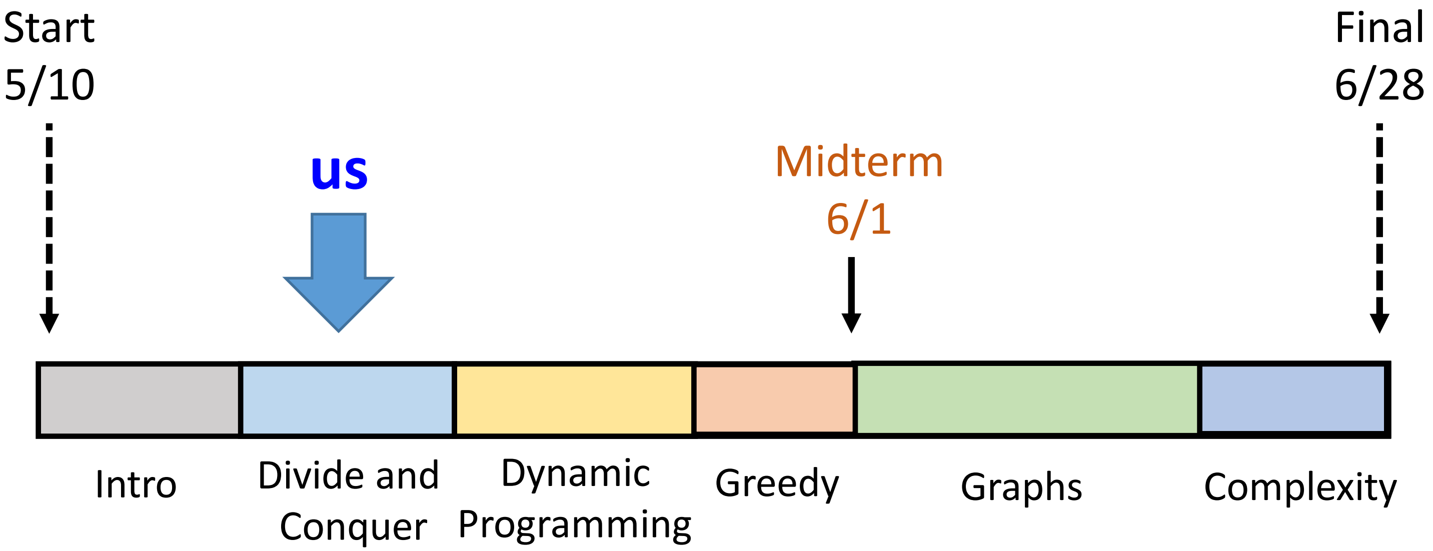
CS3000: Algorithms & Data
Drew van der Poel

Lecture 6

- Divide & Conquer: Karatsuba's
- Master Theorem

May 18, 2021

Outline



Last class: divide and conquer: Merge sort

Next class: divide and conquer: Selection (Median)

Multiplication

- Given *n*-digit numbers x, y output $x \cdot y$

				1	2	3	~~4~~ 8
			x	1	1	2	2
	0	0	0	2	4	6	8
+	0	0	2	4	6	8	0
+	0	1	2	3	4	0	0
+	1	2	3	4	0	0	0
	1	3	8	4	5	4	8

n^2 mults

$\le n-1$ adds.

$n-1$ adds. or $\Theta(n)$-digit #'s

↓

$n-1$ additions of $\Theta(n)$-digit #s

$cn \cdot (n-1) = cn^2 - cn$

Running Time: $O(n^2)$

$cn^2 + c'n^2 - c''n =$

Divide and Conquer Multiplication

$n = 4$

	1	2	3	4
x	1	1	2	2

$1234 = (12 \cdot 10^2) + 34$

$1122 = (11 \cdot 10^2) + 22$

general n

	a	***b***
x	***c***	***d***

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

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

$$\underbrace{1\,2}_{a}\;\underbrace{3\,4}_{b}$$

$$ab \cdot cd$$

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

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

Divide and Conquer Multiplication

	$\boldsymbol{a}$	$\boldsymbol{b}$
x	$\boldsymbol{c}$	$\boldsymbol{d}$

$$\boldsymbol{x} = 10^{n/2}\boldsymbol{a} + \boldsymbol{b}$$

$$\boldsymbol{y} = 10^{n/2}\boldsymbol{c} + \boldsymbol{d}$$

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

$$= \underline{10^{\textcircled{n}}}\,\underline{\boldsymbol{ac}} \;\underline{+}\; \underline{10^{n/2}}(\underline{\boldsymbol{ad}} \underline{+} \underline{\boldsymbol{bc}}) \underline{+} \underline{\boldsymbol{bd}}$$

