน แล้วตรวจสอบ	บางอย่างง่ายลงกะทันหัน — จดจ้องหะที่มันคลิก (the click) ไว้ แล้วทดสอบก่อนจะเชื่อ
10	มอบการค้นหาให้เครื่อง เก็บกรอบไว้กับตัว	เครื่องเดินพื้นที่ได้เร็วกว่าเรา — งานของเราย้ายไปอยู่กับการเลือกพื้นที่และการให้คะแนนสิ่งที่เดินมาถึง

E N

Eleven moves. They are not equals: 00 and 02 are the spine — the frame and the representation — and the other nine are what you do once the space is chosen. Which is why they compose in a definite order.

T H

สิบเอ็ดทำ น้ำหนักไม่เท่ากัน: 00 กับ 02 คือกระดูกสันหลัง — กรอบ กับรูปแบบการแทนปัญหา (representation) — อีกเก้าทำคือสิ่งที่ทำหลังเลือกพื้นที่ได้แล้ว เพราะอย่างนี้มันจึงประกอบกันตามลำดับที่แน่นอน

The standard opening

หมากเปิดมาตรฐาน

EN

Chess players do not reinvent the first ten moves of a game; they play an opening — a debugged sequence that reaches a playable middle game. Hard problems deserve the same courtesy. When something serious lands on your desk, this is the opening the whole book has been building:

TH

นักหมากรุกไม่คิดหมากสิบตาแรกขึ้นใหม่ทุกกระดาน เขาเล่นหมากเปิด — ลำดับที่ไล่ บ๊ักมาแล้วและพาไปถึงเกมกลางที่เล่นต่อได้ ปัญหายากๆ สมควรได้รับความสุภาพ แบบเดียวกัน เมื่อมีอะไรจริงจังมาวางบนโต๊ะ นี่คือหมากเปิดที่หนังสือทั้งเล่มสร้างมา ตลอด:

EN

First, refuse the package (00). Write one sentence: *what would count as solved?* Make it an outcome, not a procedure. List the rules you are obeying and mark which are real.

TH

หนึ่ง ปฏิเสธกล่องห่อ (00) เขียนหนึ่งประโยค: อะไรจึงนับว่าแก้ได้แล้ว ให้มันเป็นผลลัพธ์ ไม่ใช่ขั้นตอน ไล่รายการกฎที่กำลังทำตามอยู่ แล้วชี้ว่าข้อไหนจริง

EN

Second, choose the space (02). Draw the problem in at least two representations — a list, a graph, a table, a picture, a story. Keep the one in which the difficulty looks smallest. This is the highest-leverage minute you will spend.

TH

สอง เลือกพื้นที่ (02) วาดปัญหาด้วย representation อย่างน้อยสองแบบ — รายการ กราฟ ตาราง ภาพ เรื่องเล่า เก็บแบบที่ความยากดูเล็กที่สุดไว้ นาทีนี้ให้คำนึงถึงที่สุดในบรรดาเวลาทั้งหมดที่จะใช้ไป

EN

Third, check the wall (06). Is this problem a known monster in disguise? If the honest shape of it is intractable, decide the trade now — approximate, narrow the instance, or attack a different problem — rather than after a month of noble effort.

TH

สาม ตรวจสอบกำแพง (06) ถามว่าปัญหานี้คือสัตว์ประหลาดที่รู้จักกันดีปลอมตัวมาหรือเปล่า ถ้ารูปร่างที่แท้จริงของมันจัดการไม่ไหว (intractable) ให้ตัดสินใจแลกเสียตอนนี้ — ประมาณเอา หตกรณีให้แคบลง หรือหันไปโจมตีปัญหาอื่น — ไม่ใช่หลังจากทุ่มแรงอย่างสูงส่งไปแล้วหนึ่งเดือน

EN

Fourth, search — with tools, in order (03, 05). Name the heuristic you are using so you can notice when it fails. Rank the options so the promising paths come first. When a machine searches better, hand it the walking and keep the grading (10).

TH

สี่ ค้นหา — ด้วยเครื่องมือ และตามลำดับ (03, 05) เรียกชื่อ heuristic ที่กำลังใช้ จะได้รู้ตัวตอนมันพัง จัดอันดับทางเลือกให้เส้นทางที่มีแนวโน้มมาก่อน เมื่อเครื่องจักรค้นหาได้ดีกว่า ยกงานเดินให้มัน แล้วเก็บงานให้คะแนนไว้กับตัว (10)

EN

Fifth, buy information cheaply (07, 08). Before any expensive commitment, find the small experiment that stands in for the big world, and name the one constraint that decides everything.

