Design & Analysis of Algorithms

Divide and Conquer

Prof. Wilson Rivera

Spring 2018

It’s actually pretty amazing that you can big multiply numbers quickly at all

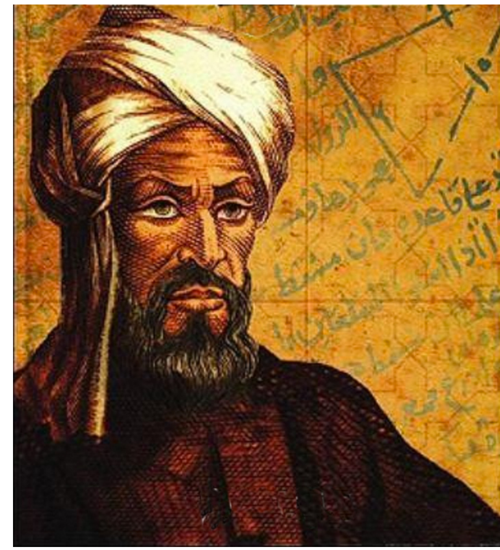
- You could do this when you were 8.
- It wasn’t always so easy!

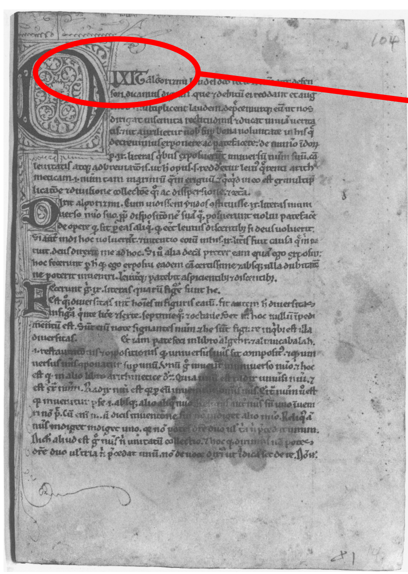
LXXXIX × CM = ?

Etymology of “Algorithm”

- Al-Khwarizmi (Persian mathematician, lived around 800AD) wrote a book about how to multiply with Arabic numerals.

- His ideas came to Europe in the 12th century.





Dixit algorizmi
(so says Al-Khwarizmi)

- Originally, “Algorisme” [old French] referred to just the Arabic number system, but eventually it came to mean “Algorithm” as we know today.

Outline

- Integer multiplication
- Divide and conquer
- Karatsuba integer multiplication

A problem you all know how to solve:

Integer Multiplication

$$
\begin{array}{r}
\overbrace{1233925720752752384623764283568364918374523856298}^{n} \\
\times \quad 4562323582342395285623467235019130750135350013753 \\
\hline
\end{array}
$$

???

How would you solve this problem?

How long would it take you?

