CS4102 Algorithms
Spring 2019

Warm up

Simplify:

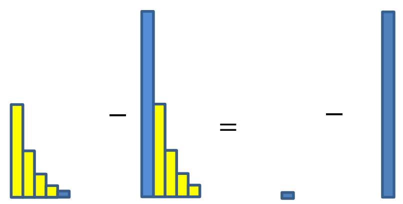
$$(1 + a + a^2 + a^3 + a^4 + \cdots + a^L)(a - 1) = ?$$

$$(a + a^2 + a^3 + a^4 + a^5 + \cdots + a^L + a^{L+1}) +$$
$$(-a - a^2 - a^3 - a^4 - a^5 - \cdots - a^L - 1) =$$
$$a^{L+1} - 1$$

$$\sum_{i=0}^{L} a^i = \frac{a^{L+1} - 1}{a - 1}$$

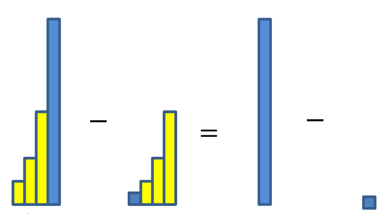
1

Finite Geometric Series $a < 1$



2

Finite Geometric Series $a > 1$



3

Today's Keywords

- Divide and Conquer
- Recurrences
- Merge Sort
- Karatsuba
- Tree Method

4

CLRS Readings

- Chapter 4

5

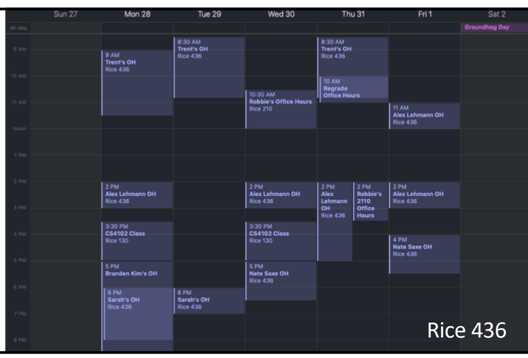
Homeworks

- Hw1 due Wed, January 30 at 11pm
  - Start early!
  - Written (use Latex!) – Submit BOTH pdf and zip!
  - Asymptotic notation
  - Recurrences
  - Divide and Conquer

6

Homework Help Algorithm

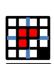
- Algorithm: How to ask a question about homework (efficiently)
  1. Check to see if your question is already on piazza
  2. If it's not on piazza, ask on piazza
  3. Look for other questions you know the answer to, and provide answers to any that you see
  4. TA office hours
  5. Instructor office hours
  6. Email, set up a meeting



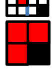
8

Divide and Conquer*

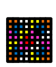
- **Divide**:
  - Break the problem into multiple subproblems, each smaller instances of the original
- **Conquer**:
  - If the subproblems are "large":
    - Solve each subproblem recursively



  - If the subproblems are "small":
    - Solve them directly (base case)



- **Combine**:
  - Merge together solutions to subproblems



*CLRS Chapter 4

9

Analyzing Divide and Conquer

1. Break into smaller subproblems
2. Use recurrence relation to express recursive running time
3. Use asymptotic notation to simplify

- **Divide:** $D(n)$ time,
- **Conquer:** recurse on small problems, size s
- **Combine:** $C(n)$ time
- **Recurrence:**
  - $T(n) = D(n) + \sum T(s) + C(n)$

10

Recurrence Solving Techniques


Tree *get a picture of recursion*

?✓ Guess/Check *guess and use induction to prove*


"Cookbook" *MAGIC!*


Substitution *substitute in to simplify*

11

Merge Sort

- **Divide:**
  - Break n-element list into two lists of $^n/_2$ elements
- **Conquer:**
  - If $n > 1$:
    - Sort each sublist recursively
  - If $n = 1$:
    - List is already sorted (base case)
- **Combine:**
  - Merge together sorted sublists into one sorted list

12

Merge

- **Combine:** Merge sorted sublists into one sorted list
- We have:
  - 2 sorted lists (L_1, L_2)
  - 1 output list (L_{out})

While (L_1 and L_2 not empty):
    If $L_1[0] \leq L_2[0]$:
        L_{out}.append(L_1.pop())
    Else:
        L_{out}.append(L_2.pop())
L_{out}.append(L_1)
L_{out}.append(L_2)

$O(n)$

13

Analyzing Merge Sort

1. Break into smaller subproblems
2. Use recurrence relation to express recursive running time
3. Use asymptotic notation to simplify

- **Divide:** 0 comparisons
- **Conquer:** recurse on 2 small subproblems, size $\frac{n}{2}$
- **Combine:** n comparisons
- **Recurrence:**
  - $T(n) = 2T(\frac{n}{2}) + n$

