GRAD SEMINAR · NOVEMBER 2026

Multiplying integers below $n \log n$

How a machine-generated preprint cracked a 55-year-old conjecture, and how the community is checking it

Andrew V Sutherland · MIT

The question

How many bit operations $M(n)$ does it take to multiply two n-bit integers?

Trivially $n \lesssim M(n) \lesssim n^2$, the upper bound being the method we all learned in school. Everything interesting happens in between.

WHY CARE

Division, square roots, GCD, base conversion, polynomial arithmetic, computing π: almost everything is $M(n)$ times a small factor.

THE CATCH

The answer depends on what counts as an operation. At this precision, the model of computation is part of the theorem.

Sixty-five years of upper bounds

When	Who	Bound on M(n)
c. 1956	Kolmogorov	conjectures $\Theta(n^2)$ is optimal
1960	Karatsuba (with Ofman, 1962)	$O(n^{\log_2 3}) \approx n^{1.585}$
1963–66	Toom; Cook	$O(n^{1+\varepsilon})$ for every $\varepsilon > 0$
1971	Schönhage–Strassen	$O(n \log n \log\log n)$; conjecture $\Omega(n \log n)$
2007	Fürer	$O(n \log n \cdot K^{\log^* n})$ for some K
2014–18	Harvey–vdH–Lecerf; Covanov–Thomé	explicit K: $8 \to 6 \to 4\sqrt{2} \to 4$
2019	Harvey–van der Hoeven (Annals 2021)	$O(n \log n)$, matching the conjecture
Sept 2026	OpenAI, preprint #109	$O(n (\log n)^{1-\kappa})$, $\kappa = 2^{-182}$

$\log^* n \le 5$ for every $n < 2^{65536}$, so the post-Fürer factor $K^{\log^* n}$ was only ever about bookkeeping; the 2019 result removed it.

Papers: Karatsuba–Ofman · Toom · Cook · Schönhage–Strassen · Fürer · Harvey–vdH–Lecerf · Covanov–Thomé · Harvey–van der Hoeven · OpenAI #109

Why *n* log *n* looked like the floor

THE FFT

$\log_2 N$ layers of N/2 butterflies

In Harvey–van der Hoeven the n log n term *is* the butterfly count. Harvey put the odds of beating that arithmetic at “precisely zero.”

RESTRICTED LOWER BOUNDS

Proved, but in weaker models

Morgenstern (1973): linear circuits with bounded coefficients need ½N $\log_2 N$ additions for the DFT. Online multiplication on multitape machines: ≳ n log n (Cook–Aanderaa; Paterson–Fischer–Meyer).

CONDITIONAL

Network coding

Afshani–Freksen–Kamma–Larsen (2019): Boolean circuits need Ω(n log n) *if* the Li–Li network coding conjecture holds. The new algorithm, if correct, refutes Li–Li.

Which machine are we counting on?

GOLD STANDARD

Multitape Turing machine

Fixed finite alphabet, fixed number of tapes, heads move one cell per step. Moving data costs time. Schönhage–Strassen, Harvey–van der Hoeven and the OpenAI paper all state their bounds here.

ARITHMETIC ONLY

Boolean circuits

Count gates; wires are free, so rearranging data costs nothing. Harvey: the circuit and Turing exponents need not agree. Multiplication is at least as hard as matrix transposition on a TM (HvdH 2025).

TOO GENEROUS

RAM and pointer machines

One unit-cost lookup moves a log n-bit word: exactly the factor we are fighting over. Schönhage (1980): storage modification machines multiply in **linear** time.

