Integer multiplication is at least as hard as matrix transposition

Joris van der Hoeven

Joint work with David Harvey

CNRS, École polytechnique, France

Funded by the European Union

European Research Council

Established by the European Commission

AIMS conference, Athens

July 10, 2026

Complexity of integer multiplication

What is the complexity of N-bit integer multiplication on a Turing machine?

<4000 BC	Mesopotamia	$O(N^2)$

Complexity of integer multiplication

What is the complexity of N-bit integer multiplication on a Turing machine?

<4000 BC	Mesopotamia	$O(N^2)$
1962	Karatsuba	$O(N^{\log 3/\log 2})$

Complexity of integer multiplication

What is the complexity of N-bit integer multiplication on a Turing machine?

<4000 BC	Mesopotamia	$O(N^2)$
1962	Karatsuba	$O(N^{\log 3/\log 2})$
	⋮	
1971	Schönhage-Strassen	$O(N \log N \log\log N)$

Complexity of integer multiplication

What is the complexity of N-bit integer multiplication on a Turing machine?

<4000 BC	Mesopotamia	$O(N^2)$
1962	Karatsuba	$O(N^{\log 3/\log 2})$
	$\vdots$	
1971	Schönhage-Strassen	$O(N \log N \log \log N)$
2007	Fürer	$O(N \log N \, 2^{O(\log^* N)})$

Complexity of integer multiplication

What is the complexity of N-bit integer multiplication on a Turing machine?

<4000 BC	Mesopotamia	$O(N^2)$
1962	Karatsuba	$O(N^{\log 3/\log 2})$
	$\vdots$	
1971	Schönhage-Strassen	$O(N \log N \log \log N)$
2007	Fürer	$O(N \log N\, 2^{O(\log^* N)})$

$$\log^* N := \min\{\ell \in \mathbb{N} : \log \circ \overset{\ell \times}{\cdots} \circ \log N \leqslant 1\}$$

Complexity of integer multiplication

What is the complexity of N-bit integer multiplication on a Turing machine?

<4000 BC	Mesopotamia	$O(N^2)$
1962	Karatsuba	$O(N^{\log 3/\log 2})$
	$\vdots$	
1971	Schönhage-Strassen	$O(N \log N \log\log N)$
2007	Fürer	$O(N \log N\, 2^{O(\log^* N)})$
2014	Harvey-vdH-Lecerf	$O(N \log N\, 8^{\log^* N})$

Complexity of integer multiplication

What is the complexity of N-bit integer multiplication on a Turing machine?

<4000 BC	Mesopotamia	$O(N^2)$
1962	Karatsuba	$O(N^{\log 3/\log 2})$
	$\vdots$	
1971	Schönhage-Strassen	$O(N \log N \log\log N)$
2007	Fürer	$O(N \log N\, 2^{O(\log^* N)})$
2014	Harvey-vdH-Lecerf	$O(N \log N\, 8^{\log^* N})$
	$\vdots$	
2019	Harvey-vdH	$O(N \log N)$

Complexity of integer multiplication

What is the complexity of N-bit integer multiplication on a Turing machine?

<4000 BC	Mesopotamia	$O(N^2)$
1962	Karatsuba	$O(N^{\log 3/\log 2})$
	⋮	
1971	Schönhage-Strassen	$O(N \log N \log\log N)$
2007	Fürer	$O(N \log N\, 2^{O(\log^* N)})$
2014	Harvey-vdH-Lecerf	$O(N \log N\, 8^{\log^* N})$
	⋮	
2019	Harvey-vdH	$O(N \log N)$

Question: is $N \log N$ also a lower bound for the complexity of multiplication?