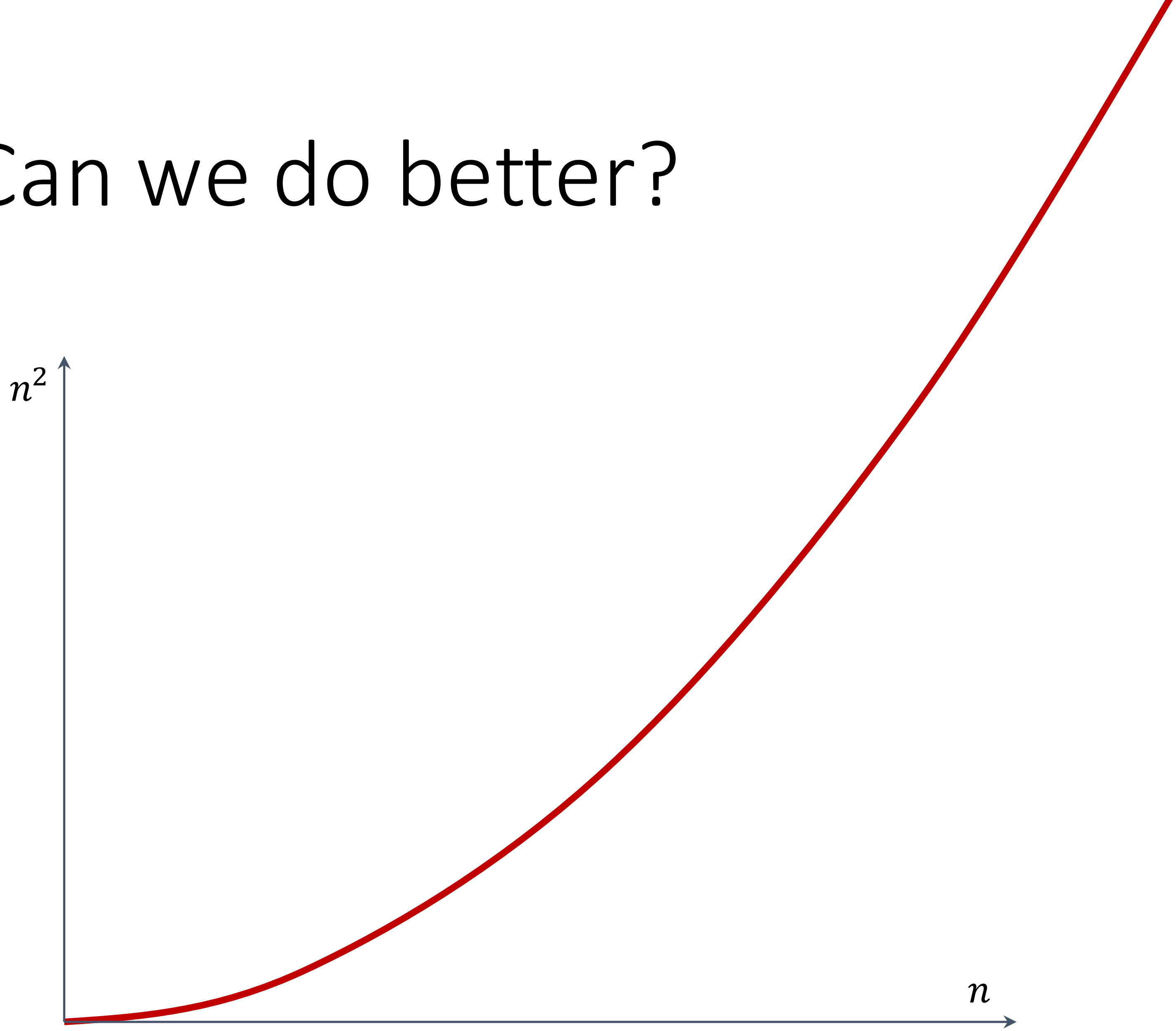
About n^2 one-digit operations

At most n^2 multiplications,

and then at most n^2 additions (for carries)

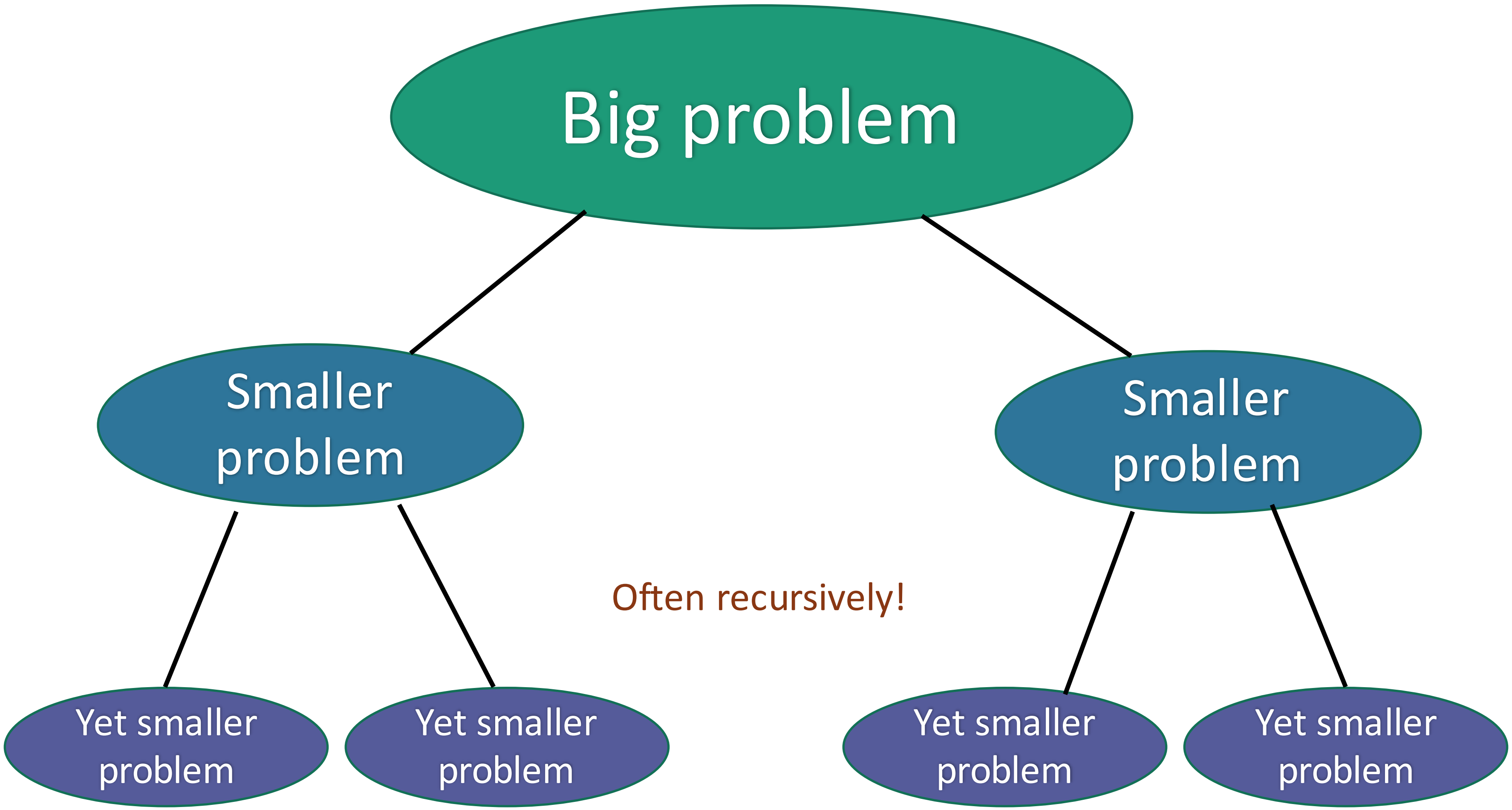
and then I have to add n different 2n-digit numbers…

