

HWS: Tue Oct 6 (today)
 GPS 6: Mon Oct 12

$$\begin{array}{r}
 23958233 \\
 \times \quad 5830 \\
 \hline
 00000000 \quad (= 23,958,233 \times 0) \\
 71874699 \quad (= 23,958,233 \times 30) \\
 191665864 \quad (= 23,958,233 \times 800) \\
 + 119791165 \quad (= 23,958,233 \times 5,000) \\
 \hline
 139676498390 \quad (= 139,676,498,390)
 \end{array}$$

$n \times n$ multiplication

$O(n^2)$ time (lattice multiplication)

Can we do better?

Divide by $\#$ digits.

Let $m = \lceil n/2 \rceil$

Express the input numbers (n -digits each) as pairs of m -digit numbers.

Given x & y . Want to know $x \cdot y$.

Set $a, b, c, & d$ so

$$x = a \cdot 10^m + b$$

$$y = c \cdot 10^m + d$$

$$x \cdot y = (a \cdot 10^m + b)(c \cdot 10^m + d)$$

$$= ac \cdot 10^{2m} + ad \cdot 10^m + bc \cdot 10^m + bd$$



m -digit mult so we can recurse!

SplitMultiply(x, y, n): $\leftarrow$ computes

if $n = 1$

return $x \cdot y$

else

$m \leftarrow \lceil n/2 \rceil$

$a \leftarrow \lfloor x/10^m \rfloor; b \leftarrow x \bmod 10^m$

$c \leftarrow \lfloor y/10^m \rfloor; d \leftarrow y \bmod 10^m$

$e \leftarrow \text{SplitMultiply}(a, c, m)$

$f \leftarrow \text{SplitMultiply}(b, d, m)$

$g \leftarrow \text{SplitMultiply}(b, c, m)$

$h \leftarrow \text{SplitMultiply}(a, d, m)$

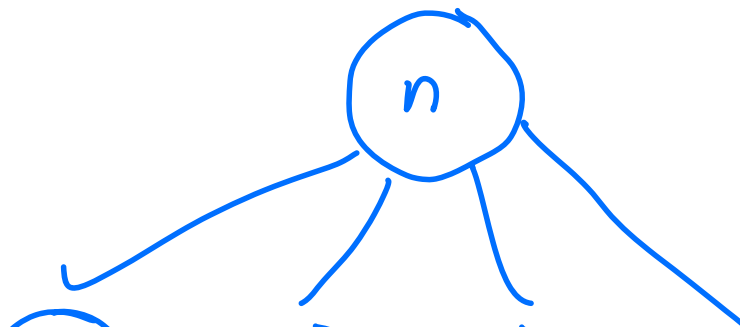
return $10^{2m}e + 10^m(g + h) + f$

x, y assuming both
use $\leq n$ digits
($\&$ are non-neg.
integers)

single digit add, mults

$T(n)$: ~~time~~ for SplitMultiply(x, y, n):

$$T(n) = 4T(n/2) + O(n)$$



n



increasing geometric series

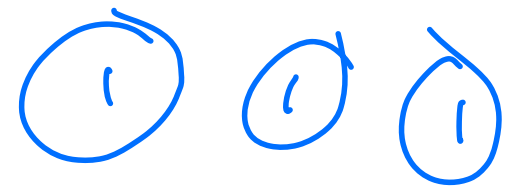
so $T(n) = O(\text{largest level sum})$

$= O(\text{sum of leaf values})$

$= O(\# \text{ leaves})$

$= O(4^{\text{depth}})$

$= O(4^{\log_2 n}) = O(n^{\log_2 4}) = \underline{O(n^2)}$



$$\begin{aligned}
 ad + bc &= (ac + bd) + (-ac - bd + ad + bc) \\
 &= ac + bd - (ac + bd - ad - bc) \\
 &= ac + bd - (a - b)(c - d)
 \end{aligned}$$