TH

ห้า ซื้อข้อมูลให้ถูก (07, 08) ก่อนผูกมัดอะไรที่แพง หากการทดลองเล็กๆ ที่ยืนแทนโลกใบใหญ่ได้ และเรียกชื่อข้อจำกัดข้อเดียวที่ชี้ขาดทุกอย่าง

E N

Throughout, listen for the click (09) — and log the route (01). A sudden simplification is data; verify it. And when the problem yields, write down *how* — the method is the part of the answer you get to keep.

T H

ตลอดทาง ฟังหาจังหวะคลิก (09) — และจดเส้นทางไว้ (01) อะไรที่ง่ายลง
กะทันหันคือข้อมูล ตรวจสอบมัน และเมื่อปัญหายอมแพ้ เขียนลงไปว่า ทำอย่างไร —
วิธีคือส่วนของคำตอบที่เก็บไว้ใช้ได้

E N

A worked miniature. Suppose the problem is the drearily common one: *our weekly report takes two days to produce*. The package says "write the report faster." Refuse it: what counts as solved is *decisions get made with current numbers* — perhaps no report at all (00). Redraw: not a document but a pipeline — sources, transforms, a template; suddenly the two days are visibly three hours of queries and thirteen hours of formatting and chasing (02, and the measurement is 04's). The formatting is mechanical — delegate it to a machine under your template (10); the chasing violates a single load-bearing constraint — numbers are never where the report is written (08). The solved state turns out to be a dashboard and a fifteen-minute Monday conversation. Nobody searched harder. The space changed, and the path got short.

T H

ตัวอย่างย่อหนึ่งชิ้น สมมติปัญหาคือเรื่องที่เจอกันจนน่าเบื่อ: รายงานประจำสัปดาห์ของเราใช้เวลาทำสองวัน กล่องห่อบอกว่า "เขียนรายงานให้เร็วขึ้น" ปฏิเสธมัน สิ่งที่น่าจะแก้ได้แล้วคือ การตัดสินใจเกิดขึ้นบนตัวเลขล่าสุด — อาจไม่ต้องมีรายงานเลยก็ได้ (00) วาดใหม่: ไม่ใช่เอกสาร แต่เป็นท่อลำเลียง — แหล่งข้อมูล การแปลง แม่แบบ สองวันนั้นปรากฏชัดทันทีว่าเป็นคิวรีสามชั่วโมง กับจัดรูปแบบและตามงานอีกสิบสามชั่วโมง (02 ส่วนการวัดเป็นของ 04) จัดรูปแบบเป็นงานกลไก — มอบให้เครื่องทำภายใต้แม่แบบของเรา (10) ตามงานละเมิดข้อจำกัดที่รับน้ำหนักอยู่ข้อเดียว — ตัวเลขไม่เคยอยู่ที่เดียวกับที่เขียนรายงาน (08) สถานะที่น่าจะแก้ได้แล้วกลายเป็นแดชบอร์ดหนึ่งอัน กับบทสนทนาสิบห้านาทีเช้าวันจันทร์ ไม่มีใครค้นหาหนักขึ้น พื้นที่เปลี่ยน แล้วเส้นทางก็สั้นลง

The thirty-day route

เส้นทางฝึกสามสิบวัน

EN

Reading about moves does not install them. Use makes them reflexes — which is the difference between owning a book and owning a kit. The route below takes thirty days, roughly thirty minutes each except the two capstones. It assumes you have real problems; it works only on real problems. Do not practice on puzzles alone — puzzles have owners' frames, and the whole point is learning to question frames that matter.

TH

อ่านเรื่องทำไมได้ทำให้ทำติดตัว ใช้บ่อยต่างหากที่ทำให้มันเป็นปฏิกิริยาอัตโนมัติ — นั่นคือความต่างระหว่างมีหนังสือหนึ่งเล่มกับมีชุดเครื่องมือหนึ่งชุด เส้นทางข้างล่างใช้เวลาสามสิบวัน วันละราวสามสิบนาที ยกเว้นสองวันปิดท้าย มันสมมติว่ามีปัญหาจริงอยู่ในมือ และได้ผลกับปัญหาจริงเท่านั้น อย่าฝึกกับปริศนาอย่างเดียว — ปริศนามีกรอบของเจ้าของติดมาด้วย และประเด็นทั้งหมดคือหัดตั้งคำถามกับกรอบที่มีความหมายจริง

