Karatsuba Multiplication

Fun with Algorithms

July 10th, 2021 (Class #1)

The multiplication problem

Input: Two n-digit integers x and y.

Output: Their product xy.

$$\begin{array}{r} 1357 \\ \times\ 2468 \\ \hline \boxed{} \end{array}$$

What is an algorithm?

A series of <u>simple steps</u> to solve a problem.

Running time $T(n)$: The number of simple steps!

Simple steps for multiplication?

Adding single-digit numbers.

Multiplying single-digit numbers.

Algorithm for multiplication?

$$\begin{array}{r} 1357 \\ \times\ 2468 \\ \hline \end{array}$$

Grade-school Multiplication!

$$
\begin{array}{cccccccc}
 & & & & & 1 & 1 & \\
 & & & & 1 & 2 & 2 & \\
 & & & & 2 & 3 & 4 & \\
 & & & & 2 & 4 & 5 & \\
 & & & & 1 & 3 & 5 & 7 \\
 & & & \times & 2 & 4 & 6 & 8 \\
\hline
 & & & 1 & 0^{2} & 8 & 5 & 6 \\
 & & & 8 & 1 & 4 & 2 & 0 \\
 & & 5^{1} & 4 & 2 & 8 & 0 & 0 \\
 & 2^{1} & 7 & 1 & 4 & 0 & 0 & 0 \\
\hline
 & 3 & 3 & 4 & 9 & 0 & 7 & 6
\end{array}
$$

How fast is grade-school multiplication?

Multiplication versus **Addition**

```
    1357
 x  2468
 -------
   10856
   81420
  542800
 2714000
 -------
 3349076
```

```
  1357
+ 2468
------
  3825
```

Grade-school addition is $O(n)$. No room for improvement.

Grade-school multiplication is $O(n^2)$. Room for improvement!

The goal of algorithm design

We always want to come up with faster algorithms.

But what does "faster" mean?

Large Input Part 1 (1)

Step 1.
2.
3.
4.
5.

$T(n) = n^2$

6.
7.
8.

$T(n) = n$

$T(n) =$ ______

Large Input Part 1 (2)

$T(n) = n^2 + n$

try $n = 100$

$T(100) = 100^2 + 100$

$= 10{,}100$

100 is roughly ____% of 10,100!

Large Input Part 1 (3)

$T(n) = n^2 + n$

100,000 is roughly ______%

try $n = 100{,}000$

of 10,000,100,000 !

$T(100{,}000) = 100{,}000^2 + 100{,}000$

$= 10{,}000{,}100{,}000$

$\Longrightarrow$

Rule 1 for Big O

$T(n) = f(n) + g(n)$, pick SLOWEST

Step 1.
2.
3.
4.
5.

$T(n) = n^2$

6.
7.
8.