Can we do better?



Divide and conquer

Break problem up into smaller (easier) sub-problems



Divide and conquer for multiplication

Break up an integer:

$$1234 = 12 \times 100 + 34$$

$$
\begin{aligned}
&1234 \times 5678 \\
&= (12 \times 100 + 34)(56 \times 100 + 78) \\
&= \underbrace{(12 \times 56)}_{1}10000 + (\underbrace{34 \times 56}_{2} + \underbrace{12 \times 78}_{3})100 + \underbrace{(34 \times 78)}_{4}
\end{aligned}
$$

One 4-digit multiply $\Rightarrow$ Four 2-digit multiplies

More generally

Break up an n-digit integer:

$$[x_1 x_2 \cdots x_n] = [x_1 x_2 \cdots x_{n/2}] \times 10^{n/2} + [x_{n/2+1} x_{n/2+2} \cdots x_n]$$

$$x \times y = (a \times 10^{n/2} + b)(c \times 10^{n/2} + d)$$

$$= \underbrace{(a \times c)}_{1} 10^n + (\underbrace{a \times d}_{2} + \underbrace{c \times b}_{3}) 10^{n/2} + \underbrace{(b \times d)}_{4}$$

One n-digit multiply → Four (n/2)-digit multiplies

Divide and conquer algorithm

x,y are n-digit numbers

Multiply(x, y):

- Write $x = a\,10^{\frac{n}{2}} + b$
- Write $y = c\,10^{\frac{n}{2}} + d$
- Recursively compute ac, ad, bc, bd:
  - ac = **Multiply**(a, c), etc…
- Add them up (with shifts) to get xy

a, b, c, d are
n/2-digit numbers

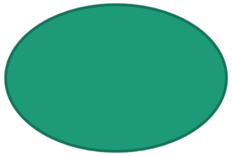
Questions about the algorithm?

How long does this take?

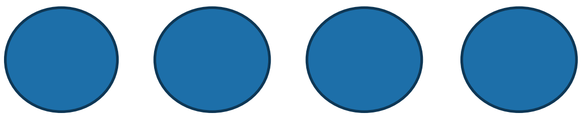
Better or worse than the grade-school algorithm?

Another way to see this*

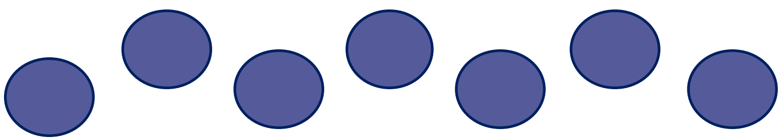
*we will come back to this sort of analysis later and still more rigorously.