EN

Week 1 — Frames and spaces (chapters 00–02)

TH

สัปดาห์ที่ 1 — กรอบและพื้นที่ (บทที่ 00–02)

EN

- **Day 1.** Read 00. Pick one live problem. Write in one sentence what would count as solved. If your sentence names a procedure ("implement X"), rewrite it as an outcome.
- **Day 2.** Frame audit: list every rule you are obeying on that problem. Mark each *real* or *habit*. Ask of one habit-rule: who set this, and does it bind me?
- **Day 3.** Read 01. Take something you did well last week and write its route — steps, order, checks — as if teaching it. Name the method. You now own it as an object.

- **Day 4.** Read 02. Redraw one current problem in a different representation: prose into a diagram, a dispute into a table of claims and evidence, a process into boxes and arrows. Keep whichever drawing makes the difficulty smaller.
- **Day 5.** Gauss day. Find one repetitive task in your week and look for the pairing, symmetry, or batch structure that collapses it — the fifty-first way of seeing the same sum.
- **Day 6.** Königsberg day. Take one thing widely called impossible in your world and prove *which constraint* makes it impossible — count the thing that must balance and doesn't. An impossibility argument is a map of exactly what would have to change.
- **Day 7.** Review. Write the week's best reframe on one card, in one sentence. This card starts your deck.

TH

- **วันที่ 1.** อ่านบทที่ 00 เลือกปัญหาที่กำลังเกิดอยู่จริงหนึ่งข้อ เขียนหนึ่งประโยคว่าอะไรจึงน่าจะแก้ได้แล้ว ถ้าประโยคนั้นเรียกชื่อขั้นตอน ("ทำ X ให้เสร็จ") ให้เขียนใหม่เป็นผลลัพธ์
- **วันที่ 2.** ตรวจสอบ: ไล่รายการกฎทุกข้อที่กำลังทำตามอยู่กับปัญหานั้น ทำเครื่องหมายแต่ละข้อว่า จริง หรือ ความเคยชิน แล้วถามกฎที่เป็นความเคยชินข้อหนึ่งว่า ใครตั้งกฎนี้ และมันผูกมัดเราจริงหรือไม่
- **วันที่ 3.** อ่านบทที่ 01 หยิบสิ่งที่ทำได้ดีเมื่อสัปดาห์ก่อนมาหนึ่งอย่าง แล้วเขียนเส้นทางของมันออกมา — ขั้นตอน ลำดับ จุดตรวจ — เหมือนกำลังสอนคนอื่น ตั้งชื่อวิธีนั้น ตอนนั้นมันเป็นวัตถุที่ถือไว้ได้แล้ว
- **วันที่ 4.** อ่านบทที่ 02 วาดปัญหาปัจจุบันหนึ่งข้อขึ้นใหม่ด้วย representation แบบอื่น: ความเรียงเป็นแผนภาพ ข้อพิพาทเป็นตารางข้ออ้างกับหลักฐาน กระบวนการเป็นกล่องกับลูกศร เก็บภาพที่ทำให้ความยากเล็กลงไว้
- **วันที่ 5.** วัน Gauss หางานซ้ำๆ ในสัปดาห์ของตัวเองหนึ่งอย่าง แล้วมองหาการจับคู่ความสมมาตร หรือโครงสร้างแบบรวบเป็นก้อนที่ทำให้มันยุบลง — วิธีที่ห้าสิบเอ็ดในการมองผลบวกก่อนเดิม

- **วันที่ 6.** วัน Königsberg หยิบสิ่งหนึ่งที่คนในวงการเรียกกันว่าเป็นไปไม่ได้ แล้วพิสูจน์ว่า ข้อจำกัดข้อไหน ที่ทำให้มันเป็นไปไม่ได้ — นับสิ่งที่ต้องสมดุลแต่ไม่สมดุล ข้อโต้แย้งว่าเป็นไปไม่ได้คือแผนที่ที่บอกเบ๊ๆ ว่าอะไรจะต้องเปลี่ยน
- **วันที่ 7.** ทบทวน เขียนการเปลี่ยนกรอบที่ดีที่สุดของสัปดาห์ลงบนการ์ดหนึ่งใบ ในหนึ่งประโยค การ์ดใบนี้คือใบแรกของสำหรับ

E N

Week 2 — The named tools (chapter 03)