$T(n) = n$

$T(n) = n^2 + n =$ ______

Large Input Part 2

$T(n) = n^2$ vs $T(n) = 100n$

Try $n = 10^{10}$

$T(10^{10}) = 10^{20}$

$T(10^{10}) = 10^{12}$ → factor of ____ apart

try $n = 10^{20}$

$T(10^{20}) = 10^{40}$

$T(10^{20}) = 10^{22}$ → factor of ____ apart

behaves like ____

behaves like ____

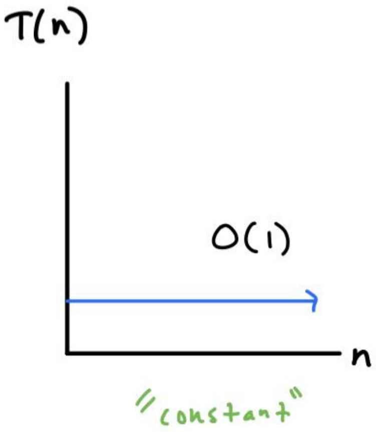
Rule 2 for Big O

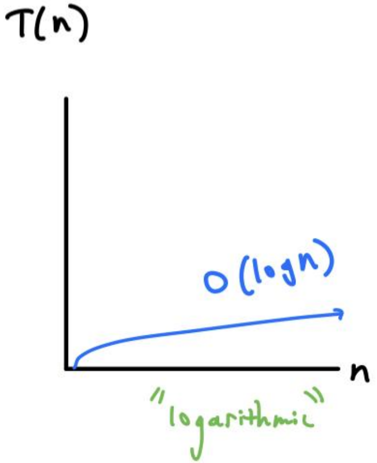
$T(n) = cf(n)$, ignore constant

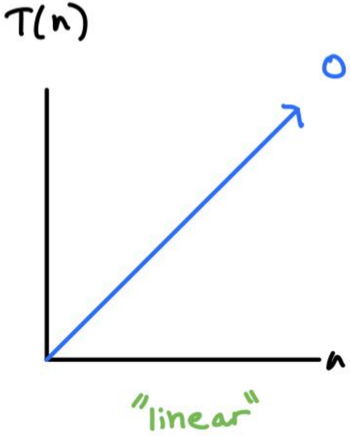
$T(n) = 6 \log n =$ ________

Time Complexity Graphs

FAST → SLOW

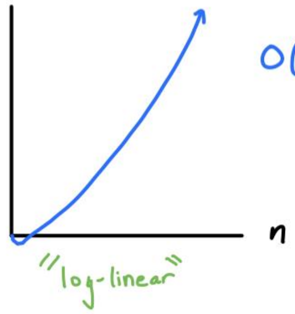


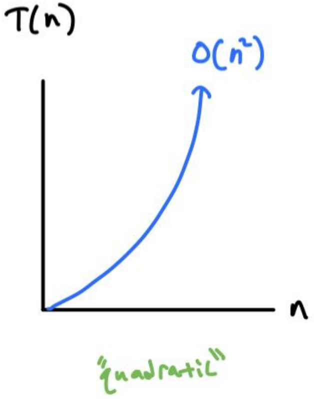


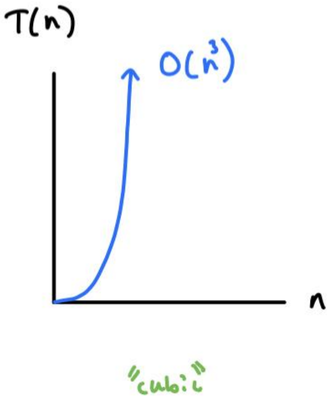


Time Complexity Graphs (continued)

FAST → SLOW

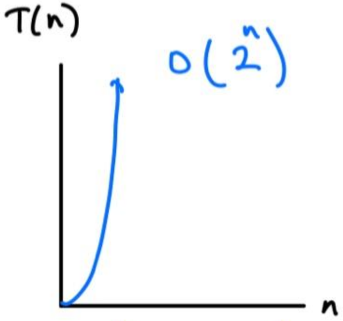




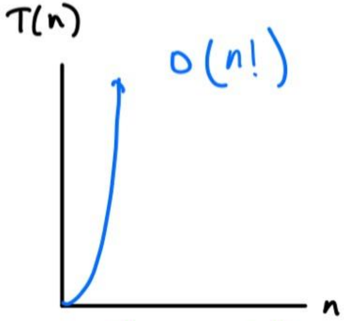


Time Complexity Graphs (continued)