THEOREM (CLAIMED) · OPENAI, "INTEGER MULTIPLICATION BELOW N LOG N," DATED 23 SEPT, RELEASED 6 OCT 2026

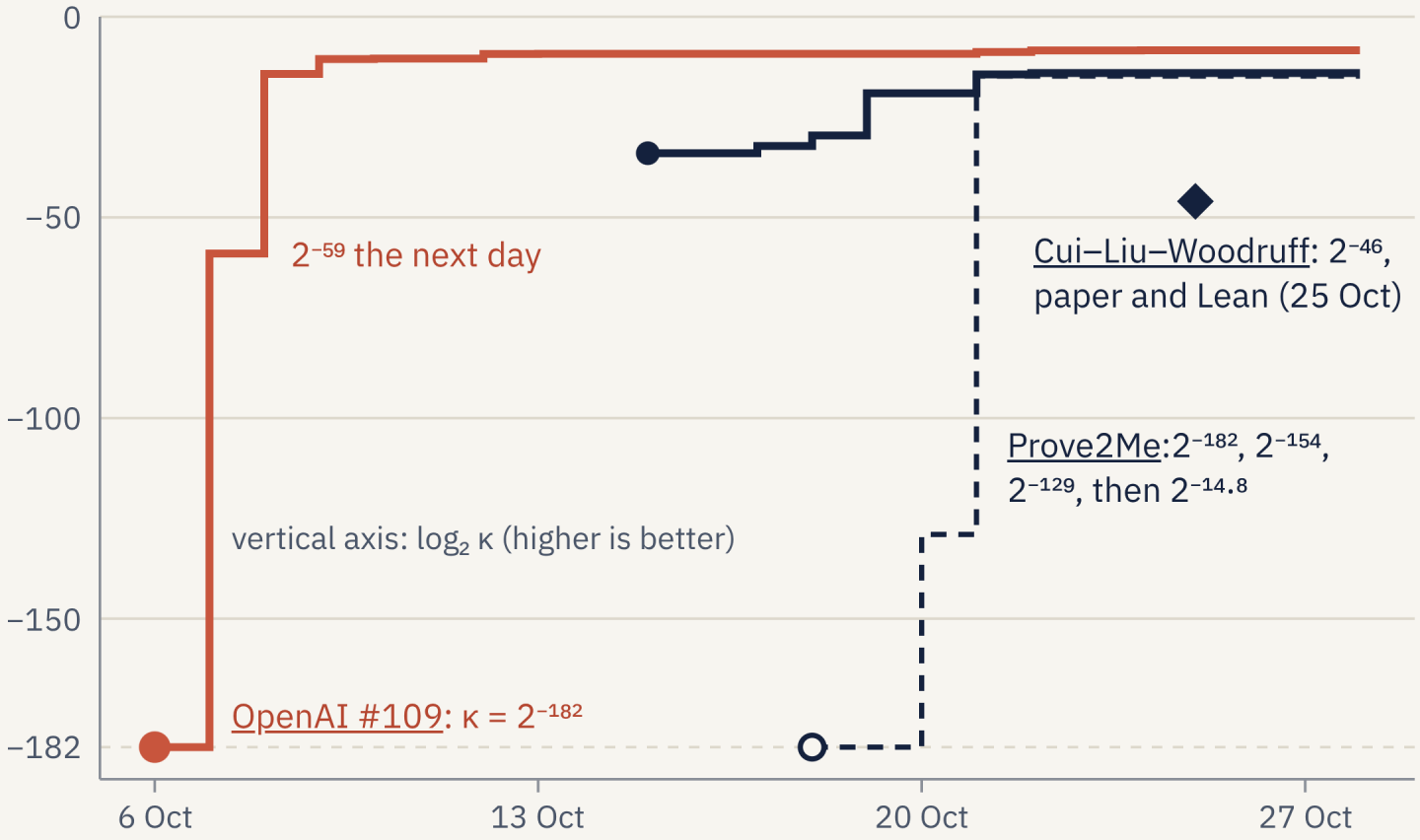
A fixed deterministic multitape Turing machine multiplies two n-bit integers in time $O(n\,(\log n)^{(1-\kappa)})$, with $\kappa = 2^{-1882}$.

The saving $(\log n)^{\kappa}$ exceeds 2 only when $\log_2 \log_2 n > 2^{1882}$. No practical consequence whatsoever.

Yet as Harvey stresses, a power of log n dwarfs Fürer's gain over Schönhage–Strassen, which was less than log log n.

Result #109 of 722 AI-generated manuscripts. Companion claim, with Lean from OpenAI: the exact DFT in $O(N\,(\log N)^{(1-\delta)})$ operations.

Then the internet joined in: κ in three weeks



Community, conditional

$\kappa \approx 3.0 \times 10^{-3} \approx 2^{-8.4}$ (24 Oct). Assumes the #109 framework plus five stated hypotheses.

Lean, end to end

$\kappa \approx 6.0 \times 10^{-5} \approx 2^{-14}$ (22 Oct), on Mathlib's Turing machines.

Prove2Me campaign (Lean)

$\kappa \geq 3.6 \times 10^{-5}$ (21 Oct), multitape TM statement.

Live tracker: beyond-n-log-n.netlify.app

Is it true? The evidence on 3 November

HUMAN EXPERT · 20 OCT

Harvey: "plausible"

Checked the core gadget and the amplification carefully. Not yet: synthetic roots on r×⋯×r arrays, growing dimension d, Turing data movement.

LEAN · FROM 15 OCT

platypii & jdb19937

End-to-end theorem for one machine in Mathlib's Turing.FinTM2, standard axioms only; $\kappa \approx 6.0\times10^{-5}$. Not yet rebuilt by an independent party.

LEAN · 21 OCT

Prove2Me campaign

Statement for a fixed multitape TM; marked solved with $\kappa \geq 3.6\times10^{-5}$, after missions at 2^{-182}, 2^{-154}, 2^{-129}.

PAPER + LEAN · 25 OCT

Cui–Liu–Woodruff

Google and CMU authors: a simpler algorithm, $\kappa = 2^{-46}$, with written proofs and a formal statement they report checking. Their Lean assumes nothing from #109.