- Four $n/2$-digit mults., three n-digit adds & some shifts
- Recurrence: $T(n) = \underline{4T\left(\frac{n}{2}\right)} + \underline{\Theta(n)}$

Divide and Conquer Multiplication

$H(n)$ $\quad T(n) = \Omega(n^2)$

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

$$T(n) = 4 \cdot T(n/2) + Cn$$
$$T(1) = 1$$

Base: $H(1) \to T(1) = 1 \geq 1^2$ ✓

$H(\frac{k}{2})$

Ind: $H(1) \land \ldots \land H(k-1) \longrightarrow H(k)$

WTS $H(k) \to T(k) \geq k^2$

$\downarrow$

$4T(\frac{k}{2}) + Ck$

by IH:
$T(\frac{k}{2}) \geq (\frac{k}{2})^2$

$\geq 4(\frac{k}{2})^2 + Ck = k^2 + Ck$

$T(k) \geq k^2 + Ck \geq k^2$

□

Karatsuba's Algorithm

	a	b
x	c	d

$$\boldsymbol{x} = 10^{n/2}\boldsymbol{a} + \boldsymbol{b}$$

$$\boldsymbol{y} = 10^{n/2}\boldsymbol{c} + \boldsymbol{d}$$

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

- Key Identity
  - $(\boldsymbol{b} - \boldsymbol{a})(\boldsymbol{c} - \boldsymbol{d}) =$ $bc - \underline{bd} - \underline{ac} + ad$

$$\boxed{\underline{(b-a)(c-d)} + \underline{bd} + \underline{ac} = ad + bc}$$

- Only three $n/2$-digit mults (plus some adds & shifts)!
  - 1. ac
  - 2. bd
  - 3. $(b-a)(c-d)$

Karatsuba’s Algorithm

```
Karatsuba(x,y,n):
   If (n = 1): Return $x \cdot y$                 // Base Case

   Let $m \leftarrow \lceil n/2 \rceil$                    // Split
   Write $x = 10^m a + b$,  $y = 10^m c + d$

   Let e ← Karatsuba(a,c,m)                 // Recurse
       f ← Karatsuba(b,d,m)
       g ← Karatsuba(b-a,c-d,m)

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

Correctness of Karatsuba

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

H(n): Karat(x, y, n) is correct $\forall x, y \in \mathbb{N}$ w/ $n \in \mathbb{N}$ digits

Base: H(1) → trivial

Correctness of Karatsuba

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

Inductive: $\overbrace{\underline{H(1)\ \&\ H(2)\ \&\ \dots\ \&\ H(k-1)}}^{IH} \to H(k)$

$a,\ b,\ c,\ d,\ b-a,\ c-d$ all have $< k$ digits

$$\therefore \left.\begin{array}{l} e = ac \\ f = bd \\ g = (b-a)(c-d) \end{array}\right\} \text{by } IH$$

$$\text{Karat}(x, y, n) = 10^{k} e + 10^{k/2}(e+f+g) + f$$

$$= 10^{k} ac + 10^{k/2}(bc+ad) + bd \qquad \square$$

Running Time of Karatsuba

$T(1) = O(1)$

$T(n) = \underline{3}T(\underline{\tfrac{n}{2}}) + \underline{O(n)}$

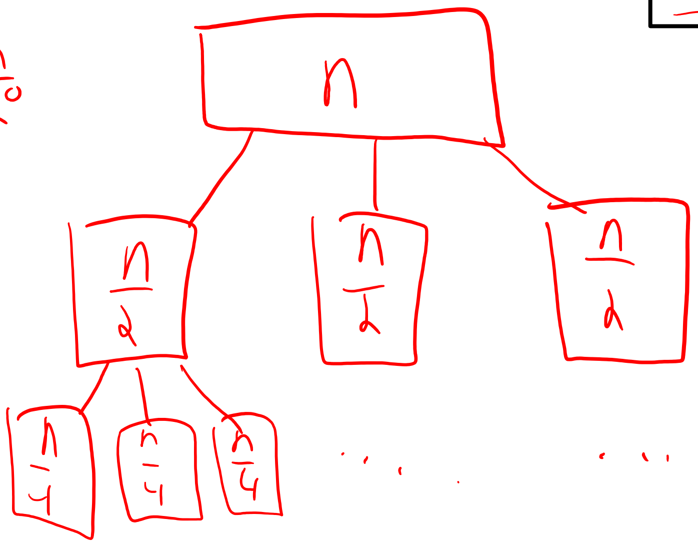
# of digits

