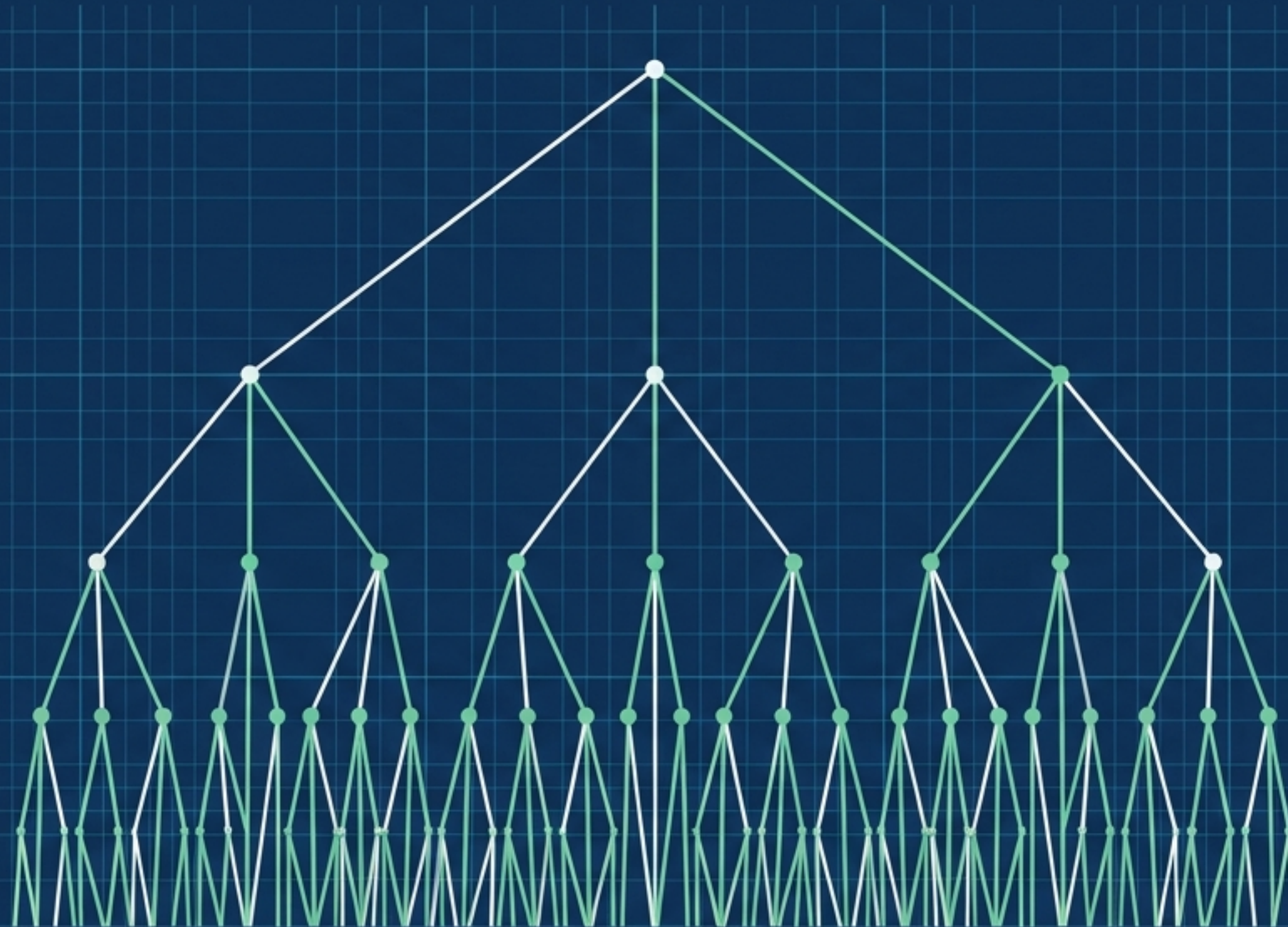


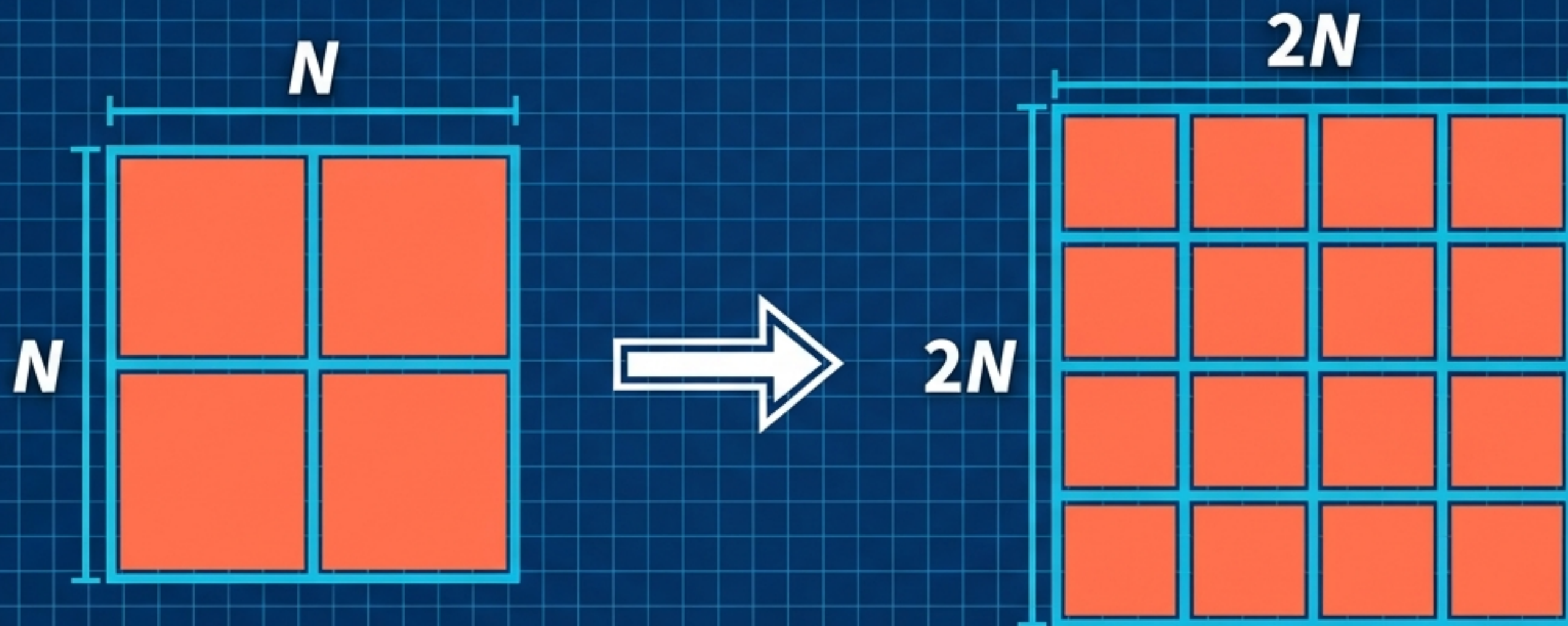
カラツバ法：人類がはじめて壁を越えた「掛け算」の魔法

12×34のパズルから読み解く、 $O(n^{1.585})$ への飛躍



何千年もの間、人類は「筆算」に縛られていた

N 桁の掛け算を行う場合、従来の筆算法ではすべての桁同士を掛け合わせる必要があり、計算量は $O(N^2)$ になります。



桁数が2倍になれば、計算時間は4倍になる

1960年、23歳の学生が「数学の常識」を覆す