Caveats: all three Lean developments were written largely by AI agents, for descendants or simplifications of the construction; no outside rebuild or audit found. The formal statements are what need an independent audit. Nothing is refereed: no arXiv, Hexagon or Palomar entry found; OpenAI's repository still has no Lean for #109.

The idea in one slide

1 · COUNT

A DFT on $(\mathbb{C}^2)^{\otimes m}$ applies the butterfly $B = (1\ 1;\ 1\ -1)$ along each of the m axes: m "slice calls." In HvdH this count *is* the n log n.

2 · BATCH

Find a fixed gadget transforming W arrays with $\rho \cdot W_m$ slice calls, $\rho < 1$, mixing arrays by additions between calls. Like Strassen: n matrix–vector products cost less than n^3.

3 · AMPLIFY

Substitute the problem into the gadget recursively: $\log_m \log N$ levels each save ρ, so the cost is $N (\log N)^{(1-\kappa)}$ with $\kappa = \log(1/\rho) / \log m$.

WHY A GADGET EXISTS

Rescale B to C with eigenvalues 1 and i. In the Hadamard basis the butterfly along any $v \in \mathbb{F}_2^m$ is $\mathrm{diag}(i^{(v \cdot x)})$: all commute, and weight mod 4 is additive on orthogonal vectors. So partial progress is a nondegenerate subspace of $\mathbb{F}_2^m$, and arrays at different stages can still be combined.

BACKUP

The gadget, with registers and subspaces

$C = aI + bX$, $a = (1+i)/2$, $b = (1-i)/2$; for $v \in \mathbb{F}_2^m$, $C_v = aI + bX_v$ is "C in direction v."

For nondegenerate $U \subseteq \mathbb{F}_2^m$ put $\Phi_U = H^{\otimes m} \cdot \mathrm{diag}(i^{wt(P_U x)}) \cdot H^{-\otimes m}$. Then $\Phi_0 = I$, $\Phi_{all} = C^{\otimes m}$, $\Phi_{U \oplus V} = \Phi_U \Phi_V$, and adding an odd-weight $v \perp U$ costs one slice call.

A register stores $\Phi_U f$; two registers may be added only when their U agree.

Baby program on $(\mathbb{C}^2)^{\otimes h}$: data registers x_T start at $\langle v_T \rangle$ for 3-sets $T \subset [h]$; key fact $v_S \cdot v_T = 0$ when $|S \cap T| \in \{0, 2\}$. Slice calls: $Wh - 2 \cdot C(h,3) + 2(h^2 + h)$.

Main program: $m = h^3$, three stages over a $C(h,3)^3$ cube of data registers. Total $Wm - |T|^2(2|T| - 6h(h+1)) < Wm$ once $h \geq 22$.

Harvey checked this by hand; Claude Fable ran the baby program numerically for small h.

Source: D. Harvey, "On the recently proposed o(n log n) integer multiplication algorithm," §7 (20 Oct 2026)

What changes, and what is open

A new constant to chase

α = inf{a : M(n) = O(n (log n)^a)}. Now α < 1. Within this framework Gaussian resampling blocks α < 1/2.

Models may now diverge

Is the circuit exponent the Turing exponent? Lower bounds for bounded-coefficient and online models still stand.

Knock-on effects

The Li–Li conjecture falls; 𝔽_p[x] multiplication likely improves conditionally; the DFT itself is claimed below N log N.

How mathematics gets done

The leading expert read it through an LLM; agents formalized it within two weeks; GitHub leaderboards replaced the preprint. Within three weeks a Google DeepMind–CMU team posted a simpler algorithm with its own Lean proof.

Practical impact: none. You may keep writing O(n log n) in your papers.

Where to read more

- OpenAI, Integer multiplication below n log n: github.com/openai/math, preprint #109
- D. Harvey, On the recently proposed o(n log n) integer multiplication algorithm: web.maths.unsw.edu.au/~davidharvey/research/subnlogn.pdf
- P. Cui, Y. Liu, D. P. Woodruff, A simpler algorithm for integer multiplication below n log n, with a formally verified proof: github.com/lockqueue/intmul
- D. Harvey, J. van der Hoeven, Integer multiplication in time O(n log n), Ann. of Math. 193 (2021)
- D. Harvey, J. van der Hoeven, Integer multiplication is at least as hard as matrix transposition, arXiv:2503.22848
- Conditional bounds: github.com/CrocSwap/integer-mult-bounds; tracker beyond-n-log-n.netlify.app
- Lean: github.com/platypii/integer-mult-bounds-lean; prove2.me/campaigns/integer-multiplication-kappa
- Status board: distributedagenthub.xyz (DARC)
- Hexagon, a repository for AI-assisted results: hexagonmath.org
- Palomar, a registry of Lean-verified results: palomar-registry.org