

CS3000: Algorithms & Data Jonathan Ullman

Lecture 3:

- Divide and Conquer: Karatsuba
- Solving Recurrences

Jan 13, 2020

Recap: Ask the Audience

- “**Big-Oh**” Notation: $f(n) = O(g(n))$ if there exists $c \in (0, \infty)$ and $n_0 \in \mathbb{N}$ such that $f(n) \leq c \cdot g(n)$ for every $n \geq n_0$.
- Which of these statements are true?
 - $3n^2 + n = O(n^3)$ TRUE
 - $n^3 = O(n^2)$ FALSE
 - $3n^2 + n = O(n^2)$ TRUE
 - $\log_2(n^2) = O(\log_2(n))$

$$3n^2 + n = O(n^2)$$

$$\exists n_0 \in \mathbb{N} \text{ s.t. } \forall n \geq n_0 \quad f(n) \leq c \cdot g(n)$$

$$3n^2 + n \leq 4n^2$$

$$3n^2 = O(n^2) \quad (\text{drop leading constant})$$

$$n = O(n) = O(n^2) \quad (\text{bigger exponent})$$

$$3n^2 + n = O(n^2) \quad (\text{add together terms})$$

$$f(n) = \log_2(n^2) \quad g(n) = \log_2(n)$$

$$\log_2(n^2) = O(\log_2(n))$$

$$\log_2(n^2) = 2 \cdot \log_2(n) = O(g(n))$$

$$3n^2 + n = O(n^3 + 100n \log n)$$

Integer Multiplication: Karatsuba's Algorithm

Addition

- Given n -digit numbers u, v output $u + v$

A diagram illustrating column addition. Two numbers, u and v , are shown vertically. u has digits 1, 2, 1, 3, 1, 5, 3, 4 from left to right. v has digits 2, 1, 1, 2, 2, 6, 2, 2. A horizontal line with a '+' sign separates the two numbers, and an '=' sign is at the bottom. Blue arrows and ovals indicate carries: a carry of 1 from the first column goes to the second; carries of 1 and 2 from the second column go to the third and fourth respectively; carries of 1 and 2 from the fourth column go to the fifth and sixth respectively; and a carry of 1 from the sixth column goes to the seventh.

$O(n)$ operations

Multiplication

- Given n -digit numbers u, v output $u \cdot v$

$$\begin{array}{r} & 1 & 2 \\ \times & 1 & 1 \\ \hline & 2 & 4 & 6 & 8 \\ + & 2 & 4 & 6 & 8 & 0 \\ + & 1 & 2 & 3 & 4 & 0 & 0 \\ + & 1 & 2 & 3 & 4 & 0 & 0 \\ \hline & 1 & 3 & 8 & 4 & 5 & 4 & 8 \end{array}$$

The diagram shows the multiplication of two 2-digit numbers, 12 and 11. The result is 138. Blue annotations illustrate the steps: a blue arrow points from the digit 3 in the product to the digit 3 in the second multiplicand; blue circles highlight the digits 3 and 2 in the second multiplicand, and the digits 4 and 2 in the product; blue underlines group the partial products 24, 68, and 24, and the final summand 12.

Running time of this alg is $\Theta(n^2)$

Divide and Conquer Multiplication

	1	2	3	4
x	1	1	2	2

	a	b
x	c	d

$$u = 10^2 \cdot 12 + 34$$

$$v = 10^2 \cdot 11 + 22$$

$$u = 10^{n/2} \cdot a + b$$

$$v = 10^{n/2} \cdot c + d$$

n is even

Divide and Conquer Multiplication

	a	b	$u = 10^{n/2} \cdot a + b$
x	c	d	$v = 10^{n/2} \cdot c + d$

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

- Four $n/2$ -digit mults, three n -digit adds
 - Multiplying by 10^n is “free” because it’s a shift
- Recurrence: $T(n) = 4T\left(\frac{n}{2}\right) + 3n$

Divide and Conquer Multiplication

- **Claim:** $T(n) \geq n^2$

$$\boxed{\begin{aligned} T(n) &= 4 \cdot T(n/2) + 3n \\ T(1) &= 1 \end{aligned}}$$

$$T(n) = 4 \cdot T\left(\frac{n}{2}\right) + 3n$$

$$\geq 4 \cdot \left(\frac{n}{2}\right)^2 + 3n$$

$$= 4 \cdot \frac{n^2}{4} + 3$$

$$\geq n^2$$

Karatsuba's Algorithm

	a	b	$u = 10^{n/2} \cdot a + b$
x	c	d	$v = 10^{n/2} \cdot c + d$

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

- Key Identity
 - $(b - a)(c - d) = ad + bc - ac - bd$

- Only three $n/2$ -digit mults (plus some adds)!