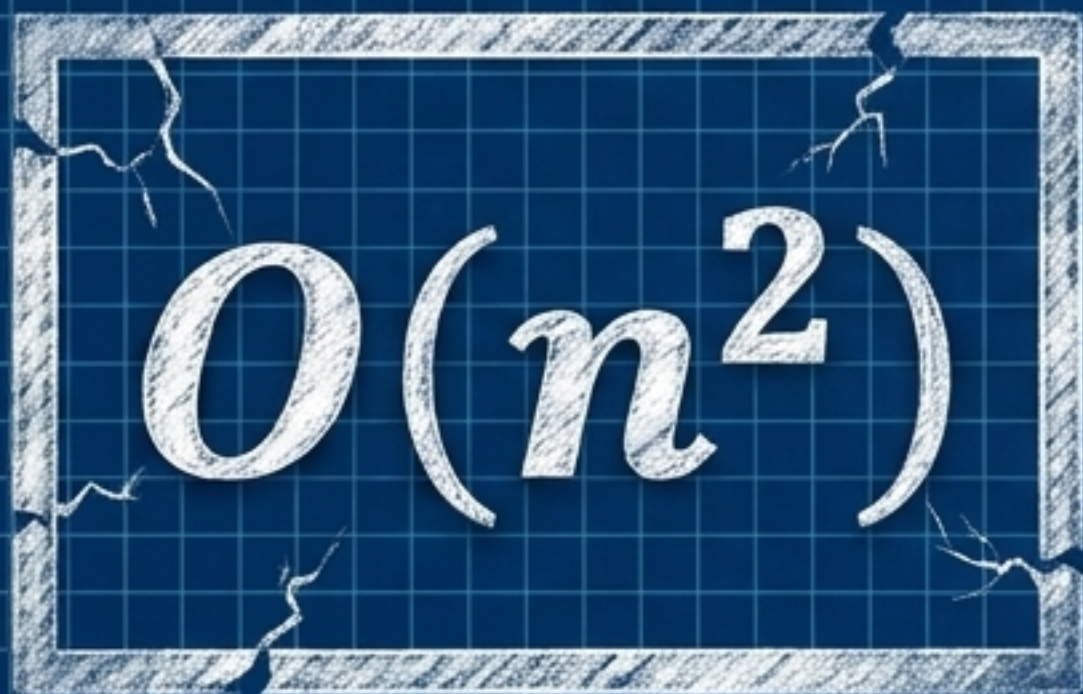
1960年
モスクワ大学

天才数学者コルモゴロフの予想：
「乗算に $O(n^2)$ より速いアル
ゴリズムは存在しない」



23歳の学生カラツバの反証：
わずか1週間で $O(n^{1.585})$ の
乗算アルゴリズムを発見。

わずか1週間後



$O(n^{1.585})$

掛け算の歴史が、この日動いた。

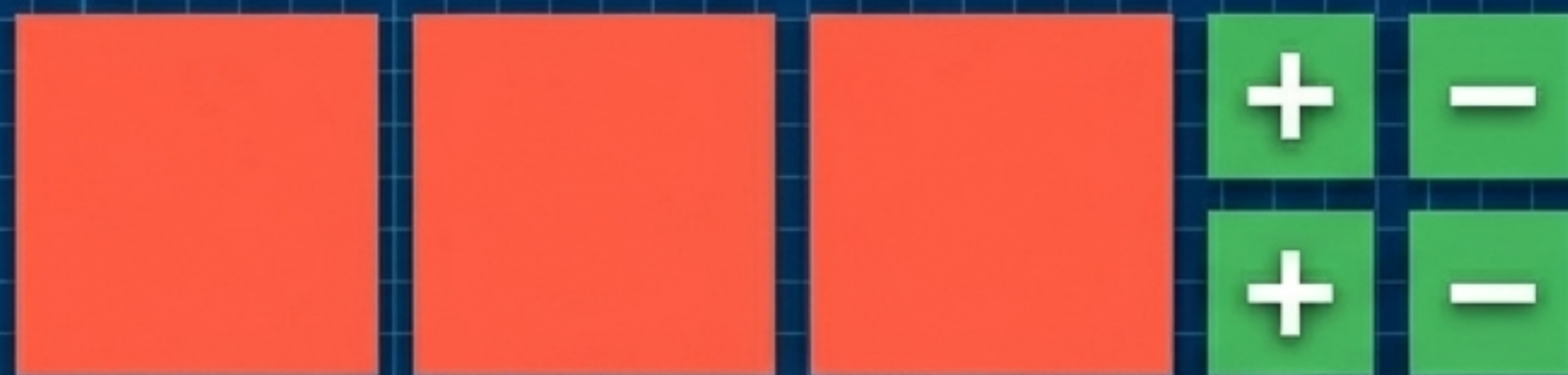
カラツバの魔法:「4回の掛け算」を「3回」に減らす