FAST → SLOW

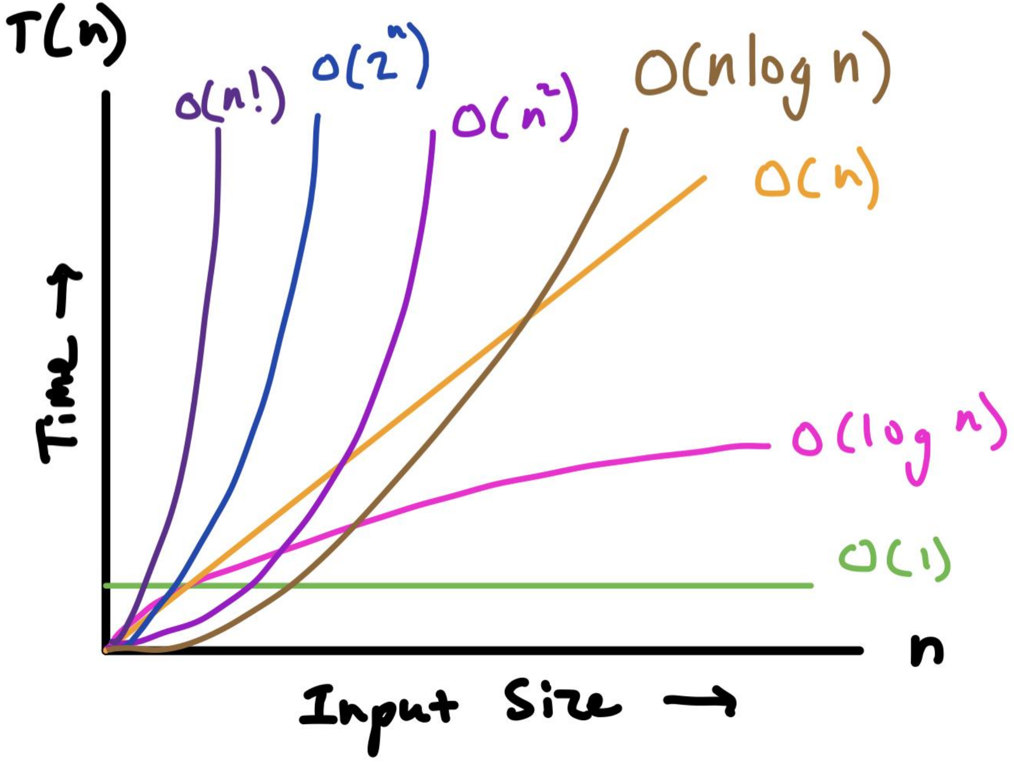


"exponential"



"factorial"

Big O



Big O Examples

1. $T(n) = 5n^2 + 10n$

2. $T(n) = 12n + 3\log n$

3. $T(n) = 17n^3 + 5 \cdot 2^n$

4. $T(n) = 1 + 2 + 3 + \cdots + n$

[5 Minute Break](#)

After the break...

Can we do better than $O(n^2)$?

Finding a **better** algorithm… divide and conquer paradigm

$\boxed{\text{Divide}}$

"Magically" solve smaller instances of the problem ("subproblems").

$\boxed{\text{Conquer}}$

Use the solutions to the subproblems to solve the original problem!

How do we actually solve subproblems?

Just divide again! (Recursion!)
Until n=1.

Divide and conquer applied to multiplication

$n = 8$:

$$x = \overset{A}{\boxed{5143}}\,\overset{B}{\boxed{2361}}$$

$$y = \underset{C}{\boxed{2981}}\,\underset{D}{\boxed{4772}}$$

How do we conquer?

$$x = \overbrace{\boxed{5143}}^{A}\overbrace{\boxed{2361}}^{B} = 10^4 A + B$$

$$y = \underbrace{\boxed{2981}}_{C}\underbrace{\boxed{4772}}_{D} = 10^4 C + D$$

Doing the algebra

$$\begin{aligned} xy &= (10^4 A + B)(10^4 C + D) \\ &= 10^8 AC + 10^4 AD + 10^4 BC + BD. \end{aligned}$$

The divide-and-conquer algorithm for multiplication!

- Do 4 multiplications of size $\frac{n}{2}$ (how? by recursion!) to find AC, AD, BC, and BD.
- By shifting and adding, find

$$xy = 10^{n} AC + 10^{\frac{n}{2}} AD + 10^{\frac{n}{2}} BC + BD.$$

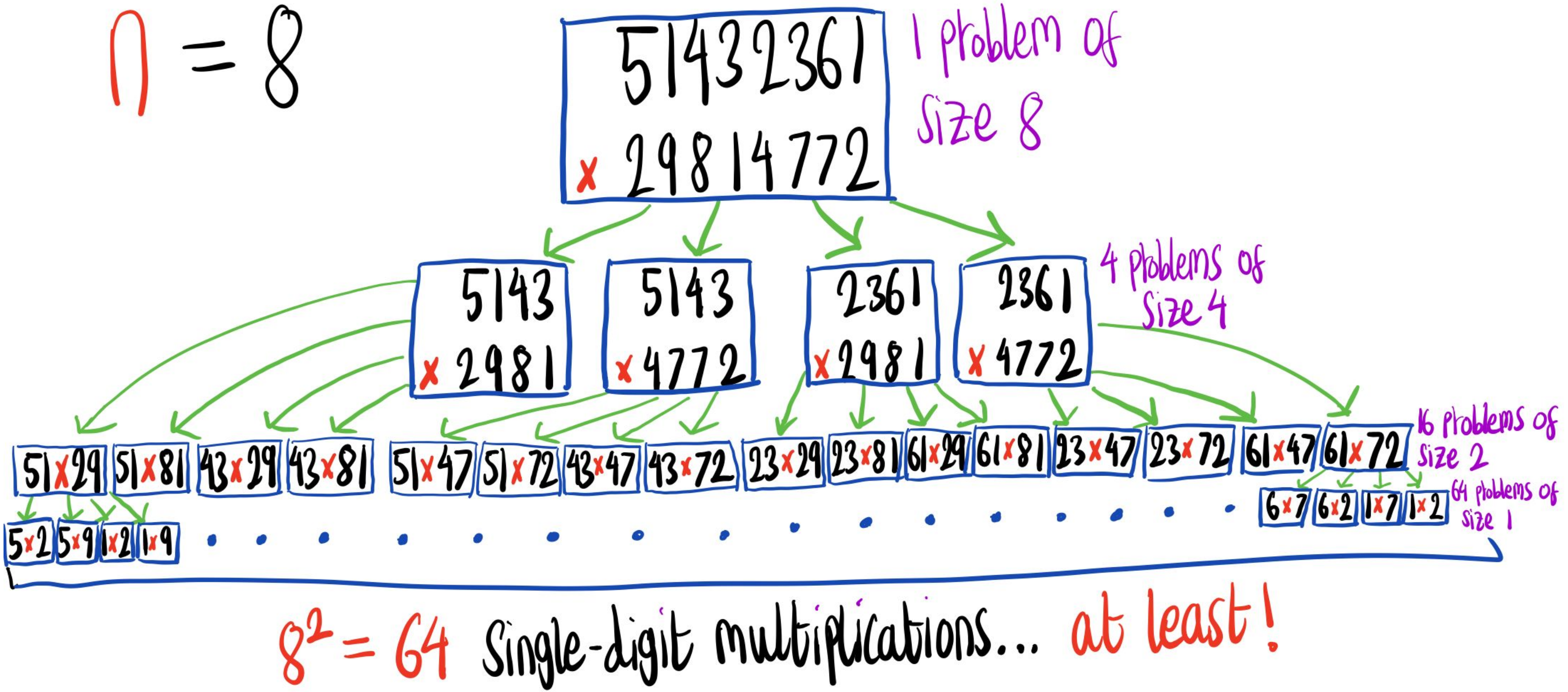
- We're done!