Compute $b-a$, $c-d$

Compute ac , bd , $(b-a)(c-d)$

Karatsuba's Algorithm

Karatsuba (u, v, n) :

If ($n = 1$) : Return $u \cdot v$ // Base Case

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

Write $u = 10^m \cdot a + b$, $v = 10^m \cdot c + d$

Let $e \leftarrow \text{Karatsuba}(a, c, m)$ // Recurse

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

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

Return $10^{2m} \cdot e + 10^m \cdot (e + f + g) + f$ // Merge

Correctness of Karatsuba

- **Claim:** The algorithm **Karatsuba** is correct

Running Time of Karatsuba

Karatsuba (u, v, n) :

If ($n = 1$) : Return $u \cdot v$

$\mathcal{O}(n)$

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

$3 \times T\left(\frac{n}{2}\right)$

Write $u = 10^m a + b$, $v = 10^m c + d$

{ Let $e \leftarrow \text{Karatsuba}(a, c, m)$
 $f \leftarrow \text{Karatsuba}(b, d, m)$
 $g \leftarrow \text{Karatsuba}(b-a, c-d, m)$

$\mathcal{O}(n)$

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

$$T(n) = 3 \times T\left(\frac{n}{2}\right) + C_n$$

Recursion Tree

$$T(n) = 3 \cdot T(n/2) + Cn$$
$$T(1) = C$$

Level

Problems

Work

0



Cn

1



$$3 \times C \left(\frac{n}{2}\right) = C \times \left(\frac{3}{2}\right) \times n$$

2



$$9 \times C \times \frac{n}{4} = C \times \frac{9}{4} \times n$$

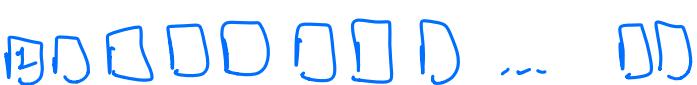
i

$$\underbrace{\left[n/2^i\right]}_{\vdots} \times 3^i$$

$$3^i \times C \times \frac{n}{2^i} = C \times \left(\frac{3}{2}\right)^i \times n$$

⋮

$\log_2(n)$



$$C \times \left(\frac{3}{2}\right)^{\log_2(n)} \times n$$

$$C \times \frac{3^{\log_2(n)}}{2^{\log_2(n)}} \times n = C \times 3^{\log_2(n)} = C \times n^{\log_2(3)}$$

$$\begin{aligned}
 \sum_{i=0}^{\log_2(n)} C \times \left(\frac{3}{2}\right)^i \times n &= C \times n \times \left(\sum_{i=0}^{\log_2(n)} \left(\frac{3}{2}\right)^i \right) \\
 &= C \times n \times C' \times \left(\frac{3}{2}\right)^{\log_2(n)} \\
 &= \Theta\left(n^{\log_2(3)}\right)
 \end{aligned}$$

$$\log_2(3) \approx 1.59$$

Geometric Series

- Series $S = \sum_{i=0}^{\ell-1} r^i$

$$S = 1 + r + r^2 + \dots + r^{\ell-1}$$

$$rS = r + r^2 + r^3 + \dots + r^{\ell-1} + r^\ell$$

$$(1-r)S = 1 - r^\ell$$

$$S = \frac{1-r^\ell}{1-r} = \frac{r^\ell - 1}{r - 1}$$

- Solution:

$$\bullet r \neq 1, S = \frac{1-r^\ell}{1-r} = \frac{r^\ell - 1}{r - 1} \quad r > 1 : S = \Theta(r^\ell)$$

$$\bullet r = 1, S = \ell \quad r < 1 : S = \Theta(1)$$

$$r = 1 : S = \Theta(\ell)$$

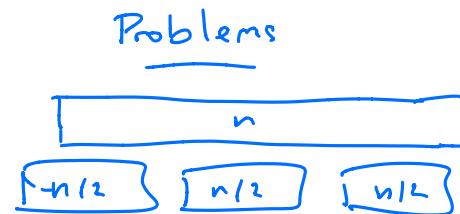
Karatsuba Wrapup

- Multiply n digit numbers in $O(n^{1.59})$ time
 - Improves over naïve $O(n^2)$ time algorithm
 - **Fast Fourier Transform:** multiply in $\approx O(n \log n)$ time
- Divide-and-conquer approach
 - Uses a clever algebraic trick to split
 - **Key Fact:** adding is faster than multiplying
- Prove correctness via induction
- Analyze running time via recursion tree
 - $T(n) = 3T(n/2) + Cn$

Ask the Audience!