Matrix transposition on a Turing machine

$$\begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 \\ c_0 & c_1 & c_2 & c_3 \\ d_0 & d_1 & d_2 & d_3 \end{pmatrix}$$

Matrix transposition on a Turing machine

$$\begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 \\ c_0 & c_1 & c_2 & c_3 \\ d_0 & d_1 & d_2 & d_3 \end{pmatrix}$$

a_0	a_1	a_2	a_3	b_0	b_1	b_2	b_3	c_0	c_1	c_2	c_3	d_0	d_1	d_2	d_3

Matrix transposition on a Turing machine

$$\begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 \\ c_0 & c_1 & c_2 & c_3 \\ d_0 & d_1 & d_2 & d_3 \end{pmatrix}$$

a_0	a_1	a_2	a_3	b_0	b_1	b_2	b_3	c_0	c_1	c_2	c_3	d_0	d_1	d_2	d_3

Matrix transposition on a Turing machine

$$\begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 \\ c_0 & c_1 & c_2 & c_3 \\ d_0 & d_1 & d_2 & d_3 \end{pmatrix}$$

a_0	b_0	c_0	d_0	a_1	b_1	c_1	d_1	a_2	b_2	c_2	d_2	a_3	b_3	c_3	d_3	

a_0	a_1	b_0	b_1	c_0	c_1	d_0	d_1	a_2	a_3	b_2	b_3	c_2	c_3	d_2	d_3	

In general: $r \times c$ transposition $\longrightarrow O(rc \lg \min(r,c))$ operations

Folklore algorithm. $O(n^2 \lg n)$

Complexity of $n \times n$ matrix transposition

Folklore algorithm. $O(n^2 \lg n)$

Stoß (1974). Matching lower bound in variant of Turing model for which we are only allowed to *move* data (and not to perform actual computations).

Complexity of $n \times n$ matrix transposition

Folklore algorithm. $O(n^2 \lg n)$

Stoß (1974). Matching lower bound in variant of Turing model for which we are only allowed to *move* data (and not to perform actual computations).

Kirchherr (1988), Dietzfelbinger–Maass (1993). Lower bounds for Turing machines with a single tape.

Complexity of $n \times n$ matrix transposition

Folklore algorithm. $O(n^2 \lg n)$

Stoß (1974). Matching lower bound in variant of Turing model for which we are only allowed to *move* data (and not to perform actual computations).

Kirchherr (1988), Dietzfelbinger–Maass (1993). Lower bounds for Turing machines with a single tape.

Warning. We implicitly assumed 1-bit entries.
For b-bit entries, the complexity becomes $O(b n^2 \lg n)$.

The main theorem

$\mathsf{M}(N)$ complexity of integer multiplication
$\mathsf{T}(N)$ complexity of $r \times c$ matrix transposition with b-bit entries and $brc \leqslant N$.

Theorem

If the conjectured lower bound

$$\mathsf{T}(N) = \Omega(N \log N)$$

holds for transposition, then so does the one for multiplication:

$$\mathsf{M}(N) = \Omega(N \log N).$$

Bluestein: DFT $\longrightarrow$ convolution

$\zeta_n := \mathrm{e}^{\frac{2\pi\mathrm{i}}{n}}, \quad X \in \mathbb{C}^n$

$$Y_t \;:=\; \mathrm{DFT}_n(X)_t \;:=\; \frac{1}{n}\sum_{0\leqslant s<n} \zeta_n^{-st} X_s$$

$$X_s \;=\; \mathrm{DFT}_n^{-1}(Y)_s \;=\; \sum_{0\leqslant t<n} \zeta_n^{st} Y_t$$

Bluestein: DFT $\longrightarrow$ convolution

$\zeta_n := \mathrm{e}^{\frac{2\pi \mathrm{i}}{n}}, \quad X \in \mathbb{C}^n$

$$Y_t \;:=\; \mathrm{DFT}_n(X)_t \;:=\; \frac{1}{n} \sum_{0 \leqslant s < n} \zeta_n^{-st} X_s$$

$$X_s \;=\; \mathrm{DFT}_n^{-1}(Y)_s \;=\; \sum_{0 \leqslant t < n} \zeta_n^{st} Y_t$$