1 problem
of size n

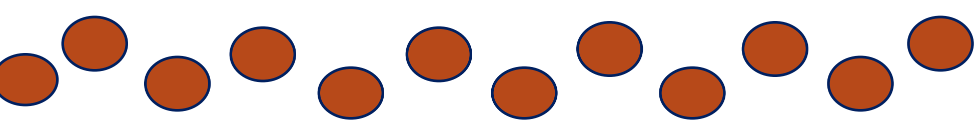


4 problems
of size n/2



4^t problems
of size $n/2^t$

...



$\underline{n^2}$ problems
of size 1

- If you cut n in half $\log_2(n)$ times, you get down to 1.
- So we do this $\log_2(n)$ times and get…

$4^{\log\_2(n)} = n^2$
problems of size 1.

Yet another way to see this

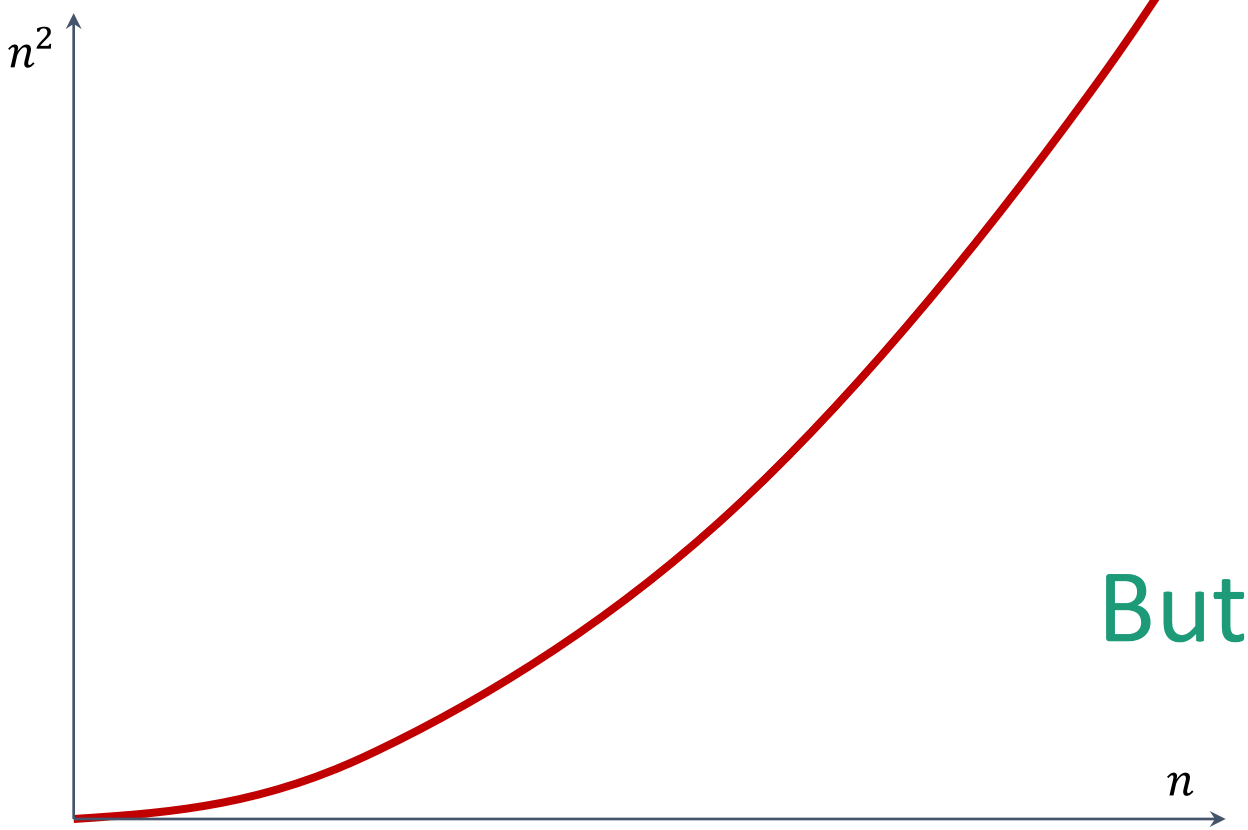
- Let T(n) be the time to multiply two n-digit numbers.
- Recurrence relation:
  - $T(n) = 4 \cdot T\left(\frac{n}{2}\right) +$ (about n to add stuff up)

Ignore this term for now…

$$
\begin{aligned}
T(n) &= 4 \cdot T(n/2) \\
&= 4 \cdot (4 \cdot T(n/4)) && 4^2 \cdot T(n/2^2) \\
&= 4 \cdot (4 \cdot (4 \cdot T(n/8))) && 4^3 \cdot T(n/2^3) \\
&\vdots \\
&= 2^{2t} \cdot T(n/2^t) && 4^t \cdot T(n/2^t) \\
&\vdots \\
&= n^2 \cdot T(1). && 4^{\log_2(n)} \cdot T(n/2^{\log_2(n)})
\end{aligned}
$$

That's a bit dissappointing

All that work and still n^2...