- $T(n) = 3T(n/2) + n^2$
- $T(1) = 1$

Level



Work

$$n^2$$

$$3 \times \left(\frac{n}{2}\right)^2 = \frac{3}{4} \times n^2$$

$$3^i \times \left(\frac{n}{2^i}\right)^2 = \left(\frac{3}{4}\right)^i \times n^2$$

$$\left(\frac{3}{4}\right)^{\log_2(n)} \times n^2$$

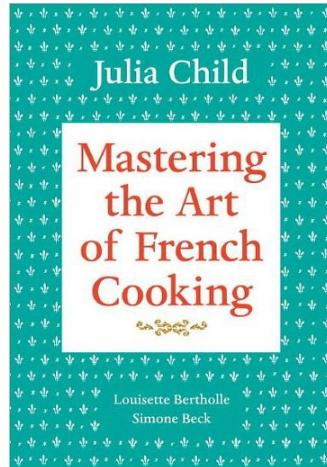
$$n^2 \cdot \sum_{i=0}^{\log_2(n)} \left(\frac{3}{4}\right)^i = n^2 \cdot \left(\frac{1 - \left(\frac{3}{4}\right)^{\log_2(n)+1}}{1 - \frac{3}{4}} \right) \leq n^2 \times 4 \\ = O(n^2)$$

Solving Recurrences: “The Master Theorem”

The “Master Theorem”

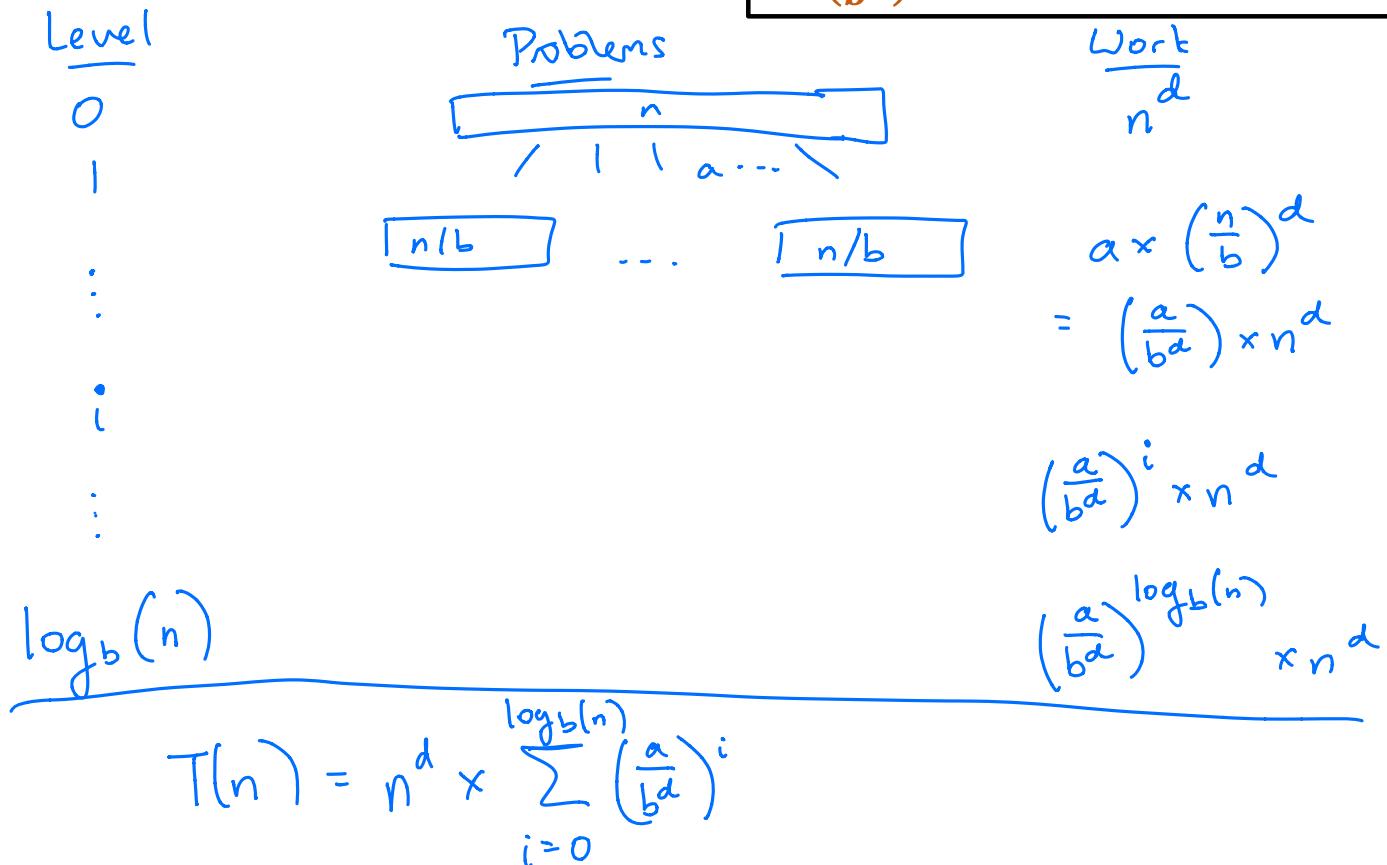
a, b, d independent of n

- Generic divide-and-conquer algorithm:
 - Split into a pieces of size $\frac{n}{b}$ and merge in time $O(n^d)$
- Recipe for recurrences of the form:
 - $T(n) = a \cdot T(n/b) + Cn^d$