Setting $\omega_k := \zeta_n^{\binom{k}{2}}$, we have

$$Y_t \;=\; \frac{1}{n} \bar{\omega}_t \sum_{0 \leqslant s < n} (\bar{\omega}_{-s} X_s)\, \omega_{t-s}$$

DFT of length n $\xrightarrow{\text{reduces}}$ product of $\tilde{X}, \Omega \in \mathbb{C}[u]$ of degrees $< n$, with Ω fixed

Kronecker: convolution $\longrightarrow$ integer multiplication

DFT: approximated using b-bit fixed point arithmetic

Reduces to arithmetic on b-bit integer coefficients

Kronecker: convolution $\longrightarrow$ integer multiplication

DFT: approximated using b-bit fixed point arithmetic

Reduces to arithmetic on b-bit integer coefficients

$$\tilde{X} = 12u^2 + 85u + 27, \qquad \Omega = 33u^2 + 42u + 11$$

Kronecker: convolution $\longrightarrow$ integer multiplication

DFT: approximated using b-bit fixed point arithmetic

Reduces to arithmetic on b-bit integer coefficients

$$\tilde{X} = 12u^2 + 85u + 27, \qquad \Omega = 33u^2 + 42u + 11$$

$$\tilde{X}(10^5) = 120008500027, \qquad \Omega(10^5) = 330004200011$$

Kronecker: convolution $\longrightarrow$ integer multiplication

DFT: approximated using b-bit fixed point arithmetic

Reduces to arithmetic on b-bit integer coefficients

$$\tilde{X} = 12\,u^2 + 85\,u + 27, \qquad \Omega = 33\,u^2 + 42\,u + 11$$

$$\tilde{X}(10^5) = 1200008500023, \qquad \Omega(10^5) = 3300004200011$$

$$(\tilde{X}\Omega)(10^5) = 396033090446101901000253$$

Kronecker: convolution $\longrightarrow$ integer multiplication

DFT: approximated using b-bit fixed point arithmetic

Reduces to arithmetic on b-bit integer coefficients

$$\tilde{X} = 12\,u^2 + 85\,u + 27, \qquad \Omega = 33\,u^2 + 42\,u + 11$$

$$\tilde{X}(10^5) = 120008500027, \qquad \Omega(10^5) = 330004200011$$

$$(\tilde{X}\Omega)(10^5) = 396033090446101901000253$$

$$\tilde{X}\Omega = 396\,u^4 + 3309\,u^3 + 4461\,u^2 + 1901\,u + 253$$

Kronecker: convolution $\longrightarrow$ integer multiplication

DFT: approximated using b-bit fixed point arithmetic
Reduces to arithmetic on b-bit integer coefficients

$$\tilde{X} = 12u^2 + 85u + 27, \qquad \Omega = 33u^2 + 42u + 11$$

$$\tilde{X}(10^5) = 1200085000 23, \qquad \Omega(10^5) = 330004200011$$

$$(\tilde{X}\Omega)(10^5) = 3960330904461019010 0253$$

$$\tilde{X}\Omega = 396u^4 + 3309u^3 + 4461u^2 + 1901u + 253$$

Lemma

If $\lg n = O(b)$, $\lg^2 b = O(n)$, *then computing a b-bit DFT of length n takes time*

$$O(\mathsf{M}(nb) + n\,\mathsf{M}(b)).$$

Sliced DFTs

$$\begin{array}{ccc} X_0^{[0]},\ldots,X_0^{[l-1]}, & \cdots & ,X_{n-1}^{[0]},\ldots,X_{n-1}^{[l-1]} \\ & \downarrow & \\ Y_0^{[0]},\ldots,Y_0^{[l-1]}, & \cdots & ,Y_{n-1}^{[0]},\ldots,Y_{n-1}^{[l-1]} \end{array}$$