14

Recurrence Solving Techniques

 Tree

Guess/Check

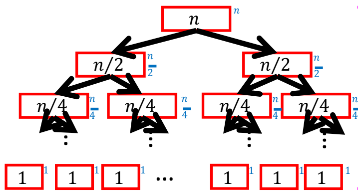
 "Cookbook"

Substitution

15

Tree method

$T(n) = 2T(\frac{n}{2}) + n$



$\Rightarrow n$ total / level

$\log_2 n$ levels of recursion

$= n \sum_{i=1}^{\log_2 n} 1 = n \log_2 n$

$T(n) = \sum_{i=1}^{\log_2 n} n = n \log_2 n$

16

Multiplication

- Want to multiply large numbers together

$$\begin{array}{r} 4\,1\,0\,2 \\ \times\, 1\,8\,1\,9 \\ \hline \end{array}$$

n-digit numbers

- What makes a "good" algorithm?
- How do we measure input size?
- What do we "count" for run time?

17

"Schoolbook" Method

How many total multiplications?

n-digit numbers

$$\begin{array}{r} 4\,1\,0\,2 \\ \times\, 1\,8\,1\,9 \\ \hline 3\,6\,9\,1\,8 \\ 4\,1\,0\,2 \\ 3\,2\,8\,1\,6 \\ +\ 4\,1\,0\,2 \\ \hline 7\,4\,6\,1\,5\,3\,8 \end{array}$$

n mults
n mults
n mults
n mults

n levels $\Rightarrow \theta(n^2)$

18

Divide and Conquer method

1. Break into smaller subproblems

$$\boxed{a}\,\boxed{b} = 10^{\frac{n}{2}}\,\boxed{a} + \boxed{b}$$
$$\times\,\boxed{c}\,\boxed{d} = 10^{\frac{n}{2}}\,\boxed{c} + \boxed{d}$$

$$10^n(\boxed{a}\times\boxed{c}) +$$
$$10^{\frac{n}{2}}(\boxed{a}\times\boxed{d} + \boxed{b}\times\boxed{c}) +$$
$$(\boxed{b}\times\boxed{d})$$

4102 = 4100 + 02
= 100 · 41 + 02
= 10^2 · 41 + 02
n = 4, $\frac{n}{2}$ = 2

19

Divide and Conquer Multiplication

- **Divide**:
  - Break n-digit numbers into four numbers of $^n/_2$ digits each (call them a, b, c, d)
- **Conquer**:
  - If $n > 1$:
    - Recursively compute ac, ad, bc, bd
  - If $n = 1$: (i.e. one digit each)
    - Compute ac, ad, bc, bd directly (base case)
- **Combine**:
  - $10^n(ac) + 10^{\frac{n}{2}}(ad + bc) + bd$

