

## Administrivia, Lecture 5

- HW #2 was assigned on Sunday, January 20. It is due on Thursday, January 31.
  - Please use the correct edition of the textbook!
- We have a quiz this Thursday (20 minutes) at the beginning of lecture
- The DQMaxMin problem is treated in CLRS 9.1
- Grading will drop the single worst HW or quiz.

# Randomized Quicksort (7.4.2/8.4.2)

- Randomized Quicksort (yet another treatment, this time straight from CLRS)
  - $i = \text{Random}(p, r)$
  - swap  $A[p] \leftrightarrow A[i]$
  - partition  $A(p, r)$

- Average analysis = **Expected** runtime

$$T(n) = \frac{1}{n} \sum_{k=1}^{n-1} [T(k) + T(n-k)] + \Theta(n) = \frac{2}{n} \sum_{k=1}^{n-1} T(k) + \Theta(n)$$

- solving recurrence  $\equiv T(n) \leq a \times n \times \log n + b$
- by induction, it is true for  $k < n$ , need to show

$$T(n) \leq \frac{2}{n} \sum_{k=1}^{n-1} [ak \log k + b] + \Theta(n) \leq \frac{2a}{n} \sum_{k=1}^{n-1} k \log k + 2b + \Theta(n)$$

# Average-Case Time Complexity of QSort

$$\begin{aligned}
 T(n) &\leq \frac{2a}{n} \sum_{k=1}^{n-1} k \log k + 2b + \Theta(n) \\
 &\leq \frac{2a}{n} \left( \frac{1}{2} n^2 \log n - \frac{1}{8} n^2 \right) + 2b + \Theta(n) \\
 &\leq a n \log n + b + (\Theta(n) + b - \frac{a}{4} n) \\
 &\leq a n \log n + b
 \end{aligned}$$

$$\sum_{k=1}^{n-1} k \log k \leq \frac{1}{2} n^2 \log n - \frac{1}{8} n^2$$

$$\begin{aligned}
 \sum_{k=0}^{n-1} k \log k &\leq \sum_{k=1}^{n/2-1} k \log k + \sum_{k=n/2}^{n-1} k \log k \leq (\log n - 1) \sum_{k=1}^{n/2-1} k + \log n \sum_{k=n/2}^{n-1} k \\
 &= \log n \sum_{k=1}^{n-1} k + \sum_{k=1}^{n/2-1} k \leq \frac{1}{2} n(n-1) \log n - \frac{1}{2} \left( \frac{n}{2} - 1 \right) \frac{n}{2}
 \end{aligned}$$

# Selection

- Recall: best pivot in QSort is the median
  - It is too expensive to find the median  $\mathcal{P}$  we use a random element as the pivot instead
- This leads to SELECTION:
  - Given a list  $L$  and a number  $k$ ,  $\text{Select}(L, k)$  returns the  $k^{\text{th}}$  smallest element of  $L$
  - e.g., when  $k = |L|/2$   $\mathcal{P}$   $\text{Select}(L, k)$  returns the median
- What is an efficient algorithm?
  - Sorting + finding  $k^{\text{th}}$  smallest  $\rightarrow O(n \log n)$  (can do better)
- D/Q Recursion: use the “pivot” from QSort  $\rightarrow$  “Randomized-Select” (CLRS 9.2, page 186)
  - if  $i < k$ , look in right part for  $(k-i)^{\text{th}}$  smallest
  - if  $i > k$ , look in left part for  $k^{\text{th}}$  smallest

# Analysis of Randomized-Select (9.2)

- **Average-case** time complexity

$$\begin{aligned} T(n) &= \frac{1}{n} \sum_{k=1}^{n-1} T(\max\{k, n-k\}) + O(n) \\ &= \frac{2}{n} \sum_{k=n/2}^{n-1} T(k) + O(n) \end{aligned}$$

- Inductive hypothesis:  $T(n) \leq c \cdot n$   $\rightarrow$  Want :

$$T(n) = \frac{2}{n} \sum_{k=n/2}^{n-1} ck + O(n) \leq \frac{2c}{n} \left( \sum_{k=1}^{n-1} k - \sum_{k=1}^{n/2-1} k \right) + O(n) \leq cn$$

