Beginning of Lec 3

Suppose Worst-case Runtime of

Alg 1 is $O(n^2)$

Alg 2 is $O(n \log n)$

Which is better?

Don't really know

This is like asking which is smaller

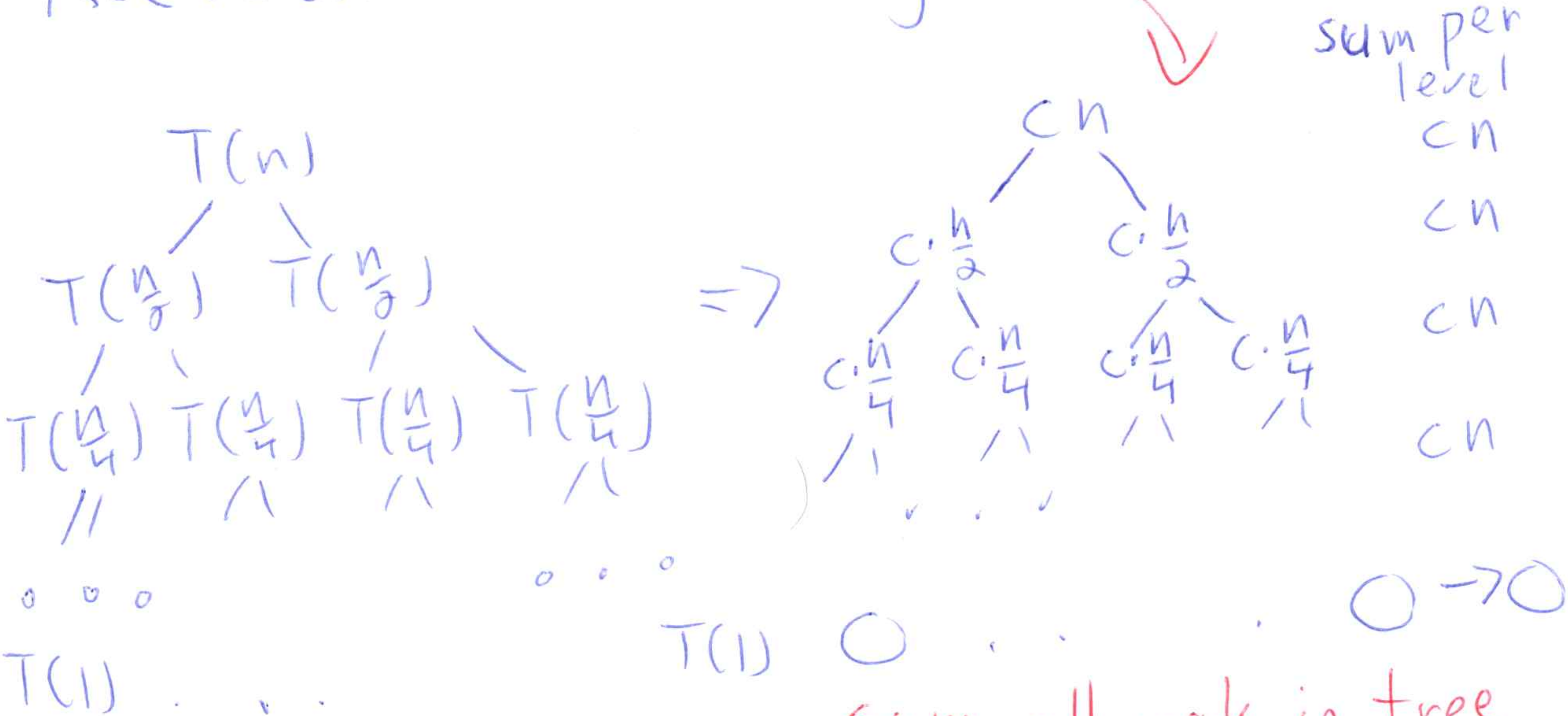
$x \leq 5$ or $y \leq 10$

To compare algs, we really need tight bounds

$\Theta(n^2)$ vs $\underline{\Theta(n \log n)}$

-> this is better

Recursion Tree for MergeSort



sum all work in tree

$n(T(1)) + c(n + 2\frac{n}{2} + 4\frac{n}{4}) + \cdots + \frac{n}{2}\left(\frac{n}{n/2}\right))$