20

Divide and Conquer method

2. Use recurrence relation to express recursive running time

$$10^n(\boxed{ac}) + 10^{\frac{n}{2}}(\boxed{ad} + \boxed{bc}) + \boxed{bd}$$

Recursively solve

$$T(n) = 4T\left(\frac{n}{2}\right) + 5n$$

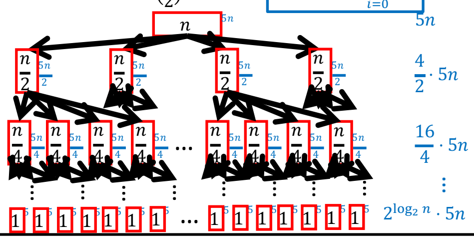
21

Divide and Conquer method

3. Use asymptotic notation to simplify

$$T(n) = 4T\left(\frac{n}{2}\right) + 5n$$

$$T(n) = 5n \sum_{i=0}^{\log_2 n} 2^i$$



22

Divide and Conquer method

3. Use asymptotic notation to simplify

$$T(n) = 4T\left(\frac{n}{2}\right) + 5n$$

$$T(n) = 5n \sum_{i=0}^{\log_2 n} 2^i$$

$$T(n) = 5n\frac{2^{\log_2 n+1} - 1}{2 - 1}$$

$$T(n) = 5n(2n - 1) \ = \Theta(n^2)$$

$\sum_{i=0}^{L} a^i = \frac{a^{L+1}-1}{a-1}$

$L = \log_2 n \ , \ a = 2$

$2^{\log_2 n+1} = 2 \cdot 2^{\log_2 n}$

$= 2n$

23

Karatsuba

1. Break into smaller subproblems

$$\boxed{a}\boxed{b} = 10^{\frac{n}{2}}\boxed{a} + \boxed{b}$$

$$\times \ \boxed{c}\boxed{d} = 10^{\frac{n}{2}}\boxed{c} + \boxed{d}$$

$$10^{n}(\boxed{a} \times \boxed{c}) +$$

$$10^{\frac{n}{2}}(\boxed{a} \times \boxed{d} + \boxed{b} \times \boxed{c}) +$$

$$(\boxed{b} \times \boxed{d})$$

24

Karatsuba

$$10^n(ac) + 10^{\frac{n}{2}}(ad + bc) + bd$$

Can't avoid these

This can be simplified

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

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

Two multiplications

One multiplication

25

Karatsuba

2. Use recurrence relation to express recursive running time

$$10^n(ac) + 10^{\frac{n}{2}}\left((a+b)(c+d) - ac - bd\right) + bd$$

Recursively solve

$$T(n) = 3T\left(\frac{n}{2}\right) + 8n$$

26

Karatsuba

- **Divide**:
  - Break n-digit numbers into four numbers of $n/2$ digits each (call them a, b, c, d)
- **Conquer**:
  - If $n > 1$:
    - Recursively compute $ac, bd, (a+b)(c+d)$
  - If $n = 1$:
    - Compute $ac, bd, (a+b)(c+d)$ directly (base case)
- **Combine**:
  - $10^n(ac) + 10^{\frac{n}{2}}\left((a+b)(c+d) - ac - bd\right) + bd$

27

a b
× c d

Karatsuba Algorithm

1. Recursively compute: $ac, bd, (a+b)(c+d)$
2. $(ad+bc) = (a+b)(c+d) - ac - bd$
3. Return $10^n(ac) + 10^{\frac{n}{2}}(ad+bc) + bd$

$$T(n) = 3T\left(\frac{n}{2}\right) + 8n$$

Pseudo-code

1. x = Karatsuba(a,c)
2. y = Karatsuba(a,d)
3. z = Karatsuba(a+b,c+d)-x-y
4. Return 10^nx +$10^{n/2}$z+y

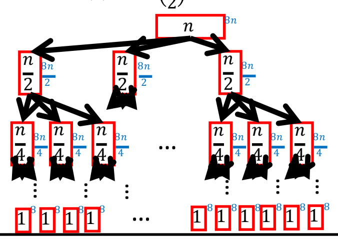
28

Karatsuba

3. Use asymptotic notation to simplify

$$T(n) = 3T\left(\frac{n}{2}\right) + 8n$$

$$T(n) = 8n \sum_{i=0}^{\log_2 n} (3/2)^i$$



$8 \cdot 1n$

$\frac{8}{2} \cdot 3n$

$\frac{8}{4} \cdot 9n$

⋮

$\frac{8}{2^{\log_2 n}} \cdot 3^{\log_2 n} n$

29

Karatsuba

3. Use asymptotic notation to simplify

$$T(n) = 3T\left(\frac{n}{2}\right) + 8n$$

$$T(n) = 8n \sum_{i=0}^{\log_2 n} (3/2)^i$$

$$T(n) = 8n \frac{(3/2)^{\log_2 n + 1} - 1}{3/2 - 1}$$

Math, math, and more math…(on board, see lecture supplement)

30

Karatsuba

$$T(n) = 8n \frac{\left(\frac{3}{2}\right)^{\log_2 n + 1} - 1}{\frac{1}{2}}$$

$$= 16n\left(\left(\tfrac{3}{2}\right)^{\log_2 n+1} - 1\right)$$

$$= 16n\left(\left(2^{\log_2 3 - 1}\right)^{\log_2 n + 1}\right) - 16n$$

$$= 16n\left(2^{\log_2 3 \log_2 n - \log_2 n + \log_2 3 - 1}\right) - 16n$$

$$= 16n\left(\left(2^{\log_2 n}\right)^{\log_2 3} \cdot 2^{-\log_2 n} \cdot 2^{\log_2 3} \cdot 2^{-1}\right) - 16n$$

$$3 = 2^{\log_2 3}$$

$$\frac{2^{\log_2 3}}{2} = 2^{\log_2 3 - 1}$$

31

Karatsuba

3. Use asymptotic notation to simplify

$$T(n) = 3T\left(\frac{n}{2}\right) + 8n$$

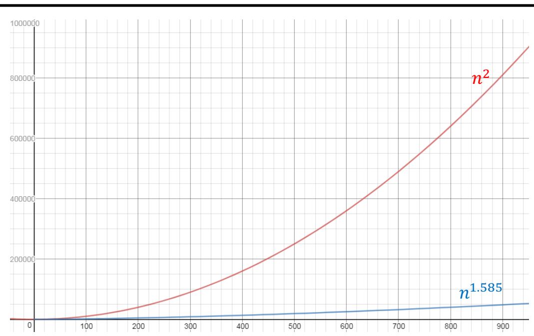
$$T(n) = 8n \sum_{i=0}^{\log_2 n} \left({}^3/_2\right)^i$$

$$T(n) = 8n \frac{\left({}^3/_2\right)^{\log_2 n + 1} - 1}{{}^3/_2 - 1}$$

Math, math, and more math…(on board, see lecture supplement)

$$T(n) = 24\left(n^{\log_2 3}\right) - 16n = \Theta\left(n^{\log_2 3}\right) \approx \Theta\left(n^{1.585}\right)$$

32



33