**Need for induction step**

# Analysis of Randomized-Select (9.2, p. 188)

• Want: 
$$T(n) = \frac{2}{n} \sum_{k=n/2}^{n-1} ck + O(n) \leq \frac{2c}{n} \left( \sum_{k=1}^{n-1} k - \sum_{k=1}^{n/2-1} k \right) + an$$
$$\leq \frac{2c}{n} \left( \frac{(n-1)n}{2} - \frac{(n/2-2)(n/2-1)}{2} \right) + an$$
$$= \frac{2c}{n} \left( \frac{(n^2 - n)}{2} - \frac{(n^2/4 - 3n/2 + 2)}{2} \right) + an$$
$$= \frac{c}{n} \left( \frac{3n^2}{4} + \frac{n}{2} - 2 \right) + an = c \left( \frac{3n}{4} + \frac{1}{2} - \frac{2}{n} \right) + an$$
$$\leq \frac{3cn}{4} + \frac{c}{2} + an = cn - \left( \frac{cn}{4} - \frac{c}{2} - an \right)$$

average-case

$$\leq cn \Rightarrow cn/4 - c/2 - an \geq 0 \Rightarrow n(c/4 - a) \geq c/2$$

Choose  $c$  s.t.  $c/4 - a > 0 \Rightarrow c > 4a \Rightarrow T(n) = O(n)$

# Randomized Selection

- Worst case:
  - $\Omega(n^2)$  as in QSort analysis
- **But: Suppose** we can guarantee a “good” pivot, e.g., such that  $n/4 \leq i \leq 3n/4$ 
  - ▷ subproblem size  $\leq 3n/4$
- Let  $s(n) \equiv$  time to find good pivot
  - ▷  $t(n) \leq s(n) + cn + t(3n/4)$

$s(n)$  = find pivot

$cn$  = pivot, make subproblem

$t(3n/4)$  = solve subproblem

# Randomized Selection

- **Suppose** further:

$S(n) \leq dn$  for some  $d$

$$P \quad t(n) \leq (c+d)n + t(3n/4)$$

- **Claim:**  $t(n) \leq kn$  for some  $k$  would follow

- Again, we use Constructive Induction (“Substitution”):

- Induction Hypothesis:  $t(m) \leq km$  for  $m \leq n-1$

- Induction Step:  $t(n) \leq (c+d)n + k(3n/4)$

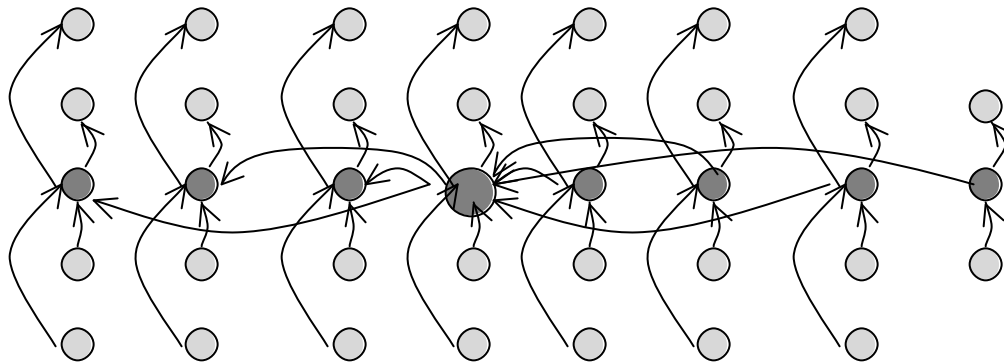
$$= (c+d+3k/4)n$$

which we want to be equivalent to  $t(n) \leq kn$

- But this is true if  $k \geq 4(c+d)$

# Worst-Case Linear-Time Selection (9.3)

- Of mostly theoretical interest
- How to find a “good” median in linear time
  - divide into groups of 5  $c_1n$
  - find the median of each group  $c_2n/5$
  - find the median of medians, recursively  $T(n/5)$
- Pivoting around this element gives at worst 3:7 split