```
Karatsuba(x,y,n):
   If (n = 1): Return $x \cdot y$

   Let $m \leftarrow \lceil n/2 \rceil$
   Write $x = 10^m a + b$,  $y = 10^m c + d$

   Let e ← 1 Karatsuba(a,c,m)
       f ← 2 Karatsuba(b,d,m)
       g ← 3 Karatsuba(b-a,c-d,m)

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

4 $O(n)$-digit additions

Recursion Tree

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



level	size	#	work @ level
0	$n = \frac{n}{2^0}$	$1 = 3^0$	cn
1	$\frac{n}{2} = \frac{n}{2^1}$	$3 = 3^1$	$\frac{cn}{2} + \frac{cn}{2} + \frac{cn}{2}$ $= \frac{3cn}{2}$
2	$\frac{n}{4} = \frac{n}{2^2}$	$9 = 3^2$	$9\left(\frac{cn}{4}\right)$
i	$\frac{n}{2^i}$	3^i	$3^i\left(\frac{cn}{2^i}\right)$

$\boxed{\log_2 n}$

$$\frac{n}{2^i} = 1 \to n = 2^i \to \log_2 n = i$$

$$\sum_{i=0}^{\log_2 n} 3^i\left(\frac{cn}{2^i}\right) = cn \sum_{i=0}^{\log_2 n}\left(\frac{3}{2}\right)^i$$

Geometric Series

- Series (r≠1, r>0) $S = \boxed{\sum_{i=0}^{\ell} r^i}$

$$S \ = 1 + r + r^2 + \cdots + r^\ell$$

$$rS = \qquad r + r^2 + \cdots + r^\ell + r^{\ell+1}$$

$$S\,(1\ - r) = S\ - rS = 1\ - r^{\ell+1}$$

$$S\,(r\ - 1) = rS\ - S = r^{\ell+1} - 1$$

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

- $S = \Theta(1)$ when $r < 1$
  $S = \underline{\Theta(r^\ell)}$ when $r > 1$

$$cn \sum_{i=0}^{\log_2 n} \left(\frac{3}{2}\right)^i \qquad r = 3/2$$

$$cn\, c' \left(\frac{3}{2}\right)^{\log_2 n}$$

$$= c\cancel{n}c' \frac{3^{\log_2 n}}{\cancel{n}}$$

$$= c c' \ 3^{\log_2 n}$$

$$= c c' \ n^{\log_2 3}$$

$$= O\left(n^{1.59}\right)$$

Karatsuba Wrapup

- Multiply n digit numbers in $\boxed{O(n^{1.59})}$ 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
    - $T(n) = 3T(n/2) + Cn$
- We will generally assume our inputs have *O(1)* digits

Solving Recurrences:
“The Master Theorem”

The "Master Theorem"

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

$\rightarrow T(n) = 2T\left(\frac{n}{2}\right) + Cn$

Mergesort

a: 2 b: 2 d: 1

Karatsuba

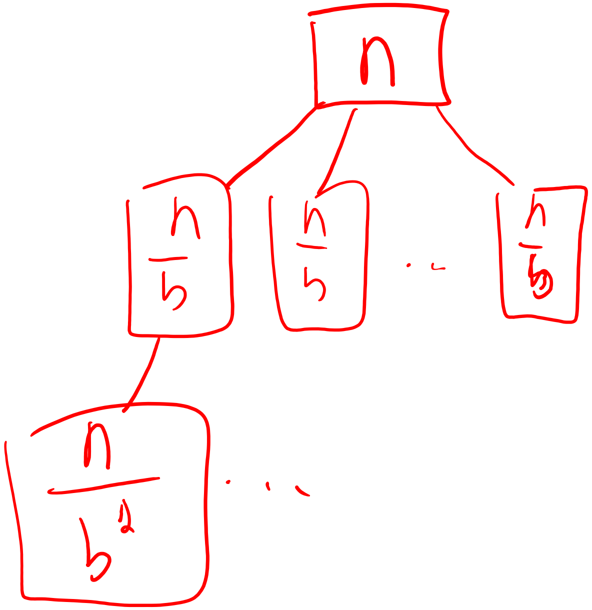
a: 3 b: 2 d: 1

Recursion Tree

$T(1) = c$

- $T(n) = \boldsymbol{a}T(n/\boldsymbol{b}) + Cn^{\boldsymbol{d}}$

level	size	# of pieces	work @ level
0	n	1	Cn^d
1	n/b	a	$a\ c\left(\frac{n}{b}\right)^d$
2	n/b^2	a^2	$a^2\ c\left(\frac{n}{b^2}\right)^d$
i	n/b^i	a^i	$a^i\ c\left(\frac{n}{b^i}\right)^d$
$\log_b n$			



