Fast multiplication of integers and real numbers

Marco Streng
Universiteit Leiden

De Leidsche Flesch – lunchlezing

Who am I?

Associate professor in Number Theory at Leiden University

Teaching:

- Linear algebra 1 for physics and astronomy
- Linear algebra 2 for mathematics (including Fall 2026)
- Algebra 1, 2, 3
- Algebraic Number Theory
- Elliptic Curves (including Spring 2027)
- Algorithms in Algebra (maybe Fall 2027)

What is number theory?

Number theory studies the positive integers $1, 2, 3, 4, 5, \ldots$ and everything related to it.

e.g.

Theorem ("Fermat's Last Theorem", Wiles (1994))

There are no positive integers a, b, c, n with $n \geq 3$ and

$$a^n + b^n = c^n.$$

Examples:

$$7^3 + 0^3 = 7^3, \qquad \text{but 0 is not positive,}$$

$$3^2 + 4^2 = 5^2, \qquad \text{but } 2 \not\geq 3.$$

This talk

- Introducing myself and my area
- What is “fast multiplication”?
- Why not to care
- Why to care
- Karatsuba’s algorithm
- The state of the art

Adding integers of n digits in $O(n)$ time

$$
\begin{array}{l}
31415926535897932384626433833 \\
\underline{27182818284590452353602874714\ +}
\end{array}
$$

Adding integers of n digits in $O(n)$ time

$$
\begin{array}{r}
31415926535897932384626433833 \\
27182818284590452353602874714 + \\
\hline
7
\end{array}
$$

Adding integers of n digits in $O(n)$ time

$$
\begin{array}{r}
31415926535897932384626433833 \\
27182818284590452353602874714 \; + \\
\hline
47
\end{array}
$$

Adding integers of n digits in $O(n)$ time

$$
\begin{array}{r}
1 \\
3141592653589793238462643383 \\
2718281828459045235360287471\mathbf{4} + \\
\hline
5\mathbf{4}7
\end{array}
$$

Adding integers of n digits in $O(n)$ time

$$
\begin{array}{r}
1 \\
3141592653589793238462643383 \\
2718281828459045235360287\color{blue}{4}714 \; + \\
\hline
\color{blue}{8}547
\end{array}
$$

Adding integers of n digits in $O(n)$ time

$$
\begin{array}{r}
31415926535897932384626433833 \\
27182818284590452353602874714 \; + \\
\hline
58598744820488384738229308547
\end{array}
$$

Adding integers of n digits in $O(n)$ time

$$
\begin{array}{r}
31415926535897932384626433833 \\
27182818284590452353602874714 \; + \\
\hline
58598744820488384738229308547
\end{array}
$$

- Time is linear in the number of digits, $O(n)$.
- Can't do better than linear: need to write down the output. So $O(n)$ is optimal.
- $2\times$ as many digits $\Longrightarrow$ $2\times$ as much work.

Multiplying integers of n digits, the primary school way

$$\begin{array}{r} 12767 \\ 31415 \times \\ \hline \end{array}$$

Multiplying integers of n digits, the primary school way

$$
\begin{array}{r}
12767 \\
31415 \times \\
\hline
5
\end{array}
$$

Multiplying integers of n digits, the primary school way

$$\begin{array}{r} 12767 \\ 31415 \times \\ \hline 35 \end{array}$$

Multiplying integers of n digits, the primary school way

$$\begin{array}{r} 12767 \\ 31415 \times \\ \hline 63835 \end{array}$$

Multiplying integers of n digits, the primary school way

$$
\begin{array}{r}
12767 \\
31415 \times \\
\hline
63835 \\
127670
\end{array}
$$

Multiplying integers of n digits, the primary school way

$$
\begin{array}{r}
12767 \\
31415 \times \\
\hline
63835 \\
127670 \\
5106800
\end{array}
$$

Multiplying integers of n digits, the primary school way

$$
\begin{array}{r}
12767 \\
31415 \times \\
\hline
63835 \\
127670 \\
5106800 \\
12767000
\end{array}
$$