加算や減算のコストは $O(n)$ であり、巨大な数においては無視できるほど小さい。

従来の筆算



カラツバ法

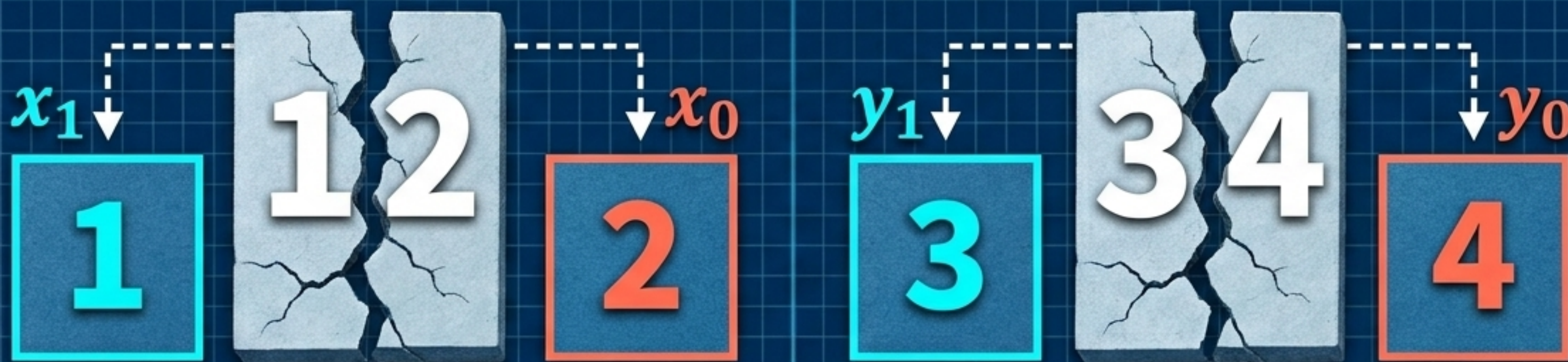


「重い乗算を1つ減らし、軽い加減算で補う」ことが、分割統治法 (Divide and Conquer) の鍵となる。

12×34

実践・タネ明かし： 12×34 を分解する

ステップ1：桁の分割



$$x_1 = 1 \quad x_0 = 2$$

$$y_1 = 3 \quad y_0 = 4$$

ステップ2：3つの「魔法の掛け算」

十の位と一の位をクロスして掛ける代わりに、足した合計同士を掛け合わせ、そこから不要な部分を引くことで乗算を1回減らす。

$$z_2 \quad \boxed{1} \times \boxed{3} \rightarrow 3$$

$$z_0 \quad \boxed{2} \times \boxed{4} \rightarrow 8$$

$$z_1 \quad (\boxed{1} + \boxed{2}) \times (\boxed{3} + \boxed{4}) - z_2 - z_0$$

(魔法の交差項)