$$\frac{n}{b^i} = \underline{\underline{1}} \rightarrow n = b^i$$

$$\log_b n = i$$

$$\sum_{i=0}^{\log_b n} a^i\ c\left(\frac{n}{b^i}\right)^d = Cn^d \sum_{i=0}^{\log_b n} \left(\frac{a}{b^d}\right)^i$$

How much work?

Total work: $$Cn^d \sum_{i=0}^{\log_b n} \underbrace{\left(\frac{a}{b^d}\right)}^i$$

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

$S = \Theta(1)$ when $r < 1$

$S = \Theta(r^\ell)$ when $r > 1$

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

$$Cn^d \; c' \left(\frac{a}{b^d}\right)^{\log_b n} = C\cancel{n^d}\, c' \, \frac{a^{\log_b n}}{\cancel{n^d}}$$

$$= C c' a^{\log_b n} = \Theta(n^{\log_b a})$$

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

$$Cn^d \underbrace{(\log_b n + 1)} = Cn^d \log_b n + Cn^d$$

$$= \Theta(n^d \log n)$$

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

$$Cn^d \; c' = \Theta(n^d)$$

The “Master Theorem”

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

- Problems: counting students, stable matching, sorting, n-digit mulitiplication

- Alg. techniques: divide & conquer

- Analysis: asymptotic analysis, recursion trees, **Master Thm.**

- Proof techniques: (strong) induction, contradiction

Ask the Audience!

$\left(\frac{a}{b^d}\right) > 1 : T(n) = \Theta\left(n^{\boxed{\log_b a}}\right)$

$\left(\frac{a}{b^d}\right) = 1 : T(n) = \Theta\left(\underline{n^d \log n}\right)$

$\left(\frac{a}{b^d}\right) < 1 : T(n) = \Theta\left(n^d\right)$

- Use the Master Theorem :
  - $T(n) = 16 \cdot T\left(\frac{n}{4}\right) + n^2$

    $a = 16 \quad b = 4 \quad d = 2$ $\qquad \frac{16}{4^2} = 1 \qquad \Theta(n^2 \log n)$
  - $T(n) = 21 \cdot T\left(\frac{n}{5}\right) + n^2$

    $a = 21 \quad b = 5 \quad d = 2$ $\qquad \frac{21}{5^2} < 1 \qquad \Theta(n^2)$
  - $T(n) = 2 \cdot T\left(\frac{n}{2}\right) + 1$

    $a = 2 \quad b = 2 \quad d = 0$ $\qquad \frac{2}{2^0} > 1 \qquad \Theta(n)$
  - $T(n) = 1 \cdot T\left(\frac{n}{2}\right) + 1$

    $a = 1 \quad b = 2 \quad d = 0$ $\qquad \frac{1}{2^0} = 1 \qquad \Theta(\log n)$

The "Master Theorem"

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

(Reduce)

Divide-and-Conquer: Binary Search

Binary Search

Is 28 in this list? If so, where?

Sorted:

2	3	8	11	15	17	28	42

A

Binary Search

Is 28 in this list? If so, where?

Sorted:	2	3	8	11	15	17	28	42	A

- Problems: counting students, stable matching, sorting, n-digit mulitiplication, **array searching**
- Alg. techniques: divide & conquer
- Analysis: asymptotic analysis, recursion trees, Master Thm.
- Proof techniques: (strong) induction, contradiction

Binary Search

```
Search(A,t):
  // A[1:n] sorted in ascending order
  Return BS(A,1,n,t)

BS(A,ℓ,r,t):
  If(ℓ > r): return FALSE

  m ← $\ell + \left\lfloor \frac{r-\ell}{2} \right\rfloor$

  If(A[m] = t): return m
  ElseIf(A[m] > t): return BS(A,ℓ,m-1,t)
  Else: return BS(A,m+1,r,t)
```

T(n):

T(1):

Running Time Analysis

$$\boxed{\begin{array}{l} T(n) = T(n/2) + C \\ T(1) = C \end{array}}$$