# Worst-Case Linear-Time Selection

- Blum-Floyd-Pratt-Rivest-Tarjan, 1973
  - divide L into groups of 5  $c_1n$
  - find the median of each group  $c_2n/5$
  - Let S = array of medians of each group  $c_3n$
  - find median of medians (i.e., median of S) recursively;  
use as pivot  $t(n/5)$
- Total work:  $t(n) \leq kn + t(n/5) + t(7n/10 + d)$
- In general:  $a(n) \leq kn + a(\hat{I}_1n) + a(\hat{I}_2n) + a(\hat{I}_3n)$ ,  
with  $k, \hat{I}_1, \hat{I}_2, \hat{I}_3$  constants s.t.  $\hat{I}_1 + \hat{I}_2 + \hat{I}_3 < 1$ ,  
then  $a(n) \leq c^*n$  for some  $c^* \rightarrow t(n) = O(n)$

## More Careful Exposition

- Recursive call has size  $\leq 7n/10 + 6$  (CLRS p. 191)
  - $T(n) \hat{=} \theta(1)$ ,  $n \leq M$  ( $M = 140$ )
  - $T(n) \leq T(n/5) + T(7n/10 + 6) + O(n)$ ,  $n > M$
- Prove linearity by substitution
  - Assume  $T(n) \leq cn$  for some constant  $c$ , and for all  $n \leq M$
  - $T(n) \leq cn/5 + c(7n/10 + 6) + an$ 
    - $\leq cn/5 + c + 7cn/10 + 6c + an$
    - $= 9cn/10 + 7c + an$
    - $= cn + (-cn/10 + 7c + an)$
  - The induction step goes through if  $-cn/10 + 7c + an \leq 0$ 
    - $\rightarrow$  pick  $c$  large enough so that  $c(n/10 - 7) > an \forall n > M$

# D/Q for Arithmetic

- Multiplying Large Integers (Karatsuba-Ofman)
  - $A = a_0 + r^1 a_1 + \dots + r^{n-1} a_{n-1}$ ,  $r \equiv \text{radix}$
  - “classic” approach  $\mathcal{P} \Theta(n^2)$  work
- Can we apply D/Q?
  - Let  $n = 2s$ ,  $r = 10 \equiv \text{radix}$
  - $AB = xz + 10^s(wz + xy) + 10^{2s}wy$
  - $T(n) = 4T(n/2) + \theta(n)$   $\mathcal{P} a = 4, b = 2$  in Master Method
  - $T(n) \hat{=} \Theta(n^2)$
  - Need to reduce # subproblems, i.e., want  $a < 4$
- Observation:  $r' = (w+x)(y+z) = wy + (wz+xy) + xz$ 
  - $r' \rightarrow (w+x)(y+z)$
  - $p \rightarrow wy$
  - $q \rightarrow xz$
  - return  $10^{2s}p + 10^s(r'-p-q) + q$
  - $T(n) \hat{=} O(n^{\log_2 3}) = O(n^{1.59})$

# Matrix Multiplication

- $A = [\dots]$  ,  $B = [\dots]$  are  $n \times n$  matrices
- $a_{11}$ ,  $a_{12}$ , etc are  $n/2 \times n/2$  submatrices
- $M = AB = [\dots]$ 
  - where  $m_{11} = a_{11}b_{11} + a_{12}b_{21}$  etc.
  - Evaluation requires 8 multiplies, 4 adds
- $T(n) = 8T(n/2) + O(n) \hat{=} \Theta(n^3)$

# Strassen's Matrix Multiplication

$$p_1 = (a_{21} + a_{22} - a_{11})(b_{22} - b_{12} + b_{11})$$

$$p_2 = a_{11}b_{11}$$

$$p_3 = a_{12}b_{21}$$

$$p_4 = (a_{11} - a_{21})(b_{22} - b_{12})$$

$$p_5 = (a_{21} + a_{22})(b_{12} - b_{11})$$

$$p_6 = (a_{12} - a_{21} + a_{11} - a_{22})b_{22}$$

$$p_7 = a_{22}(b_{11} + b_{22} - b_{12} - b_{21})$$

$$AB_{11} = p_2 + p_3$$

$$AB_{12} = p_1 + p_2 + p_5 + p_6$$

$$AB_{21} = p_1 + p_2 + p_4 + p_7$$

$$AB_{22} = p_1 + p_2 + p_4 + p_5$$

- $T(n) \hat{=} \Theta(n^{2.81})$  // 7 multiplies, 24 adds
  - Can get to 15 adds

# Break

- Question: Given the adjacency matrix of an undirected graph, describe a fast algorithm that finds all triangles in the graph.
  - (Why am I asking this now?)