T H

สัปดาห์ที่ 2 — เครื่องมือที่มีชื่อ (บทที่ 03)

E N

- **Day 8.** Read 03. Make five heuristic cards: *analogy* · *invert* · *work backwards* · *relax a constraint* · *specialize first*. One line each on when it fires and how it fails.
- **Day 9.** Work backwards from a real goal: write the final state, then the state just before it, until you reach today. Note where the backward chain disagrees with your forward plan — that disagreement is information.
- **Day 10.** Inversion day. For one project ask: *how would I guarantee failure?* List five answers. Negate them. You now have the five things that actually matter.
- **Day 11.** TRIZ day. State one live contradiction — "we need it stronger AND lighter," "faster AND safer" — and refuse the compromise for thirty minutes; search for the resolution that gets both. Most compromises are surrenders negotiated too early.
- **Day 12.** Analogy day. Ask: what other field has my problem's *shape*? Queues, contagion, inventory, evolution, auctions — the distant field has usually been at it longer.
- **Day 13.** Incubation test. Load one hard problem carefully — read everything, state it precisely — then take a walk with no phone. Log what surfaces, even nothing. You are calibrating your own background process, not summoning magic.

- **Day 14.** Review. For each card in the deck: did it fire this week? What is its failure mode? A tool whose failure mode you cannot name is a superstition.

TH

- **วันที่ 8.** อ่านบทที่ 03 ทำการ์ด heuristic ห้าใบ: เทียบเคียง · กลับด้าน · ทำย้อนกลับ · คลายข้อจำกัด · เจาะกรณีเฉพาะก่อน ใบละหนึ่งบรรทัดว่ามันจุดติดเมื่อไหร่ และมันพังอย่างไร
- **วันที่ 9.** ทำย้อนกลับจากเป้าหมายจริงหนึ่งข้อ: เขียนสถานะสุดท้าย แล้วเขียนสถานะก่อนหน้ามันหนึ่งขั้น ไล่จนมาถึงวันนี้ สังเกตจุดที่โยย้อนกลับไม่ตรงกับแผนเดินทางที่วางไว้ — ความไม่ตรงกันนั้นคือข้อมูล
- **วันที่ 10.** วันกลับด้าน หยิบโครงการหนึ่งมาถามว่า จะทำอะไรให้มันล้มเหลวแน่นอน เขียนคำตอบห้าข้อ แล้วกลับด้านทั้งห้าข้อ ตอนนี้ในมือคือห้าเรื่องที่สำคัญจริง
- **วันที่ 11.** วัน TRIZ ตั้งความขัดแย้งที่มีอยู่จริงหนึ่งข้อ — "ต้องแข็งแรงขึ้นด้วยและเบาลงด้วย" "เร็วขึ้นด้วยและปลอดภัยขึ้นด้วย" — แล้วปฏิเสธการประนีประนอมเป็นเวลาสามสิบนาที ค้นหาทางแก้ที่ได้ทั้งสองอย่าง การประนีประนอมส่วนใหญ่คือการยอมแพ้ที่เจรจากันเร็วเกินไป
- **วันที่ 12.** วันเทียบเคียง ถามว่า สาขาไหนอีกบ้างที่มี รูปร่าง แบบเดียวกับปัญหาของเรา คิว การระบาด สินค้าคงคลัง วิวัฒนาการ การประมูล — สาขาที่อยู่ไกลออกไปมักทำเรื่องนี้มานานกว่า
- **วันที่ 13.** ทดสอบการบ่มความคิด (incubation) บ่อนปัญหายากหนึ่งข้อเข้าไปอย่างระมัดระวัง — อ่านให้หมด ตั้งใจทยอยให้ชัด — แล้วออกไปเดินโดยไม่เอาโทรศัพท์ไปด้วย จดสิ่งที่ผุดขึ้นมา แม้จะไม่มีอะไรผุดขึ้นเลยก็จด นี่คือการปรับเทียบกระบวนการเบื้องหลังของตัวเอง ไม่ใช่การเรียกเวทมนตร์
- **วันที่ 14.** ทบทวน ถามการ์ดทุกใบในสำหรับ: สัปดาห์นี้มันจุดติดไหม โหมดความล้มเหลวของมันคืออะไร เครื่องมือที่เรียกชื่อโหมดความล้มเหลวของมันไม่ได้ คือความเชื่อมมาย

E N

Week 3 — The formal engines (chapters 04–06)