$$3 \times 7 - 3 - 8 = 21 - 11 = 10$$

ステップ3：桁をシフトして再構築

求めた3つの値 (\$3, 10, 8\$)
に元の桁数に応じた重み
(\$100, 10, 1\$) を掛けて
足し合わせるだけ。

これで $12 \times 34 = 408$ が
完成！

$$z_2 \times 10^2$$

$$3 \times 100 \rightarrow 300$$

$$z_1 \times 10^1$$

$$10 \times 10 \rightarrow 100$$

$$z_0 \times 10^0$$

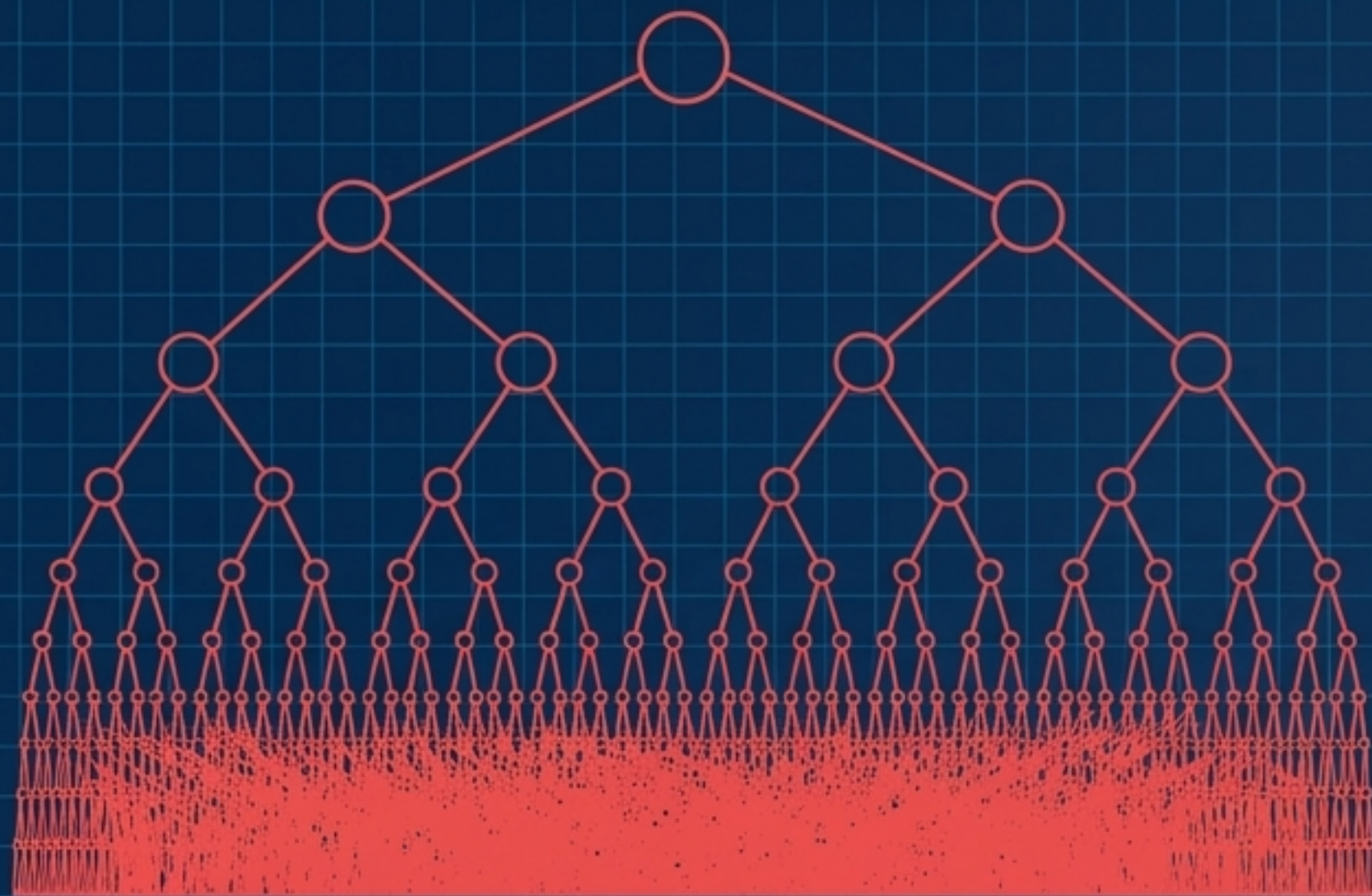
$$8 \times 1 \rightarrow 8$$

408

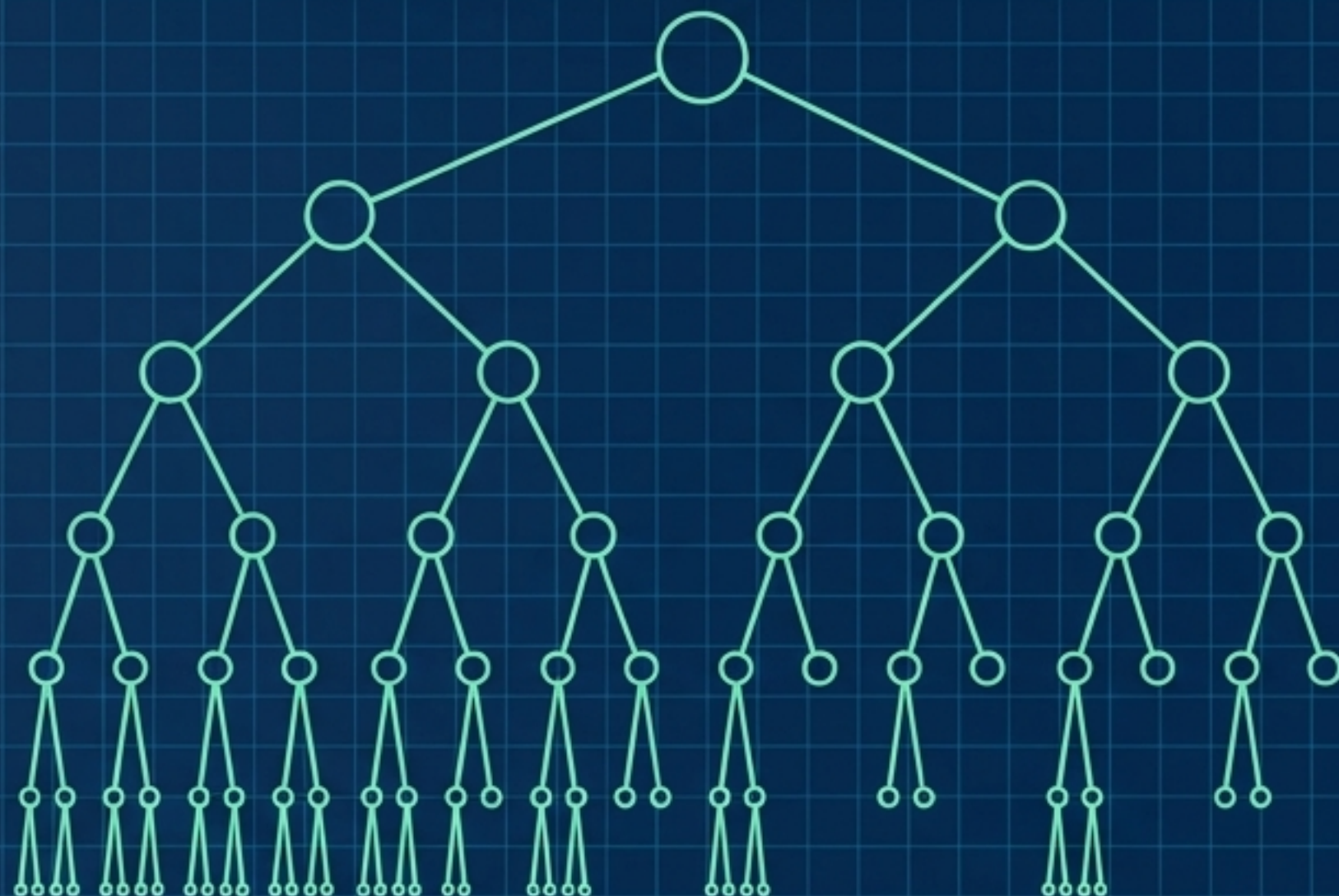
なぜ「1回減る」だけで世界が変わるのか？

カラツバ法の真の威力は「再帰（Recursion）」にあります。数千桁、数万桁の巨大な数の計算を分割し続けると、末端における乗算回数の差は圧倒的なものになります。

筆算法 $O(n^2)$



カラツバ法 $O(n^{1.585})$



【1024桁の計算時】 筆算の乗算：約**104万回** / カラツバ法の乗算：約**5.9万回**

オーダーの計算①：分割統治の漸化式

計算量 $T(N)$ は、自己相似的な漸化式として定義されます。
重要なのは、乗算が3回に減ったことで先頭の係数が「4」から「3」に落ちている点です。

$$T(2^n) = 3T(2^{n-1}) + 8 \cdot 2^n$$

分割された半分の
サイズの乗算3回分

加算やシフト処理に
かかるコスト ($O(N)$)