$$Y^{[i]} = \mathrm{DFT}(X^{[i]})$$

Sliced DFTs

$$
\begin{array}{ccc}
X_0^{[0]}, \ldots, X_0^{[l-1]}, & \ldots & , X_{n-1}^{[0]}, \ldots, X_{n-1}^{[l-1]} \\
& \downarrow & \\
Y_0^{[0]}, \ldots, Y_0^{[l-1]}, & \ldots & , Y_{n-1}^{[0]}, \ldots, Y_{n-1}^{[l-1]}
\end{array}
\qquad Y^{[i]} = \mathrm{DFT}(X^{[i]})
$$

$$Y^{[i]}(u) = \tilde{X}^{[i]}(u)\,\Omega(u)$$

Sliced DFTs

$$X_0^{[0]}, \ldots, X_0^{[l-1]}, \quad \ldots \quad , X_{n-1}^{[0]}, \ldots, X_{n-1}^{[l-1]}$$

$$\downarrow$$

$$Y_0^{[0]}, \ldots, Y_0^{[l-1]}, \quad \ldots \quad , Y_{n-1}^{[0]}, \ldots, Y_{n-1}^{[l-1]}$$

$$Y^{[i]} = \mathrm{DFT}(X^{[i]})$$

$$Y^{[i]}(u) = \tilde{X}^{[i]}(u)\,\Omega(u)$$

$$Y^{[0]}(u) + \cdots + Y^{[l-1]}(u)\,u^{2n(l-1)} = (\tilde{X}^{[0]}(u) + \cdots + \tilde{X}^{[l-1]}(u)\,u^{2n(l-1)})\,\Omega(u)$$

Sliced DFTs

$$\begin{array}{ccc} X_0^{[0]},\ldots,X_0^{[l-1]}, & \cdots & ,X_{n-1}^{[0]},\ldots,X_{n-1}^{[l-1]} \\ & \downarrow & \\ Y_0^{[0]},\ldots,Y_0^{[l-1]}, & \cdots & ,Y_{n-1}^{[0]},\ldots,Y_{n-1}^{[l-1]} \end{array} \qquad Y^{[i]} = \mathrm{DFT}(X^{[i]})$$

$$Y^{[i]}(u) = \tilde{X}^{[i]}(u)\,\Omega(u)$$

$$Y^{[0]}(u) + \cdots + Y^{[l-1]}(u)\,u^{2n(l-1)} = (\tilde{X}^{[0]}(u) + \cdots + \tilde{X}^{[l-1]}(u)\,u^{2n(l-1)})\,\Omega(u)$$

Lemma

If $\lg n = O(b)$ *and* $\lg^2 b = O(n)$*, then computing* l *sliced* b*-bit DFTs of length* n *can be done in time*

$$O(\mathsf{M}(l\,n\,b) + l\,n\,\mathsf{M}(b)).$$

$n_1 \times n_2$ transposition via DFTs

$$\begin{array}{cl} X_0, \ldots, X_{n_1 n_2 - 1} & \\ \downarrow & \mathrm{DFT}_{n_1 n_2} \\ Y_0, \ldots, Y_{n_1 n_2 - 1} & \\ \downarrow & \mathrm{DFT}^{-1}_{n_1, n_2} \\ X_0, \ldots, X_{(n_1-1) n_2}, \ldots, X_{n_2 - 1}, \ldots, X_{n_1 n_2 - 1} & \end{array}$$

$n_1 \times n_2$ transposition via DFTs

$$X_0, \ldots, X_{n_1 n_2 - 1}$$
$$\downarrow \qquad \mathrm{DFT}_{n_1 n_2}$$
$$Y_0, \ldots, Y_{n_1 n_2 - 1}$$
$$\downarrow \qquad \mathrm{DFT}^{-1}_{n_1, n_2}$$
$$X_0, \ldots, X_{(n_1-1)n_2}, \ldots, X_{n_2-1}, \ldots, X_{n_1 n_2 - 1}$$

$$\mathrm{DFT}^{-1}_{n_1,n_2} \overset{\text{Cooley-Tukey}}{=\!=\!=\!=} (\mathrm{Id}_{n_2} \otimes \mathrm{DFT}^{-1}_{n_1}) \circ \mathrm{Twiddle} \circ (\mathrm{DFT}^{-1}_{n_2} \otimes \mathrm{Id}_{n_1})$$