T H

สัปดาห์ที่ 3 — เอนจินรูปนัย (บทที่ 04–06)

E N

- **Day 15.** Read 04. Instrument before improving: time one whole process end-to-end and find where the hours actually go. Expect to be wrong about it.
- **Day 16.** Whole-system day. Find one local optimization around you that damages the whole — the report polished for a meeting that shouldn't exist. Name the true objective it forgot.
- **Day 17.** Read 05. Draw a real decision as a six-node graph — states, transitions, costs — and run Dijkstra by hand. The exercise is not the arithmetic; it is feeling a decision become a *space*.
- **Day 18.** Strategy day. Classify three current tasks: greedy (take the best now), divide-and-conquer (split, solve, merge), or dynamic programming (overlapping subproblems — cache one repeated sub-answer this week and watch the savings compound).
- **Day 19.** Read 06. Reduction day: before inventing anything, ask *which solved problem is this in disguise?* Spend the thirty minutes searching the shelf, not the space.
- **Day 20.** Wall day. Identify one thing you brute-force that is genuinely hard — combinatorial, political, physical. Choose the trade explicitly: approximate, randomize, narrow the instance, or stop paying. An unnamed wall collects tolls forever.
- **Day 21.** Review. Write your personal "known monsters" list — the recurring problem shapes in your work and the standard trade for each.

T H

- **วันที่ 15.** อ่านบทที่ 04 คิดเครื่องวัดก่อนปรับปรุง: จับเวลากระบวนการหนึ่งตั้งแต่ต้นจนจบ แล้วหาว่าชั่วโมงหายไปไหนจริงๆ เตรียมใจไว้ว่าจะเดาผิด

- **วันที่ 16.** วันทั้งระบบ หากการปรับให้เหมาะสมที่สุดเฉพาะจุดรอบตัวหนึ่งอย่างที่ทำร้ายทั้งระบบ — รายงานที่ชัดเจนเงินเพื่อการประชุมที่ไม่ควรมีอยู่ตั้งแต่แรก เรียกชื่อเป้าหมายที่แท้จริงซึ่งมันลึ้มไป
- **วันที่ 17.** อ่านบทที่ 05 วาดการตัดสินใจจริงหนึ่งเรื่องเป็นกราฟหกโหนด — สถานะการเปลี่ยนสถานะ ต้นทุน — แล้วเดิน Dijkstra ด้วยมือ แบบฝึกหัดนี้ไม่ได้อยู่ที่การคิดเลข แต่อยู่ที่การรู้สึกว่าการตัดสินใจหนึ่งกลายเป็น พื้นที่
- **วันที่ 18.** วันกลยุทธ์ จัดประเภทงานปัจจุบันสามชิ้น: โลก (greedy — เอาสิ่งที่ดีที่สุดตอนนี้) แบ่งแล้วพิชิต (แยก แยก รวม) หรือกำหนดการพลวัต (dynamic programming — ปัญหาย่อยที่ทับซ้อนกัน ลองแก้คำตอบย่อยที่ต้องใช้ซ้ำไว้หนึ่งอันในสัปดาห์นี้ แล้วดูว่าที่ประหยัดได้ทบต้นขึ้นแค่ไหน)
- **วันที่ 19.** อ่านบทที่ 06 วันลตรูป: ก่อนจะประดิษฐ์อะไร ให้ถามว่า นี่คือนปัญหาที่มีคนแก้ไปแล้วข้อไหนที่ปลอมตัวมา ใช้สามสิบนาที่นั่นค้นชั้นหนังสือ ไม่ใช่ค้นพื้นที่
- **วันที่ 20.** วันกำแพง ระบุสิ่งหนึ่งที่ตัวเองไล่ทุกทาง (brute force) อยู่และมันยากจริงๆ — ยากเชิงการจัดหมู่ เชิงการเมือง หรือเชิงกายภาพ เลือกรากได้แลกเสียออกมาให้ชัด: ประมาณเอา สุ่ม หดกรณีให้แคบลง หรือเลิกจ่าย กำแพงที่ไม่มีใครเรียกชื่อจะเก็บค่าผ่านทางไปตลอดกาล
- **วันที่ 21.** ทบทวน เขียนรายการ "สัตว์ประหลาดที่รู้จักแล้ว" ของตัวเอง — รูปร่างปัญหาที่วนกลับมาในงานของตัวเอง และการแลกได้แลกเสียมาตรฐานของแต่ละรูปร่าง

EN

Week 4 — Galleries, beauty, machines (chapters 07–10)

TH

สัปดาห์ที่ 4 — หอศิลป์เล่า ความงาม เครื่องจักร (บทที่ 07–10)

EN

- **Day 22.** Read 07. Shrink one experiment: replace an expensive test in your world with a cheap proxy — and write down what the proxy cannot see, because that blindness is the price and you should pay it knowingly.