Multiplying integers of n digits, the primary school way

$$
\begin{array}{r}
12767 \\
31415 \times \\
\hline
63835 \\
127670 \\
5106800 \\
12767000 \\
383010000
\end{array}
$$

Multiplying integers of n digits, the primary school way

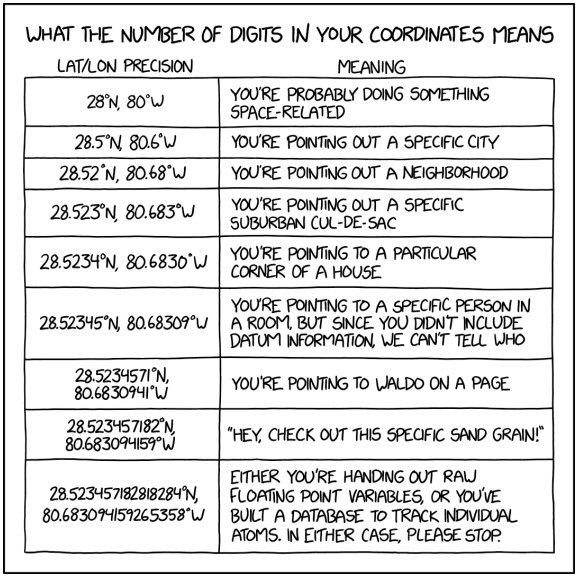
$$
\begin{array}{r}
12767 \\
31415 \times \\
\hline
63835 \\
127670 \\
5106800 \\
12767000 \\
383010000 + \\
\hline
401075305
\end{array}
$$

Multiplying integers of n digits, the primary school way

$$
\begin{array}{r}
12767 \\
31415 \times \\
\hline
63835 \\
127670 \\
5106800 \\
12767000 \\
383010000 + \\
\hline
401075305
\end{array}
$$

- Time is quadratic in the number of digits, $O(n^2)$.
- With 10000 digits, this would be slow!
- $2\times$ as many digits $\Longrightarrow$ $4\times$ as much work.
- Is this optimal or can we do better than quadratic?

Why not to care about 10000 digits

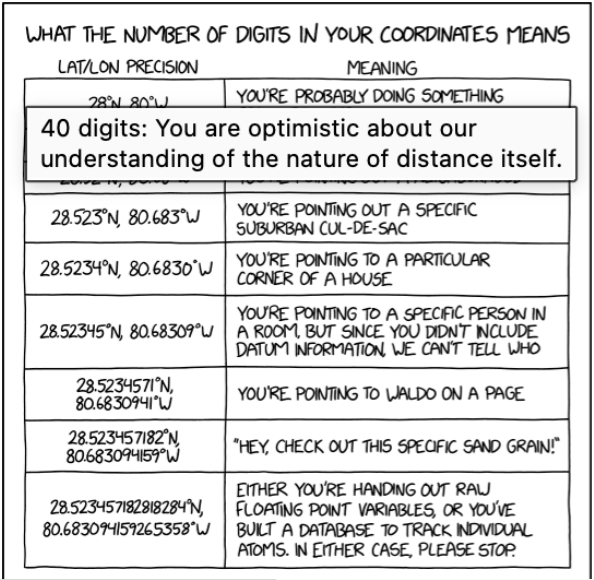


WHAT THE NUMBER OF DIGITS IN YOUR COORDINATES MEANS

LAT/LON PRECISION	MEANING
28°N, 80°W	YOU'RE PROBABLY DOING SOMETHING SPACE-RELATED
28.5°N, 80.6°W	YOU'RE POINTING OUT A SPECIFIC CITY
28.52°N, 80.68°W	YOU'RE POINTING OUT A NEIGHBORHOOD
28.523°N, 80.683°W	YOU'RE POINTING OUT A SPECIFIC SUBURBAN CUL-DE-SAC
28.5234°N, 80.6830°W	YOU'RE POINTING TO A PARTICULAR CORNER OF A HOUSE
28.52345°N, 80.68309°W	YOU'RE POINTING TO A SPECIFIC PERSON IN A ROOM. BUT SINCE YOU DIDN'T INCLUDE DATUM INFORMATION, WE CAN'T TELL WHO
28.5234571°N, 80.6830941°W	YOU'RE POINTING TO WALDO ON A PAGE
28.52345718°N, 80.68309415°W	"HEY, CHECK OUT THIS SPECIFIC SAND GRAIN!"
28.523457182818284°N, 80.683094159265358°W	EITHER YOU'RE HANDING OUT RAW FLOATING POINT VARIABLES, OR YOU'VE BUILT A DATABASE TO TRACK INDIVIDUAL ATOMS. IN EITHER CASE, PLEASE STOP.