How fast is the divide-and-conquer algorithm?

$$\boxed{T(n) = 4\,T\left(\frac{n}{2}\right) + O(n).}$$

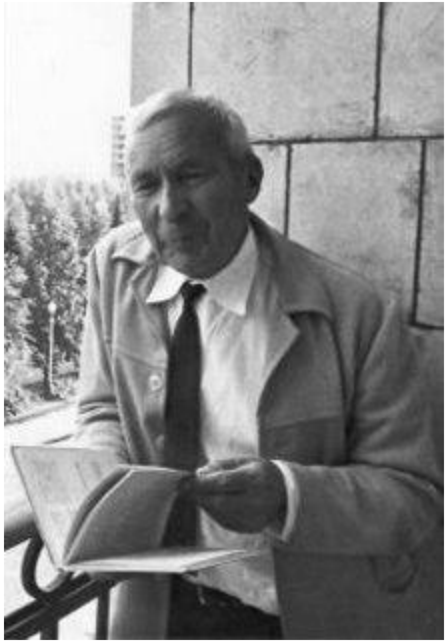
- Split into 4 problems of size $\frac{n}{2}$.
- After we've solved them, only $O(n)$ work needed for final answer.

How fast is the divide-and-conquer algorithm?

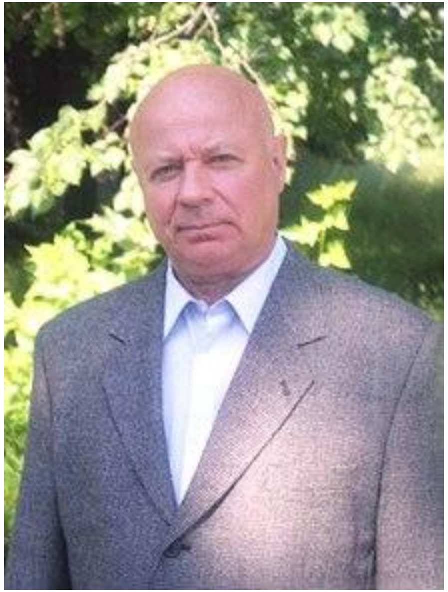


Was all this for nothing?

No!



Andrey Kolmogorov



Anatoly Karatsuba

Karatsuba's trick!

$$xy = 10^{n} AC + 10^{\frac{n}{2}} AD + 10^{\frac{n}{2}} BC + BD$$
$$= 10^{n} AC + 10^{\frac{n}{2}} (AD + BC) + BD.$$

We really want AC, BD, and $\underline{AD + BC}$.

Karatsuba's Algorithm

We really want AC, BD, and $\underline{AD+BC}$.

- Find AC.
- Find BD.
- Find... $(A+B)(C+D)$

$$= AC + BD + AD + BC.$$

Only 3 recursive calls!

Conclusion

$$T(n) = \overset{3}{\cancel{4}}\, T\left(\frac{n}{2}\right) + O(n).$$

Previously we had $O(n^2)$.

Karatsuba's algorithm is faster.

(how much faster? analysis next class!)