S.KIMMEL

Quiz!

X-tra: Show X^2 is not $O(n)$.

Announcements

Announcements:
- HW submitted through CANVAS (pdf)
  - Scanners
  - Type
- No late accepted
- Submit early to test
- New students

S.KIMMEL

Q: Create a Divide & Conquer Algorithm for Multiplication
(a, b n digit numbers, n is a power of 2)

Base Case: If 1 digit numbers, can do in $O(1)$ time

Divide: $$a = \overbrace{(a_{n-1} \cdots a_{n/2})}^{a'} \cdot 10^{n/2} + \overbrace{(a_{\frac{n}{2}-1} \cdots a_0)}^{a''}$$
$$b = \overbrace{(b_{n-1} \cdots b_{\frac{n}{2}})}^{b'} \cdot 10^{n/2} + \overbrace{(b_{n/2-1} \cdots b_0)}^{b''}$$

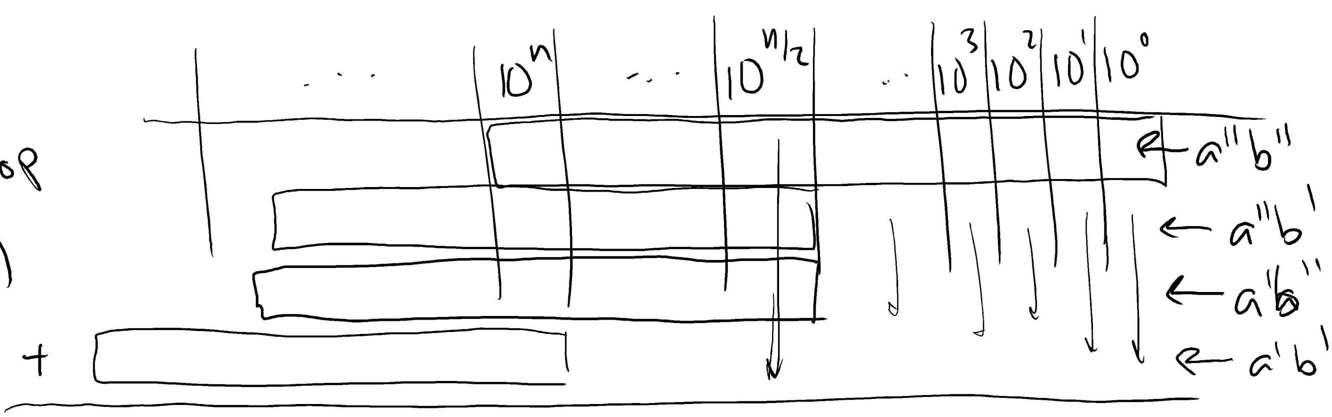
a', a'', b', b'' each $\frac{n}{2}$ digits!

e.g. $$5284 = 5 \times 10^3 + 2 \times 10^2 + 8 \times 10 + 4$$
$$= 52 \times 10^2 + 84$$

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

Conquer: Solve smaller problems: $a'b'$, $a''b'$, $a'b''$, $a''b''$
Combine: Add $a'b' \cdot 10^n + (a''b' + a'b'') 10^{n/2} + a''b''$

Use for loop $O(n)$ ops



S. KIMMEL

Doing Better: "Gauss's Trick"

Before: $a'b'\,10^n + (a'b'' + a''b')\,10^{n/2} + a''b''$

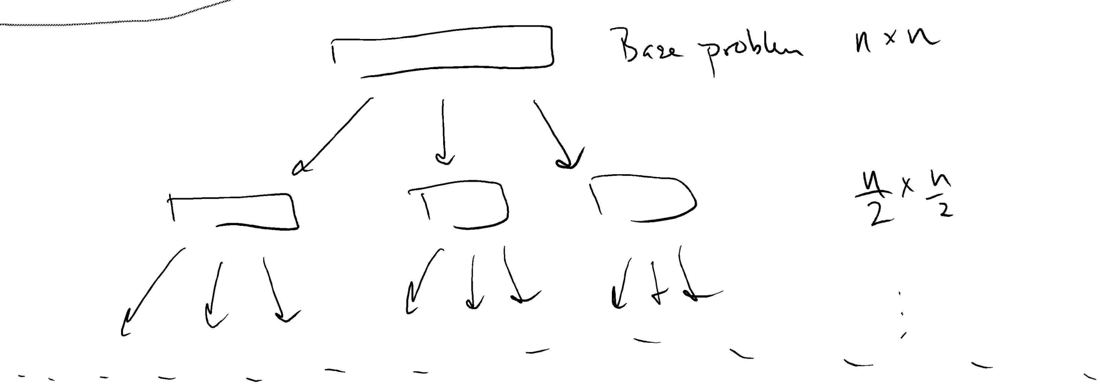
Notice
$$\underbrace{a'b'' + a''b'}_{\text{2 multiplications}} = \underbrace{(a' + a'')(b' + b'')}_{\substack{\text{1 new multiplication of}\\ \frac{n}{2}+1 \text{ digit \#'s}}} - \underbrace{a'b' - a''b''}_{\text{already perform these}}$$

If interested, shows $n/2+1$ digits as hard as $n/2$

Problem: Assumed only multiplying 2^k-digit numbers, but if $n = 2^k$, $\frac{n}{2} + 1 \neq 2^{k'}$

Instead: If d, e $\frac{n}{2}+1$ digit numbers, do

$$d \cdot e = \underbrace{(d_{n/2} \cdots d_0)(e_{n/2} \cdots e_0)}_{\text{Multiplying 2 } n/2\text{-digit numbers}} + \underbrace{d_{n/2}\, e \cdot 10^{n/2} + e_{n/2} \cdot d \cdot 10^{n/2} + d_{n/2}\, e_{n/2} \cdot 10^n}_{O(n) \text{ multiplications}}$$



S.KIMMEL

Q: How many base problems are implemented?

A: $n^{\log_3 2}$ B: $n^{\log_2 3}$ C: $n^{2\cdot\log_2 3}$ D: n^2

How many base problems? $\underbrace{3\cdot 3\cdot 3\cdots 3}_{\log_2 n \text{ times}} = 3^{\log_2 n} = \left(2^{\log_2 3}\right)^{\log_2 n} = \left(2^{\log_2 n}\right)^{\log_2 3}$

$= n^{\log_2 3}$

$\leq n^{1.59}$

Given a proposed algorithm, always need to answer 2 questions

1. Is the algorithm correct?
2. What is the asymptotic worst-case run time?

Q. Which proof method should be used to prove correctness of Karatsuba Multiplication algorithm

A) Proof by Contradiction B) Brute-force Search C) Proof by Contrapositive D) Proof by Induction

To analyze runtime, create "Recurrence Relation"

Let $T(n)$ be # of operations required to solve Karatsuba multiplication of n-digit #'s.

S. KIMMEL

Recurrence for Karatsuba

1. Base Case: $T(1) \le O(1)$ (use lookup table)

2. For $n > 1$:

$$T(n) \le \overbrace{3T\left(\frac{n}{2}\right)}^{\text{Divide}} + \underbrace{11n + 3}$$

\# of recursive calls

size of input to recursive call

Everything else the algorithm does

Dealing with $\frac{n}{2}+1$-digit multiplication

All the additions/subtractions of n-digit numbers

Recursive call initiation

Want $T(n)$ as a function of n, not as a function of $T\left(\frac{n}{2}\right)$.

2 Approaches

- Master Method
- Guess & Check