- **Day 23.** Read 08. Constraint day: for one live project, name the load-bearing constraint. Then either honor it completely or price what breaking it would cost. The hundred other rules are decoration.
- **Day 24.** Teaching day. Retell one gallery case — Snow's map, the scrubber, Rejewski's permutations — to another person, from memory, ending with its move in one line. If the retelling sags, you stored the story but not the structure.
- **Day 25.** Read 09. Start a beauty log: each time something suddenly simplifies this week, write the click down — and schedule its verification. The click is evidence, not verdict.
- **Day 26.** Elegant-wrong day. Take one tidy story you currently believe and attack it: what evidence would break it? The seduction chapter (09) exists because this exercise is the one solvers skip.
- **Day 27.** Read 10. Delegation drill: give an AI engine one real search *under your frame* — write the goal predicate and constraints first, then grade the arrivals against the predicate, then log where your frame leaked. The leak log is the lesson.
- **Day 28.** Centaur audit. Two columns: where machines now beat you (delegate, without nostalgia) and where you still add value (double down, without complacency). Date the page; it will age, and the dated page is the point.

TH

- **วันที่ 22.** อ่านบทที่ 07 ย่อการทดลองหนึ่งอย่าง: แทนที่การทดสอบราคาแพงในงานของตัวเองด้วยตัวแทนราคาถูก — แล้วเขียนลงไปว่าตัวแทนนั้นมองไม่เห็นอะไร เพราะจุดบอดนั้นคือราคาที่จ่าย และควรจ่ายมันทั้งที่รู้ตัว
- **วันที่ 23.** อ่านบทที่ 08 วันข้อจำกัด: หยิบโครงการที่กำลังเดินอยู่หนึ่งโครงการ แล้วเรียกชื่อข้อจำกัดที่รับน้ำหนัก จากนั้นเลือกอย่างใดอย่างหนึ่ง คือทำตามมันให้ครบถ้วน หรือตีราคาว่าการหักมันจะแพงแค่ไหน กฎอีกร้อยข้อคือเครื่องประดับ
- **วันที่ 24.** วันสอน เล่ากรณีจากห่อเรื่องเล่าหนึ่งเรื่อง — แผนที่ของ Snow เครื่องกรอง CO2 ของ Apollo 13 การเรียงสับเปลี่ยนของ Rejewski — ให้อีกคนฟังจากความจำ แล้วปิดท้ายด้วยท่าของมันเป็นหนึ่งในบรรทัด ถ้าเล่าแล้วเรื่องยาน แปลว่าเก็บเรื่องไว้ได้ แต่ไม่ได้เก็บโครงสร้าง

- **วันที่ 25.** อ่านบทที่ 09 เริ่มบันทึกความงาม: ทุกครั้งที่มือจะไถ่ลงกะทันหันในสัปดาห์นี้ให้จดจ้องหะคลิกนั้นไว้ — แล้วนัดเวลาตรวจสอบมัน จังหวะคลิกคือหลักฐาน ไม่ใช่คำตัดสิน
- **วันที่ 26.** วันสง่างามแต่ผิด หยิบเรื่องเล่าที่เรียบบร้อยลงตัวซึ่งตัวเองเชื่ออยู่ตอนนี้มาหนึ่งเรื่อง แล้วโจมตีมัน: หลักฐานแบบไหนจะหักมันได้ บทว่าด้วยการถูกล่อลวง (09) มียู่เพราะแบบฝึกหัดข้อนี้คือข้อที่นักแก้ปัญหาข้าม
- **วันที่ 27.** อ่านบทที่ 10 ซ้อมมอบหมายงาน: ให้เอนจิน AI ค้นหาจริงหนึ่งงาน ภายใต้กรอบของเราเอง — เขียนภาคแสดงของเป้าหมาย (goal predicate) และข้อจำกัดก่อน แล้วให้คะแนนสิ่งที่เดินมาถึงเทียบกับภาคแสดงนั้น จากนั้นจดว่ากรอบของเราเร็วตรงไหน บันทึกกรวยรั้วคือบทเรียน
- **วันที่ 28.** ตรวจสอบสถานะเซนเทอร์ (centaur) สองคอลัมน์: ตรงไหนที่เครื่องจักรชนะเราไปแล้ว (มอบให้มันไป โดยไม่ต้องคิดถึงวันเก่า) และตรงไหนที่เรายังเพิ่มคุณค่าได้ (ทุ่มลงไปอีก โดยไม่ประมาท) ลงวันที่บนหน้ากระดาษด้วย มันจะเก่า และหน้ากระดาษที่ลงวันที่ไว้นั้นแหละคือประเด็น