$\Rightarrow cn(\log n) + nd$

$\in \Theta(n \log n)$

Track recursive calls

Substitution Method

n even: $T(n) = 2T(\frac{n}{2}) + n - 1$

$\leq \cancel{2} \cdot \boxed{c \cdot \frac{n}{\cancel{2}} \log \frac{n}{2}} + n - 1$ by I.H.

$= cn \log \frac{n}{2} + n - 1$

$= cn(\log n - 1) + n - 1$

$= cn \log n - cn + n - 1$

$\leq cn \log n$ if $c \geq 1$

$\log \frac{n}{2} = \log n + \log \frac{1}{2}$

$= \log n - \log 2$

$= \log n - 1$

n odd:

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

$$\leq c\left(\frac{n-1}{2}\right)\log\left(\frac{n-1}{2}\right) + c\left(\frac{n+1}{2}\right)\log\left(\frac{n+1}{2}\right) + n - 1$$

$$\vdots$$

$$\leq cn\log n \quad \text{for } c \geq \_\_, \ n \geq \_\_$$

difficult

Fact $\boxed{\log\left(\frac{n+1}{2}\right) < \log\left(\frac{n}{2}\right) + 1 \ \forall n \geq}$

Substitution - Changing the Guess

- $T(n) \le cn$: Is the guess wrong?

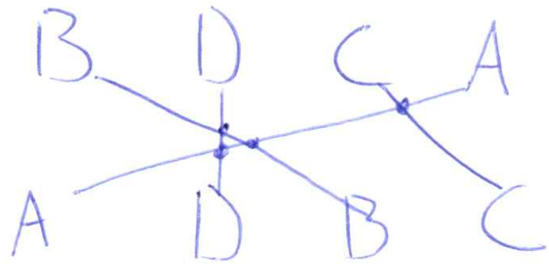
No, consider n is a power of 2

$$T(n) = 2T(\tfrac{n}{2}) + 1$$
$$= 4T(\tfrac{n}{4}) + 2 + 1$$
$$\vdots$$
$$= 2^k T\left(\frac{n}{2^k}\right) + \left(2^{k-1} + \dots + 2 + 1\right) \quad // \; n = 2^k$$

$T(1) = 1$

$$= 2^k + 2^{k-1} + \dots + 2 + 1$$
$$= 2^{k+1} - 1 \quad // \text{sum of powers of } 2$$
$$= 2n - 1$$

Counting Inversions



Pairs inverted: 4 out of 6 — ranks are not really close

BD BA DA CA

Brute force: check all $\binom{n}{2}$ pairs $\in O(n^2)$

Does sorting help? Doesn't seem to.

Divide & Conquer

Given a list $a_1, \dots, a_n$, count # inversions

- divide list in two: $m = \lceil \frac{n}{2} \rceil$

  $A = a_1 \dots a_m \quad B = a_{m+1} \dots a_n$

- Recursively count # inversions in each half
  $\Rightarrow$ returns r_A, r_B // counts
- Combine results: $r_A + r_B + r$