$n_1 \times n_2$ transposition via DFTs

$$
\begin{array}{ccc}
X_0, \ldots, X_{n_1 n_2 - 1} & & \\
\downarrow & & \mathrm{DFT}_{n_1 n_2} \\
Y_0, \ldots, Y_{n_1 n_2 - 1} & & \\
\downarrow & & \mathrm{DFT}^{-1}_{n_1, n_2} \\
X_0, \ldots, X_{(n_1-1)n_2}, \ldots, X_{n_2-1}, \ldots, X_{n_1 n_2 - 1} & &
\end{array}
$$

$$
\mathrm{DFT}^{-1}_{n_1,n_2} \overset{\text{Cooley-Tukey}}{=\!=\!=} \big(\underbrace{\mathrm{Id}_{n_2} \otimes \mathrm{DFT}^{-1}_{n_1}}_{\text{repeated DFTs}}\big) \circ \underbrace{\mathrm{Twiddle}}_{\text{linear pass}} \circ \big(\underbrace{\mathrm{DFT}^{-1}_{n_2} \otimes \mathrm{Id}_{n_1}}_{\text{sliced DFT}}\big)
$$

$n_1 \times n_2$ transposition via DFTs

$$\begin{array}{ccc} X_0, \ldots, X_{n_1 n_2 - 1} & & \\ \downarrow & & \mathrm{DFT}_{n_1 n_2} \\ Y_0, \ldots, Y_{n_1 n_2 - 1} & & \\ \downarrow & & \mathrm{DFT}^{-1}_{n_1, n_2} \\ X_0, \ldots, X_{(n_1-1)n_2}, \ldots, X_{n_2-1}, \ldots, X_{n_1 n_2 - 1} & & \end{array}$$

$$\mathrm{DFT}^{-1}_{n_1,n_2} \overset{\text{Cooley-Tukey}}{=\!=\!=} \underbrace{(\mathrm{Id}_{n_2} \otimes \mathrm{DFT}^{-1}_{n_1})}_{\text{repeated DFTs}} \circ \underbrace{\mathrm{Twiddle}}_{\text{linear pass}} \circ \underbrace{(\mathrm{DFT}^{-1}_{n_2} \otimes \mathrm{Id}_{n_1})}_{\text{sliced DFT}}$$

Theorem

$n = n_1 n_2$, $\lg n = O(b)$, $\lg^2 b = O(n)$ $\Rightarrow$ $n_1 \times n_2$ *transposition* b*-bit entries take time*

$$O(\mathsf{M}(n\,b) + n\,\mathsf{M}(b)).$$

Exponential reduction of b, the bit-size of entries

$n_1 \times n_2$ transposition, 1-bit entries, $\qquad \log n_1 \approx \log n_2$