But wait!!

Divide and conquer can actually make progress

- Karatsuba figured out how to do this better!

$$\begin{aligned} xy &= (a \cdot 10^{n/2} + b)(c \cdot 10^{n/2} + d) \\ &= ac \cdot 10^{n} + (ad + bc)10^{n/2} + bd \end{aligned}$$

Need these three things

- If only we recurse three times instead of four…

Karatsuba integer multiplication

- Recursively compute
  - ac
  - bd
  - (a+b)(c+d)

Subtract these off

get this

(a+b)(c+d) = ac + bd + bc + ad

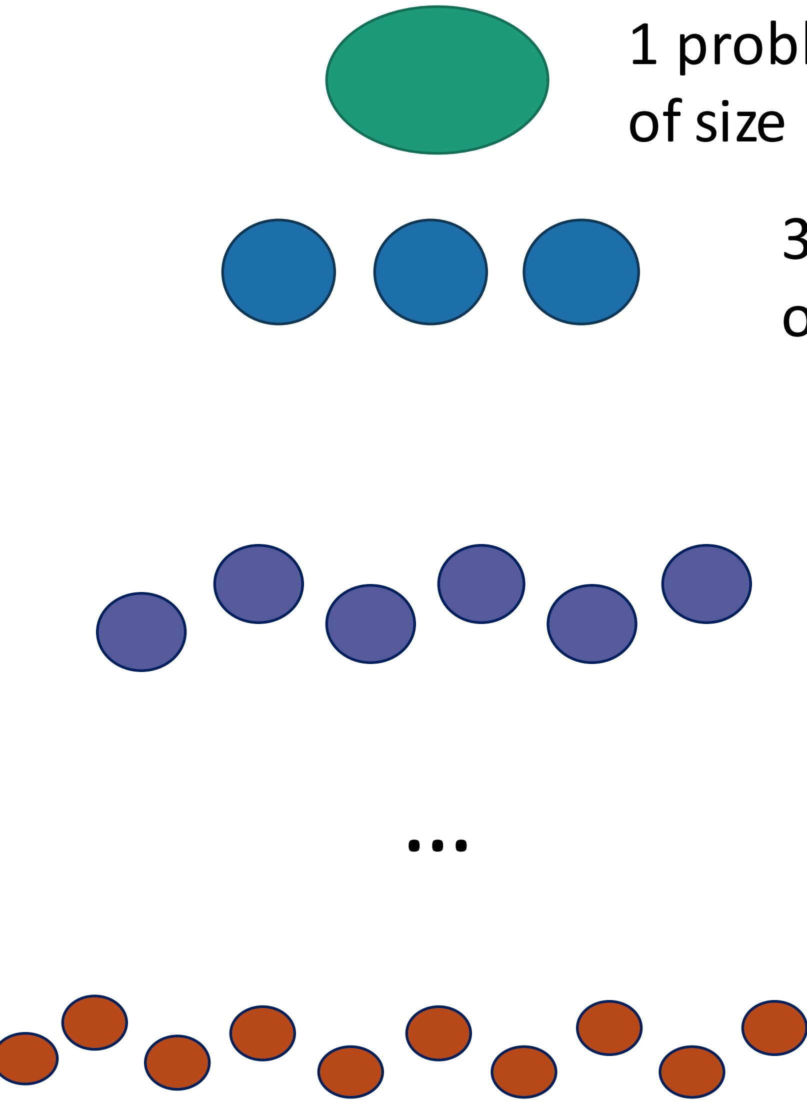
- Assemble the product:

$$xy = (a \cdot 10^{n/2} + b)(c \cdot 10^{n/2} + d)$$

$$= ac \cdot 10^{n} + (ad + bc)10^{n/2} + bd$$

✓ ✓ ✓

What's the running time?