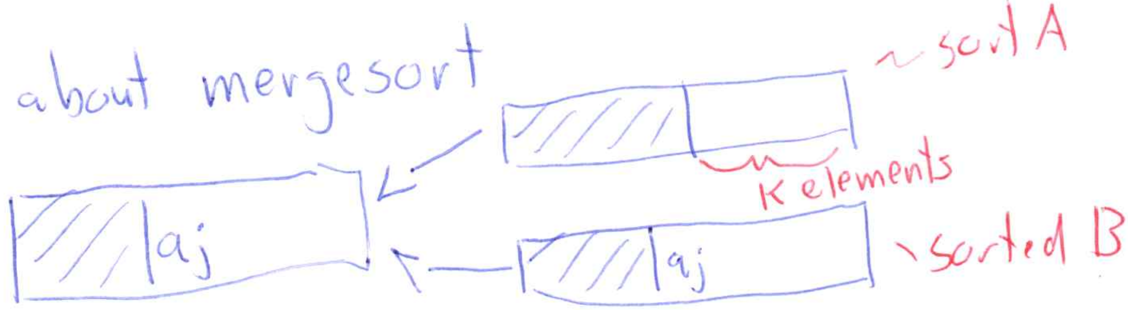
r = # inversions with one element in A and one in B
= # pairs a_i, a_j with $a_i \in A$, $a_j \in B$ $a_i > a_j$

How do we find r?

Count: for each $a_j \in B$ the # larger elements in $A \sim r_j$

Then $r = \sum_{a_j \in B} r_j$

Think about mergesort



When a_j is ~~merged~~ output to merged list: $r_j \leftarrow K$

Alg
sort-and-count (L) returns sorted L, # inversions

- divide L into A, B (half of L each)
- $(r_A, A) \leftarrow$ sort-and-count(A
  (r_B, B)
- $r \leftarrow 0$
- Do merge of A and B
  when element of B is moved to output
  $r \leftarrow r + \#$ elements remaining in A
- return $(r_A + r_B + r$, merged list)

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

similar to mergesort

Better Algs?

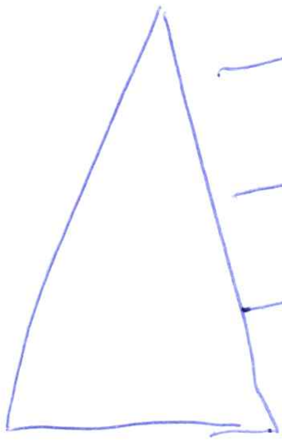
$O(n \log n / (\log \log n))$ 1989

$O(n \sqrt{\log n})$ 2010 Timothy Chan et al

using techniques / model

where sorting is $o(n \log n)$

Master Theorem - Recursion Tree



$$\xrightarrow{\Sigma} cn^k = cn^k\left(\frac{a}{b^k}\right)^0$$

$$\xrightarrow{\Sigma} ac\left(\frac{n}{b}\right)^k = cn^k\left(\frac{a}{b^k}\right)^1$$

$$\xrightarrow{\Sigma} a^2c\left(\frac{n}{b^2}\right)^k = cn^k\left(\frac{a}{b^k}\right)^2$$

$$\vdots$$

How many levels?

$$\frac{n}{b^i} = 1 \Rightarrow i = \log_b n$$

$$T(n) = a\underbrace{T\left(\frac{n}{b}\right)}_{} + cn^k$$

$$= a\left[aT\left(\frac{n}{b^2}\right) + c\left(\frac{n}{b}\right)^k\right] + cn^k$$

$$= a^2T\left(\frac{n}{b^2}\right) + ac\left(\frac{n}{b}\right)^k + cn^k$$

$$= a^3T\left(\frac{n}{b^3}\right) + a^2c\left(\frac{n}{b^2}\right)^k + ac\left(\frac{n}{b}\right)^k + cn^k$$

$$\vdots$$

$$= a^{\log_b n}T(1) + \sum_{i=0}^{\log_b n - 1} a^i c\left(\frac{n}{b^i}\right)^k$$

$$= n^{\log_b a}T(1) + cn^k\sum_{i=0}^{\log_b n - 1}\left(\frac{a}{b^k}\right)^i$$

• If $a < b^k$ (i.e. $\log_b a < k$)

then $\sum \left(\frac{a}{b^k}\right)^i$ is a geometric series with $\frac{a}{b^k} < 1$

$\Rightarrow \sum$ is constant, so

$$T(n) = n^{\log_b a} T(1) + \Theta(n^k)$$

$$T(n) = \Theta(n^k)$$

• If $a = b^k$ then $\displaystyle\sum_{i=0}^{\log_b n - 1} \left(\frac{a}{b^k}\right)^i = \sum_{i=0}^{\log_b n - 1} 1 = \Theta(\log_b n) \in \Theta(\log n)$