Exponential reduction of b, the bit-size of entries

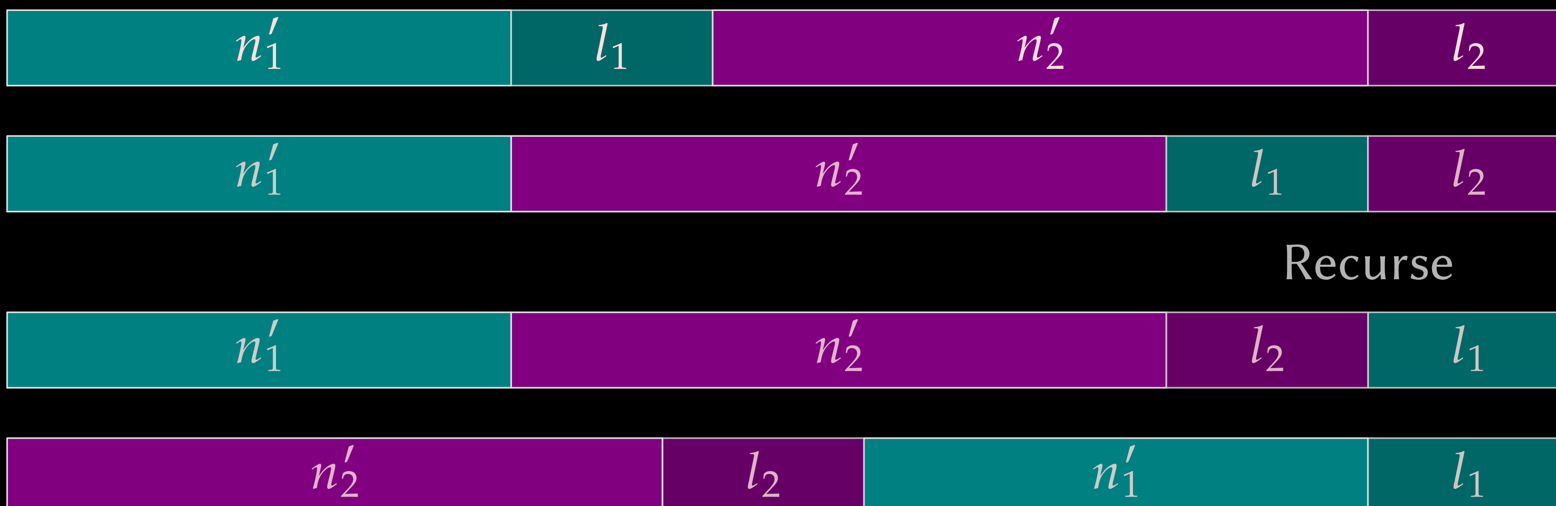
$n_1 \times n_2$ transposition, 1-bit entries, $\log n_1 \approx \log n_2$

$$n_1 = n_1' l_1, \qquad n_2 = n_2' l_2, \qquad l_1 \approx \log n_1, \qquad l_2 \approx \log n_2$$

Exponential reduction of b, the bit-size of entries

$n_1 \times n_2$ transposition, 1-bit entries, $\log n_1 \approx \log n_2$

$n_1 = n_1' l_1, \quad n_2 = n_2' l_2, \quad l_1 \approx \log n_1, \quad l_2 \approx \log n_2$



Conclusion

Theorem

Assume $M(N)/N$ non-decreasing. Then $n_1 \times n_2$ transposition can be done in time

$$O(\mathsf{M}(n_1 n_2) + n_1 n_2 \log^* \max(n_1, n_2)).$$

Conclusion

Theorem

Assume $M(N)/N$ non-decreasing. Then $n_1 \times n_2$ transposition can be done in time

$$O(M(n_1 n_2) + n_1 n_2 \log^* \max(n_1, n_2)).$$

Theorem

In general, $n_1 \times n_2$ transposition can be done in time

$$O(M(n_1 n_2) + n_1 n_2 \log\log \max(n_1, n_2)).$$

Conclusion

Theorem

Assume $M(N)/N$ non-decreasing. Then $n_1 \times n_2$ transposition can be done in time

$$O(\mathsf{M}(n_1 n_2) + n_1 n_2 \log^* \max(n_1, n_2)).$$

Theorem

In general, $n_1 \times n_2$ transposition can be done in time

$$O(\mathsf{M}(n_1 n_2) + n_1 n_2 \log\log \max(n_1, n_2)).$$

Corollary

If $\mathsf{T}(N) = \Omega(N \log N)$, then $\mathsf{M}(N) = \Omega(N \log N)$.

Thank you !

http://www.TeXmacs.org