image source:
`https://xkcd.com/2170/`
Randall Munroe, 2019
CC BY-NC 2.5

Why not to care about 10000 digits



WHAT THE NUMBER OF DIGITS IN YOUR COORDINATES MEANS

LAT/LON PRECISION	MEANING
28°N, 80°W	YOU'RE PROBABLY DOING SOMETHING
40 digits: You are optimistic about our understanding of the nature of distance itself.	
28.523°N, 80.683°W	YOU'RE POINTING OUT A SPECIFIC SUBURBAN CUL-DE-SAC
28.5234°N, 80.6830°W	YOU'RE POINTING TO A PARTICULAR CORNER OF A HOUSE
28.52345°N, 80.68309°W	YOU'RE POINTING TO A SPECIFIC PERSON IN A ROOM, BUT SINCE YOU DIDN'T INCLUDE DATUM INFORMATION, WE CAN'T TELL WHO
28.5234571°N, 80.6830941°W	YOU'RE POINTING TO WALDO ON A PAGE
28.52345718°N, 80.68309411 59°W	"HEY, CHECK OUT THIS SPECIFIC SAND GRAIN!"
28.523457182818284°N, 80.683094159265358°W	EITHER YOU'RE HANDING OUT RAW FLOATING POINT VARIABLES, OR YOU'VE BUILT A DATABASE TO TRACK INDIVIDUAL ATOMS. IN EITHER CASE, PLEASE STOP.

image source:
`https://xkcd.com/2170/`
Randall Munroe, 2019
CC BY-NC 2.5

Why to care about 10000 digits

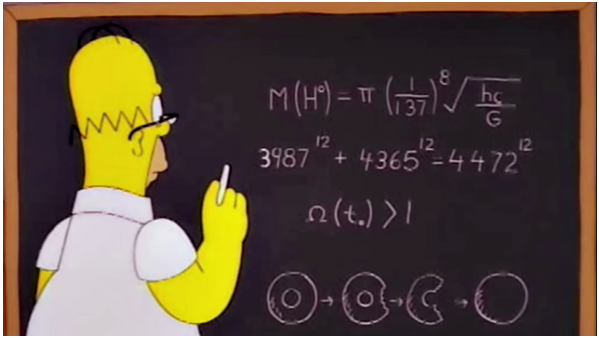


image source:
The Simpsons S10E02, 1998
The Wizard of Evergreen Terrace

$$3987^{12} + 4365^{12} = 4472^{12} \quad ?$$

$$\boxed{6.397665635E43} = \boxed{6.397665635E43}$$

Contradicts Fermat's Last Theorem!

Karatsuba's multiplication algorithm

Given A and B of n digits,

Example:

$$\overbrace{3141592653589793238462643 38}^{A} \cdot \overbrace{271828182845904523536028747}^{B}$$

Karatsuba's multiplication algorithm

Given A and B of n digits, let $m = \lceil n/2 \rceil$ and write

$$A = A_1 \cdot 10^m + A_0 \quad , \quad B = B_1 \cdot 10^m + B_0.$$

Example:

$$\overbrace{\underbrace{3141592653589}_{A_1}\underbrace{79323846264338}_{A_0}}^{A} \cdot \overbrace{\underbrace{2718281828459}_{B_1}\underbrace{04523536028747}_{B_0}}^{B}$$