So, $T(n) = n^{\log_b a} T(1) + cn^k(O(\log n))$

$$T(n) = \Theta(n^k \log n)$$

• If $a > b^k$ then $\displaystyle\sum_{i=0}^{\log_b n - 1} \left(\frac{a}{b^k}\right)^i$ is a geometric series with $\frac{a}{b^k} > 1$

so last term dominates.

$$\boxed{\sum_{i=0}^{k-1} x^i = \frac{x^k - 1}{x - 1} \in \Theta(x^k) \text{ if } x > 1}$$

$$T(n) = n^{\log_b a} T(1) + \underbrace{\Theta\left(n^k \left(\frac{a}{b^k}\right)^{\log_b n}\right)}$$

$$\Theta\left(a^{\log_b n} \cdot \frac{n^k}{(b^{\log_b n})^k}\right) \leftarrow n^k \text{, log property}$$

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

$$T(n) \in \Theta(n^{\log_b a})$$

Multiplying Large Numbers

School Method:

$$
\begin{array}{r}
{}^{2}6\,{}^{2}6\,\cancel{7} \\
1\,2\,3\,4 \\
\hline
2\,6\,6\,8 \\
2\,0\,0\,1\,0 \\
1\,3\,3\,4\,0\,0 \\
6\,6\,7 \\
\hline
8\,2\,3\,0\,7\,8
\end{array}
$$

~1956 N^2 conjectured this was the best · Kolmogorov

D & C Method $T(n) = 4T(\frac{n}{2}) + \underbrace{O(n)}_{\text{additions \& shifts}}$

Centrality of Matrix Multiplication

Reduce: Triangular Matrix Inversion to Matrix Mult

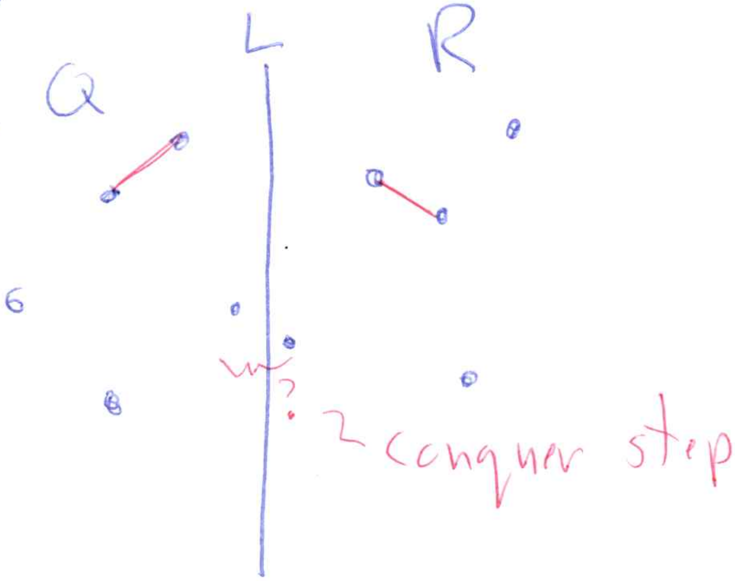
$$T = \left[\begin{array}{c|c} T_1 & U \\ \hline & T_2 \end{array}\right] \qquad T^{-1} = \left[\begin{array}{c|c} T_1^{-1} & -T_1^{-1} U T_2^{-1} \\ \hline & T_2^{-1} \end{array}\right]$$

Reduce: Matrix Mult to Triangular Matrix Inversion

$$\left[\begin{array}{c|c|c} I_n & A & \\ \hline & I_n & B \\ \hline & & I_n \end{array}\right]^{-1} = \left[\begin{array}{c|c|c} I_n & -A & AB \\ \hline & I_n & -B \\ \hline & & I_n \end{array}\right]$$

