CS 170

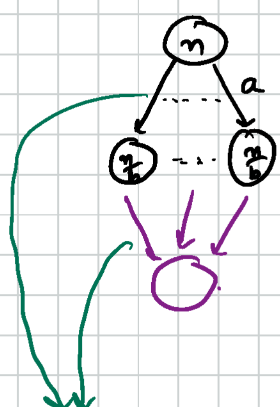
Lecture 4

Sanjam Garg.

- use iClicker

join.iclicker.com/RGWR

Divide & Conquer

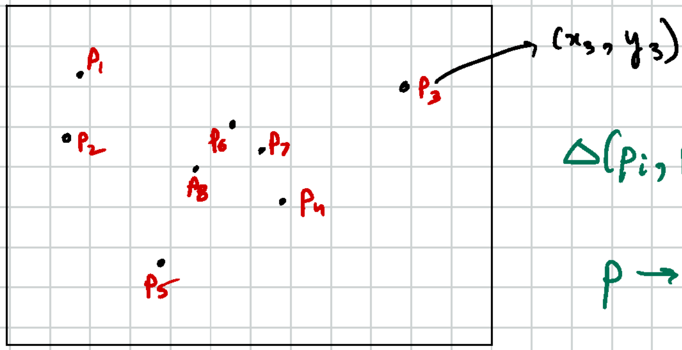


$$T(n) = a\,T\left(\frac{n}{b}\right) + O(n^d)$$

- Binary Search
- Sorting
- Integer multiplication
  - Faster Exponentiation
  - Faster Fibonacci
- Matrix multiplication
- Median finding.

$$T(n) \rightarrow \begin{cases} O(n^d) & \text{if } d > \log_b a \\ O(n^d \log n) & \text{if } d = \log_b a \\ O(n^{\log_b a}) & \text{if } d < \log_b a \end{cases}$$

Closest Pair



$$\Delta(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

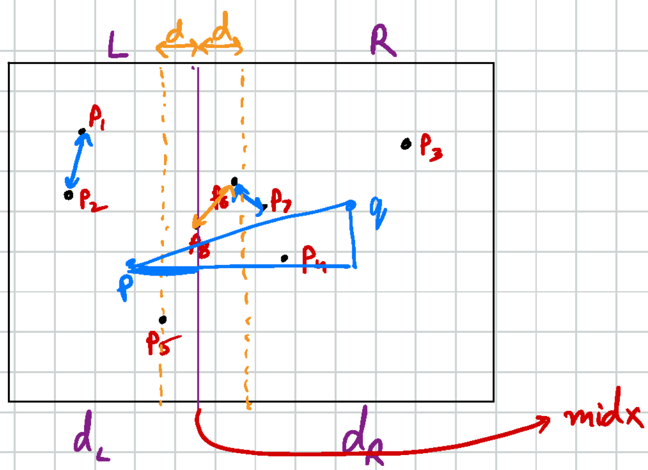
$$p \rightarrow (p.x, p.y)$$

Input: n points in a plane $P = \{p_1 \dots p_n\}$

Output: distance between the closest pair of points

Naive algorithm: $\binom{n}{2} = \frac{n(n-1)}{2} \rightarrow O(n^2)$

Intuition



$d = \min\{d_L, d_R\}$

Property 1: if $p \in L$, $q \in R$ and $\Delta(p,q) < d$

then $|p.x - midx| < d$

& $|q.x - midx| < d$

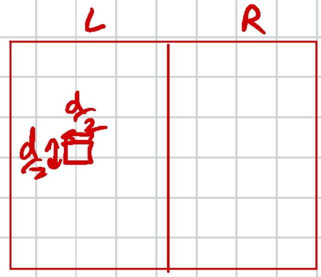
Proof:

p is outside this stip $|p.x - midx| \geq d$

the $\Delta(p.x, q.x) > \Delta(p, (midx, p.y)) \geq d$

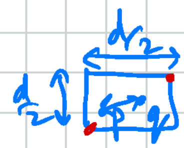
this is a contradiction.

Property 2:



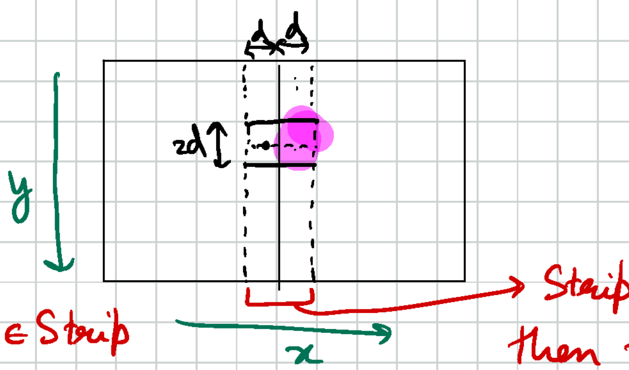
Any tile that is $\frac{d}{2}\times\frac{d}{2}$ that is entirely in L or R can have atmost one point.

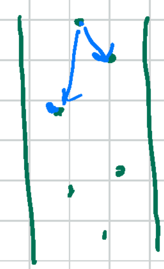
Proof: By contradiction.



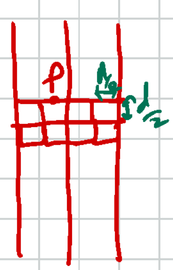
$$\Delta(p,q) \le \sqrt{\left(\frac{d}{2}\right)^2+\left(\frac{d}{2}\right)^2} = \frac{d}{\sqrt{2}} < d$$

$\therefore$ Property 2 holds.