Karatsuba's multiplication algorithm

Given A and B of n digits, let $m = \lceil n/2 \rceil$ and write

$$A = A_1 \cdot 10^m + A_0 \quad , \quad B = B_1 \cdot 10^m + B_0.$$

Example:

$$\overbrace{\underbrace{3141592653589}_{A_1}\underbrace{79323846264338}_{A_0}}^{A} \cdot \overbrace{\underbrace{2718281828459}_{B_1}\underbrace{04523536028747}_{B_0}}^{B}$$

Then

$$\begin{aligned} AB &= (A_1 \cdot 10^m + A_0)(B_1 \cdot 10^m + B_0) \\ &= A_1B_1 \cdot 10^{2m} + (A_1B_0 + A_0B_1) \cdot 10^m + A_0B_0 \end{aligned}$$

Karatsuba's multiplication algorithm

Given A and B of n digits, let $m = \lceil n/2 \rceil$ and write

$$A = A_1 \cdot 10^m + A_0 \quad , \quad B = B_1 \cdot 10^m + B_0.$$

Example:

$$\overbrace{\underbrace{3141592653589}_{A_1}\underbrace{79323846264338}_{A_0}}^{A} \cdot \overbrace{\underbrace{2718281828459}_{B_1}\underbrace{04523536028747}_{B_0}}^{B}$$

Then

$$\begin{aligned} AB &= (A_1 \cdot 10^m + A_0)(B_1 \cdot 10^m + B_0) \\ &= A_1B_1 \cdot 10^{2m} + (A_1B_0 + A_0B_1) \cdot 10^m + A_0B_0 \\ &= \boxed{A_1B_1} \cdot 10^{2m} + \Big(\boxed{(A_1 + A_0)(B_1 + B_0)} - \boxed{A_1B_1} - \boxed{A_0B_0} \Big) \cdot 10^m + \boxed{A_0B_0} \end{aligned}$$

Karatsuba's multiplication algorithm

$$\overbrace{\underbrace{3141592653589}_{A_1}\underbrace{79323846264338}_{A_0}}^{A} \cdot \overbrace{\underbrace{2718281828459}_{B_1}\underbrace{04523536028747}_{B_0}}^{B}$$

$$\begin{aligned} AB &= (A_1 \cdot 10^m + A_0)(B_1 \cdot 10^m + B_0) \\ &= A_1B_1 \cdot 10^{2m} + (A_1B_0 + A_0B_1) \cdot 10^m + A_0B_0 \\ &= \boxed{A_1B_1} \cdot 10^{2m} + \Big(\boxed{(A_1 + A_0)(B_1 + B_0)} - \boxed{A_1B_1} - \boxed{A_0B_0}\Big) \cdot 10^m + \boxed{A_0B_0} \end{aligned}$$

- Compute the three smaller products recursively.

Karatsuba's multiplication algorithm

$$\overbrace{\underbrace{3141592653589}_{A_1}\underbrace{79323846264338}_{A_0}}^{A} \cdot \overbrace{\underbrace{2718281828459}_{B_1}\underbrace{04523536028747}_{B_0}}^{B}$$

$$\begin{aligned} AB &= (A_1 \cdot 10^m + A_0)(B_1 \cdot 10^m + B_0) \\ &= A_1B_1 \cdot 10^{2m} + (A_1B_0 + A_0B_1) \cdot 10^m + A_0B_0 \\ &= \boxed{A_1B_1} \cdot 10^{2m} + \Big(\boxed{(A_1 + A_0)(B_1 + B_0)} - \boxed{A_1B_1} - \boxed{A_0B_0}\Big) \cdot 10^m + \boxed{A_0B_0} \end{aligned}$$

- Compute the three smaller products recursively.
- $2\times$ as many digits $\Rightarrow \approx 3\times$ as much work.
- $3 < 4$, so better than $O(n^2)$. In fact, $O(n^{1.59})$.

The state of the art (selection of algorithms)

primary school method	$O(n^2)$	
Karatsuba (1960)	$O(n^{1.59})$	
Schönhage - Strassen (1971)	$O(n \log(n) \log\log(n))$	Fast in practice, part of the course Algorithms in Algebra
Harvey - Van der Hoeven (2019)	$O(n \log(n))$	Expected to be asymptotically optimal, not used in practice