Closest Pair

D&C

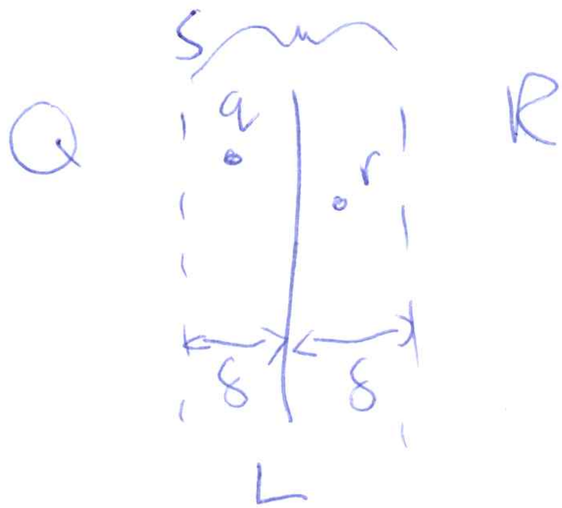


Let δ = min distance of $\begin{cases} \text{closest pair in } Q \\ \text{closest pair in } R \end{cases}$

Must check pairs $q \in Q, r \in R$ with $d(q,r) < \delta$

Claim: such points satisfy $d(q,L) < \delta$
and $d(r,L) < \delta$

<u>Pf</u>: Otherwise horizontal distance $\geq \delta$
so distance $\geq \delta$.



Let S = points in this vertical strip of width 2δ

We restrict our search to S
- may still be all n points!
- close to being 1 dimensional ~ a little bit wider

Sort by y-coord
- do not do this in each recursive call
- sort once at beginning, $O(n \log n)$
  and extract sublist of points in $O(n)$
  => sublist will be sorted.

* Similar implementation can be used for Kd-trees instead of QuickSelect

---

Alg Overview

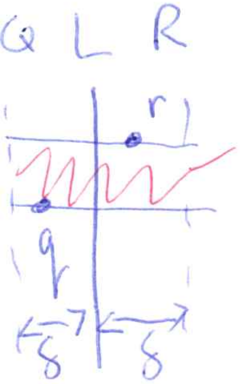
$X \leftarrow$ points sorted by x-coord
$Y \leftarrow$ points sorted by y-coord

closest(X, Y) returns distance between closest pair of points

$L \leftarrow$ dividing line (middle of X)
Extract X_Q, X_R ~ sorted by x list of points in region
Y_Q, Y_R ~ sorted by y
$\delta_Q \leftarrow$ closest(X_Q, Y_Q), $\delta_R \leftarrow$ closest(X_R, Y_R)
$\delta \leftarrow \min\{\delta_Q, \delta_R\}$
Find S vertical strip of width 2δ
$Y_S \leftarrow$ S sorted by y-coord (Extract from Y)

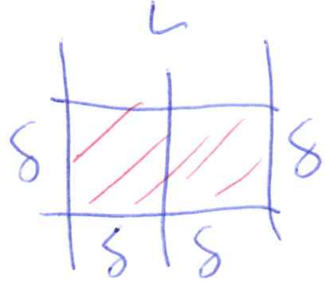
Now, what do we do with S, Y_S?

Hope: if $q, r \in S$ where $q \in Q$, $r \in R$
and $d(q,r) < \delta$
then q and r are near each other in the sorted S, Y_S.

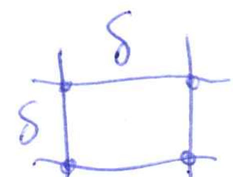


hope not many points here

so q is close to r in Y_S

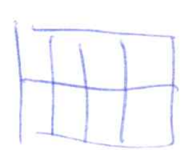
Claim: 

At most 8 points here



How many points can you put within δ of each other on each side of L?