1 problem
of size n

3 problems
of size n/2

3^t problems
of size $n/2^t$

…

$n^{1.6}$ problems
of size 1

- If you cut n in half $\log_2(n)$ times, you get down to 1.
- So we do this $\log_2(n)$ times and get…

$3^{\log\_2(n)} = n^{\log\_2(3)} = n^{1.6}$
problems of size 1.

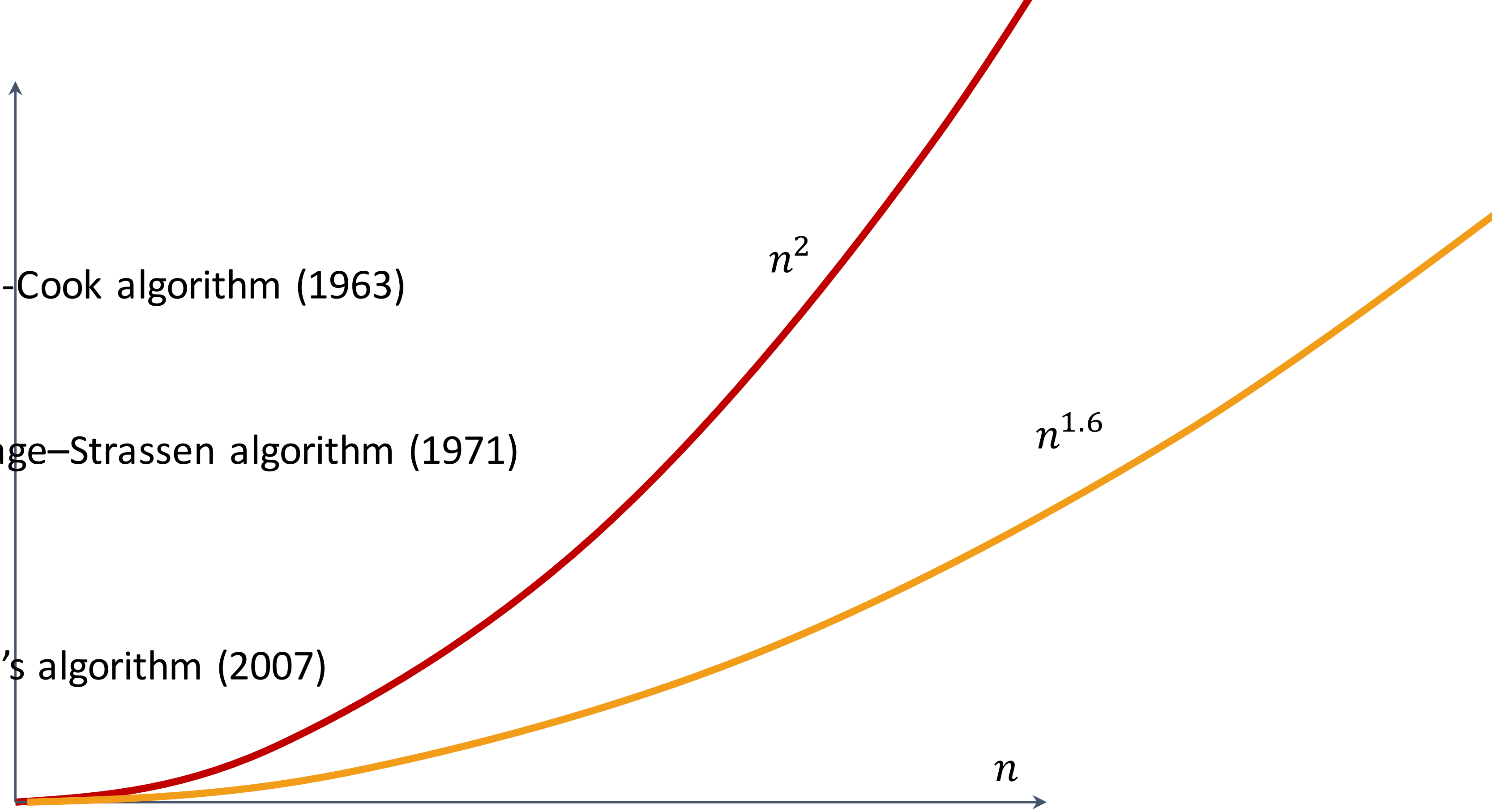
This is much better!

(Can we do better still?)

Toom-Cook algorithm (1963)

Schönhage–Strassen algorithm (1971)

Furer's algorithm (2007)



Conclusion

- Integer multiplication
- Divide and conquer
- Karatsuba integer multiplication