# A Computational Geometry Excursion

- Given simple polygon  $P$ , is  $x$  in  $P$ ?  $\theta(n)$  ?
- Given  $n$  points in the plane, draw a simple  $n$ -gon through them
  - $\theta(n \log n)$  ?
- Given  $n$  points in the plane, return their convex hull
  - $\theta(n \log n)$  ?
  - Does it matter whether we must return vertices or return the polygon?
- Given  $n$  points in the plane, return the closest pair
  - $\theta(n \log n)$  ?
- Given  $n$  h-, v- segments, return all intersections
  - $O(f(n) + g(k))$ , where  $k = \text{output size}$ ?
  - plane-sweep  $\equiv$  “inductive” paradigm

# DQ Closest Pair

- Naïve  $\Theta(n^2)$
- D/Q Approach:  $\Theta(n \log n)$ 
  - (1) Split into two pointsets  $S_1, S_2$   
by x-coord  $\equiv$  natural order
  - (2) Find closest pair distances  $d_1, d_2$  in  $S_1, S_2$  respectively  
without loss of generality, can assume  $d_1 < d_2$
  - (3) Merge the solutions  
Do we have to check all  $s_1$ - $s_2$  pairs,  $s_1 \in S_1, s_2 \in S_2$ ?  
What if there are lots of points in middle strip?
  - Key: Step (3)  
Observation: There are at most  $O(1)$  points in middle strip with  $\Delta y \leq d_1$

# DQ Closest Pair

- (1) sort by x-coord    **tool: one-time sort**  $O(n \log n)$
- (2) solve subproblems of size  $n/2$      $2T(n/2)$
- (3) eliminate points outside strips     $O(n)$
- (4) sort by y-coord within strips     $O(n \log n)$
- (5) compare each point to  $\leq 6$  neighbors     $O(n)$
- Complexity
  - $O(n \log n) + T(n) = 2T(n/2) + O(n \log n)$
  - $T(2) = 1$
  - $T(n) \hat{=} O(n \log^2 n) \dots$
  - Not what we wanted: what went wrong?

# DQ Closest Pair

- Why keep on sorting by y-coordinate?
  - (1') sort by x-coordinate, and sort by y-coordinate
- Manber textbook: **Tool: “Strengthening the I.H.”**
  - Given  $<n$  points in the plane, we can find smallest distance and output the set sorted by y-coordinate
  - N.B.: Udi Manber is Chief Scientist at Yahoo
- Note: We exploited geometry
  - e.g., what is maximum possible degree of a vertex in the minimum spanning tree?
    - If the problem is geometric?
    - If the problem is non-geometric?

# Convex Hull

- Goal: Return the hull points in order
- What is a lower bound? **Tool: reduction**
- Many approaches: see CLRS 33.3
  - Stretch  $\theta(n^2)$
  - Wrap  $\theta(n^2)$
  - Graham Scan  $\theta(n \log n)$
  - D/Q  $\theta(n \log n)$ 
    - left-right split
    - find convex hulls
    - merge convex hulls
    - **Tool: Do you need to sort in order to bisect (split) ???**

# Yet Another Geometric Problem

- Given two convex polygons, return intersection
- Note 1: Intersection of  $(A \cup B), (C \cup D)$   
= Union of  $(A \cap B), (A \cap D), (B \cap C), (B \cap D)$ 
  - I.e., intersection of unions is union of intersections
- Note 2: How is a convex polygon represented?
  - Cyclic order
- Note 3: How difficult is it to compute the intersection of a 3-gon/4-gon, or 4-gon/4-gon?
  - $O(1)$  time
  - **Tool:** Solving any bounded-size instance =  $O(1)$  time !
- **Can you get an  $O(n)$  algorithm?**