Property 3: p $\in$ Strip then there are atmost 7 points q on the Strip s.t $p.y \le q.y$ and $\Delta(p,q)$ could be less than d



Preprocessing $(P) \rightarrow (P_x, P_y)$

① $P_x = \text{Sort}(P, \text{on the x axis}) \longrightarrow O(n \log n)$

② $P_y = \text{sort}(P, \text{on the y axis}) \longrightarrow O(n \log n)$

Closest (P_x, P_y)

* $n = |P_x|$
  if $n \leq 3$ then brute force.

* Step 1: Divide

  $mid = \lfloor \frac{n}{2} \rfloor$ $\quad midx = P_x[mid].X$

  $Q_x = P_x[1 \dots mid]$ $\quad R_x = P_x[mid+1 \dots n]$

  $Q_y = [\,]$, $R_y = [\,]$

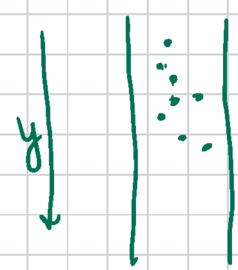
  $O(n)$ ←
  For p in P_y
      if $p.x \leq midx$
          Q_y.append(p)
      else
          R_y.append(p)

* Step 2: Recurse $\quad 2T(\frac{n}{2})$

  $d_L = \text{Closest}(Q_x, Q_y)$ $\quad d_R = \text{closest}(R_x, R_y)$

  $d = \min\{d_L, d_R\}$

* Step 3: Define the strip & conquer.



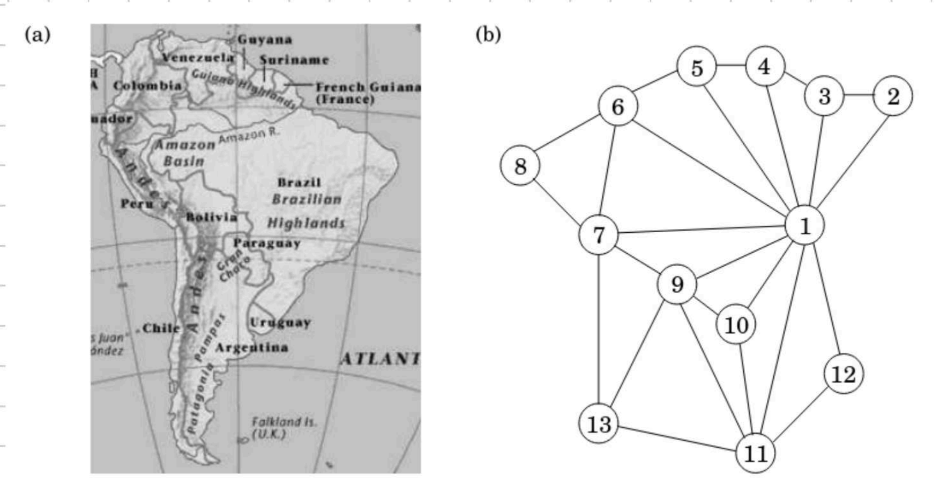
  $O(n)$
  strip = [ ]
  For p in P_y
  if $|p.x - midx| < d$
  strip.append(p)

  $O(k)$
  $k = |strip|$ $\quad d_1 \dots d_k = \infty$
  For $i = 1 \dots k$
      for $j = i+1 \dots \min\{i+7, k\}$
          $d_i = \min\{d_i, \Delta(\text{strip}[i], \text{strip}[j])\}$

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

return $\min\{d, d_1 \dots d_k\}$

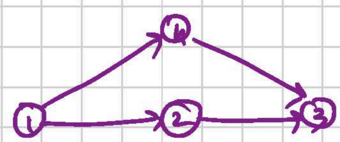
: Graphs :



$G = (V, E)$ $|V| = n$ $|E| = m < n^2$

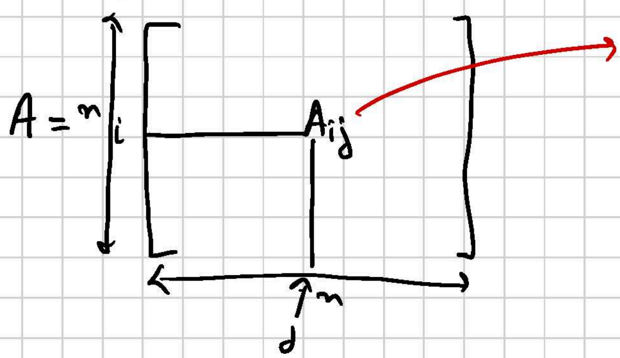
Undirected Graphs →

Directed Graphs →



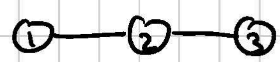
Representing Graphs

(1) Adjacency Matrix Representation



$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise.} \end{cases}$$

(2) Adjacency List Representation



1 → [2]

2 → [1, 3]

3 → [2]

	Matrix	List
Size	$O(n^2)$ space	$O(n+m)$ space
is $(u,v) \in E$	$O(1)$ time	$O(deg(u))$ time
List all neighbours of u	$O(n)$ time	$O(deg(u))$ time.

* Connectivity in Graphs.
  * is there a path from node u to node v?
  * is G connected?
  * What are the connected components?