オーダーの計算②：再帰の底まで展開する

各階層における加算コストの総和は、公比 $2/3$ の等比数列の和となります。
これを解き進めると、計算量は支配項である $O(3^n)$ に帰着します。

$$T(2^n) = 3 T(2^{n-1}) + 8 \cdot 2^n$$

$$= 3(3 T(2^{n-2}) + 8 \cdot 2^{n-1}) + 8 \cdot 2^n$$

$$= 3^2 T(2^{n-2}) + (3 \cdot 8 \cdot 2^{n-1} + 8 \cdot 2^n)$$

...

$$3^n T(2^0) + 8(3^{n-1} \cdot 2 + 3^{n-2} \cdot 2^2 + \dots)$$

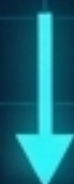


$$\rightarrow 17 \cdot 3^n - 16 \cdot 2^n$$

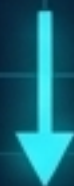
オーダーの計算③：対数 log がもたらす飛躍

「サイズが半分になるたびに、問題が3つに増える」。この関係性が数学的に処理されることで、 $O(N^2)$ だった掛け算の世界に ≈ 1.585 という魔法の数字がもたらされました。

$$T(2^n) = O(3^n)$$



$$N = 2^n \rightarrow n = \log_2 N$$



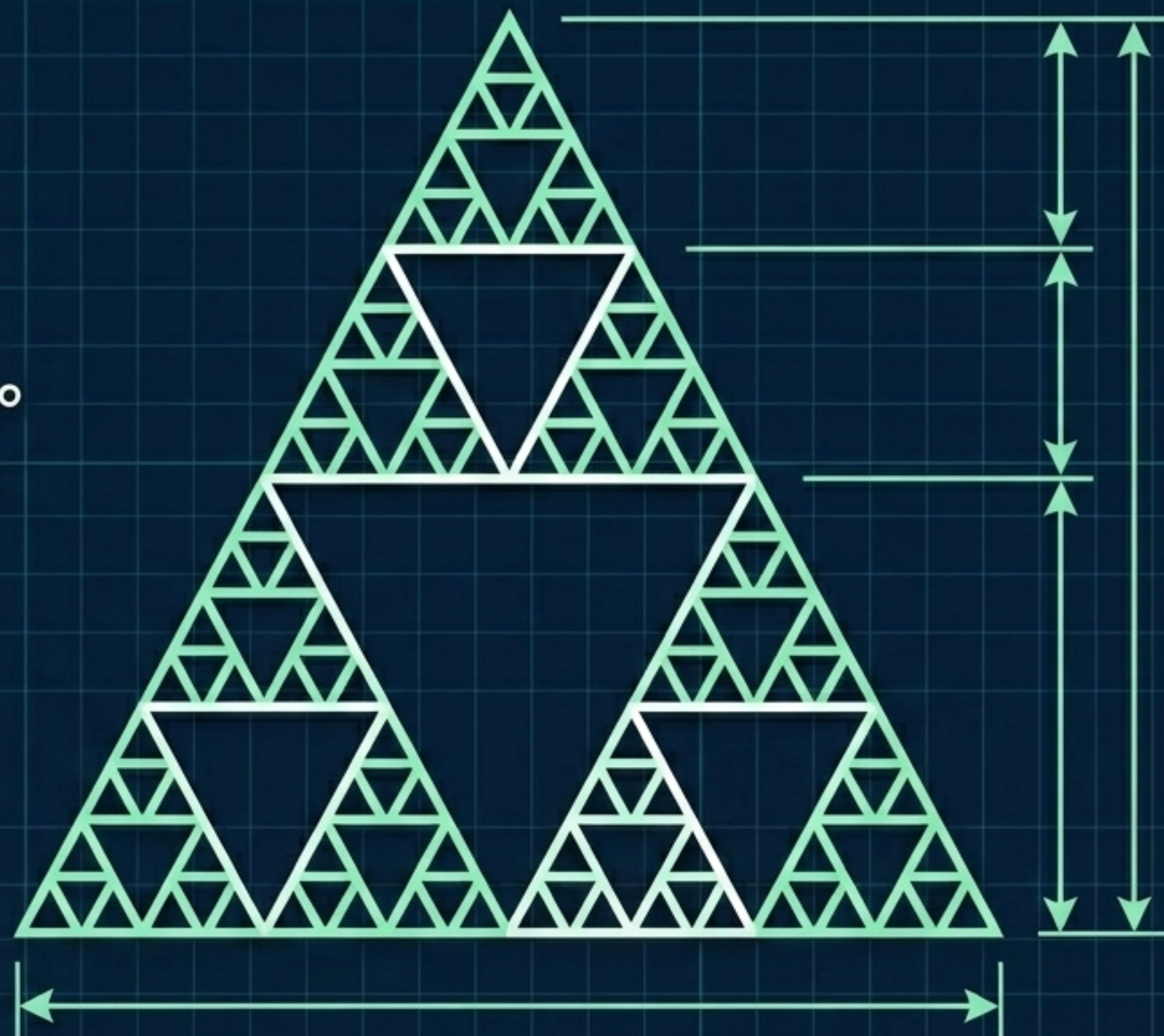