EN

Capstones

TH

งานปิดท้าย

E N

- **Day 29.** The full opening, live. Take one important problem and run the standard opening end-to-end in a timed ninety minutes, whole kit on the table. Write the path you took — which moves fired, in what order, what changed.
- **Day 30.** Write **Move 11 — yours.** Name, trigger, failure mode, in the house format. Somewhere in your domain is a move this book does not contain, one your own scars have taught you. The kit was never meant to arrive complete. It was meant to arrive *extensible*.

T H

- **วันที่ 29.** เดินหมากเปิดเต๋มรูปแบบกับของจริง หยิบปัญหาสำคัญหนึ่งข้อ แล้วเดินหมากเปิดมาตรฐานตั้งแต่ต้นจนจบภายในเก้าสิบนาทีที่จับเวลา วางชุดเครื่องมือทั้งชุดไว้บนโต๊ะ เขียนเส้นทางที่เดินไป — ทำไหนจุดติดบ้าง ตามลำดับไหน อะไรเปลี่ยนไป
- **วันที่ 30.** เขียน **Move 11 — ทำของตัวเอง** ชื่อ ตัวจุดชนวน โหมดความล้มเหลวตามรูปแบบประจำบ้าน ที่ไหนสักแห่งในสาขาของตัวเองมีท่าที่หนังสือเล่มนี้ไม่มีท่าที่แปลเป็นของตัวเองสอนมา ชุดเครื่องมือนี้ไม่เคยตั้งใจจะมาถึงในสภาพครบสมบูรณ์ มันตั้งใจจะมาถึงในสภาพที่ ต่อเติมได้

The solver's checklist

เช็คลิสต์ของนักแก้ปัญห

E N

Before: What counts as solved — as an outcome? Which rules are real? In what representation is this smallest? Is it a known monster?

T H

ก่อนเริ่ม: อะไรนับว่าแก้ได้แล้ว — ในรูปของผลลัพธ์ กฎข้อไหนจริง ปัญหานี้เล็กที่สุดใน representation แบบไหน มันคือสัตว์ประหลาดที่รู้จักกันคืออยู่แล้วหรือเปล่า

EN

During: What tool am I using — by name? Are good paths coming first? Has honest effort failed enough to suspect a wall? Did something click — and did I log it?

TH

ระหว่างทำ: กำลังใช้เครื่องมืออะไร — เรียกชื่อมันออกมา เส้นทางที่ดีขึ้นมาก่อนหรือยัง ลงแรงอย่างซื่อสัตย์แล้วล้มเหลวมากพอจะสงสัยว่ามีกำแพงหรือยัง มีอะไรคลิกใหม่ — และจดไว้หรือเปล่า

EN

After: Write the route. Shorten it once — the second draft is where elegance lives. Shelve the method as an object with a name. What would make this problem never return?

TH

หลังเสร็จ: เขียนเส้นทางลงไป ย่อมันอีกหนึ่งรอบ — ร่างที่สองคือที่อยู่ของความสง่างาม เก็บวิธีขึ้นชั้นในฐานะวัตถุที่มีชื่อ อะไรจะทำให้ปัญหานี้ไม่กลับมาอีกเลย

EN

That last question deserves its line. The chapters kept meeting solvers whose deepest solution was to make the problem extinct — Semmelweis did not treat childbed fever better, he removed its cause from the doctors' hands. The shortest path of all is the one nobody has to walk again.