Recursion Tree

- $T(n) = aT(n/b) + n^d$
- $\left(\frac{a}{b^d}\right) > 1$



Recursion Tree

- $T(n) = aT(n/b) + n^d$
- $\left(\frac{a}{b^d}\right) > 1$

Level

0

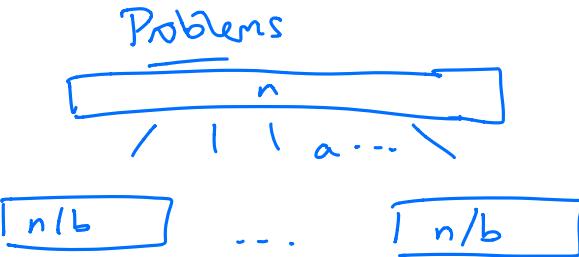
1

:

i

:

$\log_b(n)$



Work
 n^d

$$\begin{aligned} & a \times \left(\frac{n}{b}\right)^d \\ &= \left(\frac{a}{b^d}\right) \times n^d \end{aligned}$$

$$\left(\frac{a}{b^d}\right)^i \times n^d$$

$$\left(\frac{a}{b^d}\right)^{\log_b(n)} \times n^d$$

$$\begin{aligned} T(n) &= n^d \times \Theta\left(\left(\frac{a}{b^d}\right)^{\log_b(n)}\right) = n^d \times \Theta\left(\frac{a^{\log_b(n)}}{n^d}\right) \\ &= \Theta(a^{\log_b(n)}) = \Theta(n^{\log_b(a)}) \end{aligned}$$

Recursion Tree

- $T(n) = aT(n/b) + n^d$
- $\left(\frac{a}{b^d}\right) = 1$

Mergesort : $a=2$
 $b=2$
 $d=1$

$$\begin{aligned} T(n) &= n^d \times \sum_{i=0}^{\log_b(n)} 1^i \\ &= n^d \times \log_b(n) = \Theta(n^d \log(n)) \end{aligned}$$

Recursion Tree

- $T(n) = aT(n/b) + n^d$
- $\left(\frac{a}{b^d}\right) < 1$

Ex: $a = 3$
 $b = 2$
 $d = 2$

$$\begin{aligned} T(n) &= n^d \times \sum_{i=0}^{\log_b(n)} \left(\frac{a}{b^d}\right)^i \\ &\stackrel{L}{=} n^d \times \sum_{i=0}^{\infty} \left(\frac{a}{b^d}\right)^i = n^d \times \frac{1}{1 - \frac{a}{b^d}} = \Theta(n^d) \end{aligned}$$

The “Master Theorem”

- Recipe for recurrences of the form:
 - $T(n) = a \cdot T(n/b) + cn^d$
- Three cases:
 - $\left(\frac{a}{b^d}\right) > 1 : T(n) = \Theta(n^{\log_b a})$
 - $\left(\frac{a}{b^d}\right) = 1 : T(n) = \Theta(n^d \log n)$
 - $\left(\frac{a}{b^d}\right) < 1 : T(n) = \Theta(n^d)$

Ask the Audience!

- Use the Master Theorem to Solve:

$$\bullet T(n) = 16 \cdot T\left(\frac{n}{4}\right) + n^2$$

$$\bullet T(n) = 21 \cdot T\left(\frac{n}{5}\right) + n^2$$

$$\bullet T(n) = 2 \cdot T\left(\frac{n}{2}\right) + 1$$

$$\bullet T(n) = 1 \cdot T\left(\frac{n}{2}\right) + 1$$

The “Master Theorem”

- **Even More General:** all recurrences of the form
 - $T(n) = a \cdot T(n/b) + f(n)$
- Three cases:
 - $f(n) = O(n^{(\log_b a) - \varepsilon})$:
 - $T(n) = \Theta(n^{\log_b a})$
 - $f(n) = \Theta(n^{\log_b a})$:
 - $T(n) = \Theta(f(n) \cdot \log n)$
 - $f(n) = \Omega(n^{(\log_b a) + \varepsilon})$ AND $af\left(\frac{n}{b}\right) \leq Cf(n)$ for $C < 1$
 - $T(n) = \Theta(f(n))$