FastMultiply(x, y, n):

if $n = 1$

return $x \cdot y$

else

$m \leftarrow \lceil n/2 \rceil$

$a \leftarrow \lfloor x/10^m \rfloor; b \leftarrow x \bmod 10^m$

$c \leftarrow \lfloor y/10^m \rfloor; d \leftarrow y \bmod 10^m$

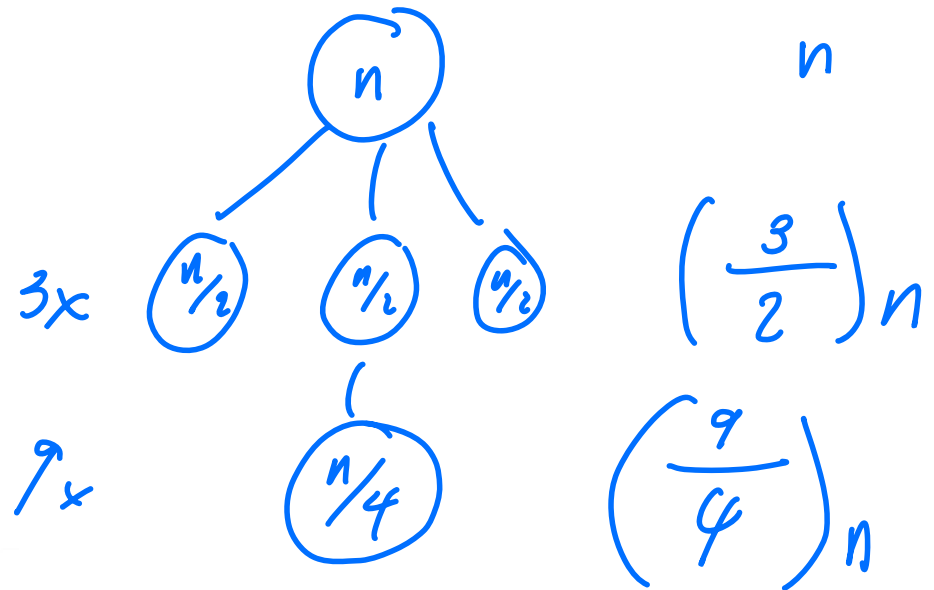
$e \leftarrow \text{FastMultiply}(a, c, m)$

$f \leftarrow \text{FastMultiply}(b, d, m)$

$g \leftarrow \text{FastMultiply}(a - b, c - d, m)$

return $10^{2m}e + 10^m(e + f - g) + f$

$$T(n) = 3T(n/2) + O(n)$$



$$T(n) = O(\# \text{leaves}) = O(3^{\log_2 n}) = O(n^{\log_2 3}) = O(n^{1.59})$$

1971: $O(n \log n \log \log n)$

2019: $O(n \log n)$

Backtracking: First make it correct.
Then make it fast.

Solve a problem by making several
decisions.

At a decision point, try every option/
choice &
recursively deal with the consequences.

Text segmentation: Given a string

$A[1..n]$.

PRIMVSDIGNITASINTAMTENVISCIANTIANONPOTEST
ESSERESENIMSVNTPARVAEPROPEINSINGVLISLITTERIS
ATQVEINTERPVNCTIONIBUSVERBORVMOCVPATAE

Can we partition A into substrings
such that each is a word?

Given a function $\text{IsWord}(X[1..k])$ +
returns True/False for if X is a
word.

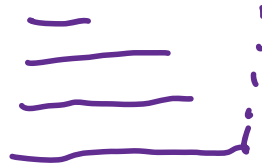
Backtracking: make a sequence of decisions...

Try to decide each next word one by one...

BLUESTEMUNITROBOTHEARTHANDSATURNSPIN

Partway through.

BLUE STEM UNIT ROBOT HEARTHANDSATURNSPIN



Could go with

HE, HEAR, HEART,

HEARTH next

Which one?

Try them all.

For each, add it to our list & ask

Recursion Fairy if we can segment

based on previous results.

based on previous decisions &
this next one.

For this problem, Recursion Fairy doesn't
care which words were chosen already,
Just what is left.

So define a problem using only what
characters remain.

Base case: Answer for empty string is
True.

Splittable($A[1 \dots n]$):

if $n = 0$

return True

~~for~~ $i \leftarrow 1$ to n

if IsWord($A[1 \dots i]$)

if Splittable($A[i + 1 \dots n]$)

return True

return False

← try every first/next word & recursively see if we can segment the rest

Correct but not fast

Preparing for fast.

Make input $A[1..n]$ a "global variable"

Use first index or suffix as input for recursive calls.

Ask RF is $A[i..n]$ is splittable.

Describe how to compute it using a recurrence.

Define

$IsWord(i, j)$: is $A[i..j]$ A word

$Splittable(i)$: is $A[i..n]$ Splittable?

~~Splittable(i)~~

$$Splittable(i) = \begin{cases} True & \text{if } i > n \\ \bigvee_{j=i}^n (IsWord(i, j) \wedge Splittable(j+1)) & \end{cases}$$

or

otherwise

Want to know Splittable (1).