$$O(3^{\log_2 N}) = O(N^{\log_2 3})$$



$$\approx O(N^{1.585})$$

フラクタル図形に隠された 「1.585次元」

シェルピンスキーのギャスケットの
フラクタル次元 $d = \log_2 3 \approx 1.585$ 。
カラツバ法の計算構造は、この美しい
幾何学模様と完全に同じ次元を
持っています。

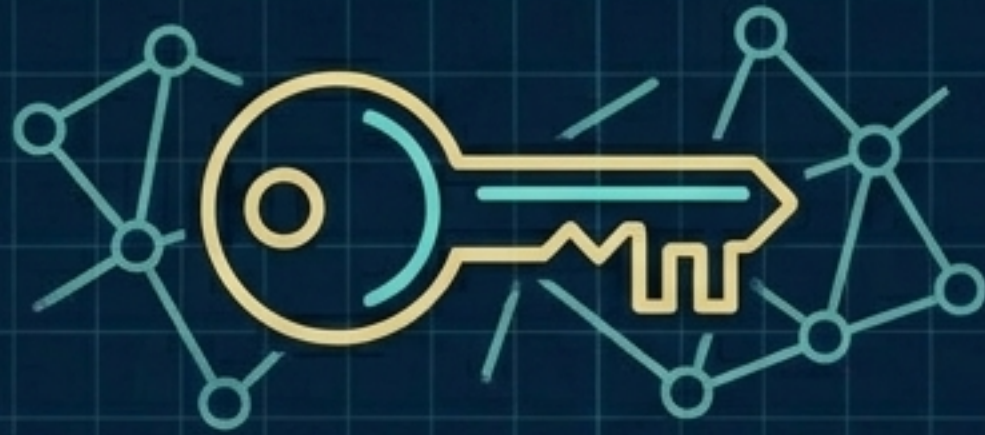


多倍長乗算の進化論：終わらない高速化の探求

カラツバ法は門をこじ開けたに過ぎません。その後、数学者たちは多項式補間や高速フーリエ変換（FFT）を利用し、計算量をさらに削ぎ落としていきました。

アルゴリズム	アプローチ	計算量（オーダー）
筆算法	-	$O(N^2)$
カラツバ法（1960）	2分割	$O(N^{1.585})$
Toom-Cook-3法 （1963）	3分割	$O(N^{1.465})$
Schönhage-Strassen法 （1971）	FFT利用	$O(N \log N \log \log N)$

現代技術を底辺で支える「分割統治」の思想



今日、数千・数万桁の巨大な整数を扱うRSA暗号や、円周率の天文学的計算などの根底には、あの23歳の学生が見出したアルゴリズムが脈々と息づいています。

数式を異なる視点から見つめ直す。
その小さな視点の変化が、計算機の歴史を大きく前進させる。