TH

คำถามข้อสุดท้ายสมควรจะมีบรรทัดของตัวเอง หลายบทที่ผ่านมาเจอนักแก้ปัญหาก็ทางแก้ที่ดีที่สุดคือทำให้ปัญหาสูญพันธุ์ — Semmelweis ไม่ได้รักษาไข้หลังคลอดให้เก่งขึ้น เขาเอาสาเหตุออกไปจากมือหมอ เส้นทางที่สั้นที่สุดในบรรดาทั้งหมดคือเส้นทางที่ไม่มีใครต้องเดินอีก

Problems worth having

ปัญหาที่ควรค่าแก่การมี

EN

One skill remains, and the book has been circling it since Gordium: the moves tell you how to solve, not what. Machines have made the *how* abund-

ant — chapter 10's engines will walk almost any space you can specify, and they get better every season the ink on that chapter dries. Which means the scarce act, the one the whole apparatus now waits on, is the choice of problem. Problem-finding — taste — is the last unautomated move, and taste is built the same way the others were: by keeping a route log, by noticing which solved problems left the world lighter and which merely left it optimized, by asking of every candidate knot not "can I undo this?" but "what comes free if I do?"

TH

เหลือทักษะอีกหนึ่งอย่าง หนังสือเล่มนี้วนรอบมันมาตั้งแต่กอร์ดีย์ม: ทำทั้งหมดบอกว่าแก้อย่างไร ไม่ได้บอกว่าแก่อะไร เครื่องจักรทำให้ อย่างไร มีเหลือเฟือ — เอนจินในบทที่ 10 เดินได้แทบทุกพื้นที่ที่ระบุออกมาได้ และเก่งขึ้นทุกฤดูกาลที่หมึกบนบทนั้นแห้งลง การกระทำที่ขาดแคลน การกระทำที่กลไกทั้งชุดรออยู่ จึงคือการเลือกปัญหา การหาปัญหา (problem-finding) — รสนิยม — คือท่าสุดท้ายที่ยังไม่ถูกทำให้เป็นอัตโนมัติ รสนิยมสร้างด้วยวิธีเดียวกับทำอื่น: จดบันทึกเส้นทางไว้ สังเกตว่าปัญหาที่แก้แล้วข้อไหนทำให้โลกเบาลง และข้อไหนแค่ทำให้โลกถูกปรับให้เหมาะสมที่สุดขึ้นเฉยๆ ถ้ามกับปมทุกปมที่เข้าคิวมา ไม่ใช่ "แก้ได้ไหม" แต่ "ถ้าแก้แล้ว อะไรจะได้มาฟรี"

E N

So the book ends by inverting its own opening. At Gordium, the knot was handed to Alexander; the prophecy, the frame, the audience — all set by others, and his genius was refusal. You are past that now. From here on, more and more, you will be the one *handing out* knots — to your tools, your machines, your colleagues, your future self. That is what it means to hold the frame while the searching is delegated: the quality of the problems you set becomes the ceiling on everything downstream.

T H

หนังสือเล่มนี้จบด้วยการกลับด้านบทเปิดของตัวเอง ที่กอร์เดียม ปมถูกยื่นให้ Alexander คำพยากรณ์ กรอบ ผู้ชม — คนอื่นตั้งไว้ให้ทั้งหมด อัจฉริยภาพของเขาคือการปฏิเสธ คุณเลยจุดนั้นมาแล้ว นับจากนี้ คุณจะเป็ฝ่าย ยื่นปม มากขึ้นเรื่อยๆ — ให้เครื่องมือ ให้เครื่องจักร ให้เพื่อนร่วมงาน ให้ตัวเองในอนาคต นั่นคือความหมายของการถือกรอบไว้ขณะที่มอบการค้นหาลงไป: คุณภาพของปัญหาที่ตั้ง คือเพดานของทุกอย่างที่อยู่ปลายน้ำ

E N

The knot is the problem as handed to you. The path is yours to choose. Both halves of that sentence were always true; the route you have just walked exists to change which half you live in.

T H

ปมคือปัญหาในรูปที่ถูกยื่นมา เส้นทางเลือกเองได้ สองท่อนของประโยคนั้นจริงมาตลอด เส้นทางที่เพิ่งเดินจบไปมีอยู่เพื่อเปลี่ยนว่าจะใช้ชีวิตอยู่ในท่อนไหน

E N

Choose well. Walk short.

T H

เลือกให้ดี เดินให้สั้น

E N

THE SHORTEST PATH · end

T H

เส้นทางที่สั้นที่สุด · จบ



Great solvers do not search harder; they change the space until the path is short.

นักแก้ปัญหาที่ยิ่งใหญ่ไม่ค้นหาหนทางขึ้น — เขาเปลี่ยนพื้นที่จนเส้นทางสั้นลง

books/thinking/the-shortest-path/ · The AICE Library · 2026-08-02