- a $\delta \times \delta$ square has ≤ 4 points

OR $\delta \times 2\delta$ can be tiled with 8 squares $\frac{\delta}{2} \times \frac{\delta}{2}$ 

- each square has ≤ 1 point since $\sqrt{2\left(\frac{\delta}{2}\right)^2} = \frac{\delta}{\sqrt{2}} < \delta$

can prove 6 instead of 8

$\Rightarrow$ q and r are at most 8 positions apart in S

To check for pair $< \delta$:

- For each $s \in Y_s$ check distance with next 7 points in Y_s

Y_S: $\underset{q}{s_1 \ldots s_i} \; \underbrace{s_{i+1} \ldots s_{i+7}}_{\text{look here for } r} \ldots$

$\Rightarrow O(7n)$ comparisons

Analysis: Sort by x, y $O(n \log n)$

then $T(n) = 2T(\frac{n}{2}) + cn \in O(n \log n)$

- Preparata & Shamos
- can find closest neighbours for all points in $O(n \log n)$
  - nearest neighbours graph $a \to b$ b is a's closest neighbour
    - graph has no cycles, no crossings, max degree 6

Quick Select

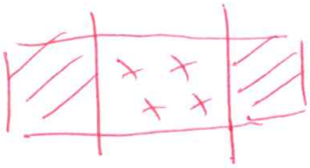
- runtime based on where pivot falls

average

$$T(n) \le \begin{cases} cn + \frac{1}{2}T(n) + \frac{1}{2}T(\lfloor \frac{3n}{4} \rfloor) & n \ge 2 \\ d & n = 1 \end{cases}$$

$\frac{1}{2}T(n)$: half time at ends

$\frac{1}{2}T(\lfloor \frac{3n}{4} \rfloor)$: half time near middle



$\in \Theta(n)$

Worst-case $T(n) = T(n-1) + cn \in \Theta(n^2)$

BFPRT worst-case $\Theta(n)$

- Blum, Floyd, Pratt, Rivest, Tarjan 1973

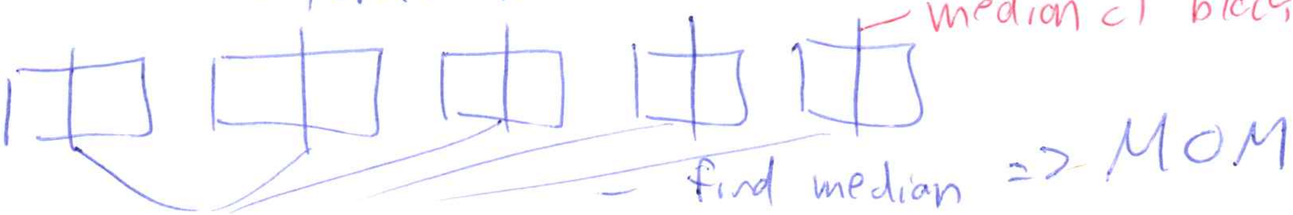
$n = 10r + 5 + \theta$ where $r \ge 1$ and $0 \le \theta \le 9$

Why θ? • want odd # of groups of 5

$\Rightarrow 2r+1$ groups, $r = \frac{n-5-\theta}{10}$

MOM - median of medians

Pivot: • find medians of blocks of 5 items
• take median of medians



r blocks have median < MOM

$\Rightarrow$ 3r elements < MOM

+2 from block containing MOM

Max size of subproblem

$n-(3r+2)-1$ ← don't include median of medians

$= n-3\left(\frac{n-5-\theta}{10}\right)-2-1$

$= \frac{10n}{10}+\frac{-3n+15+3\theta}{10}-\frac{30}{10}$

$= \frac{7n-15+3\theta}{10} \leq \left\lfloor \frac{7n+12}{10} \right\rfloor$ since $\theta \leq 9$

$$\Rightarrow T(n) \leq \begin{cases} T\left(\left\lfloor \frac{n}{5} \right\rfloor\right)+T\left(\frac{7n+12}{10}\right)+\Theta(n), & n \geq 15 \\ d & , n \leq 14 \end{cases}$$

Can be shown that $T(n) \in O(n)$.