Nov 14

Closest pair of points

Input: n points: $p_1, \dots, p_n$ $\quad p_i = (x_i, y_i)$

Output: p_i, p_j s.t. $d(p_i, p_j)$ is min
$i \neq j$

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

ASSUMPTIONS:

(1) Given p_i, p_j can compute $d(p_i, p_j)$ in $O(1)$ time
→ WLOG we can ignore $\sqrt{\ }$ (square root)
$d(p_i, p_j)$ is min $\iff$ $d^2(p_i, p_j)$ is min

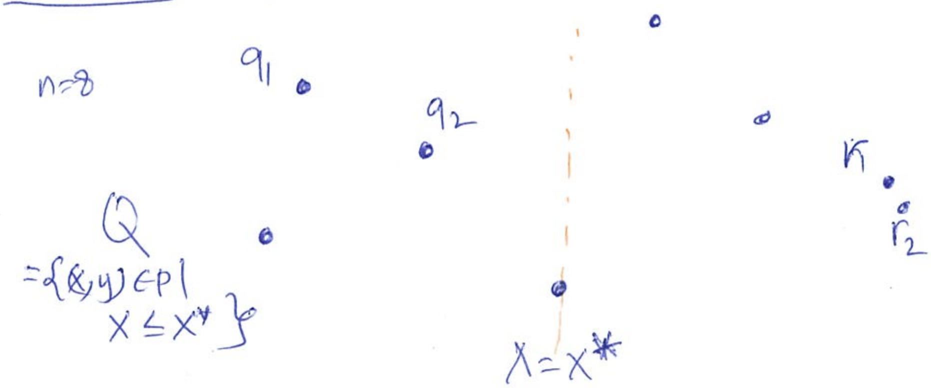
$$d^2(p_i,p_j) = (x_i - x_j)^2 + (y_i - y_j)^2$$

(2) All the x_i's are distinct
——— y_i's ———
If not (i) "Rotate" points slightly
(ii)

Notation: Given a set of points P

P_X: pts in P sorted ~~by~~ in increasing order of x-coord
P_Y: ——— Y ———
$O(n\log n)$

Define: (x^*, y^*) s.t. $P_X\left[\lceil \frac{n}{2} \rceil\right] = (x^*, y^*)$



$R = \{(x,y) \in P \mid x > x^*\}$

By recursion:
(i) (q_1, q_2) be the closest pair of points in Q
(ii) (r_1, r_2) ——— R

Imp Def: $\delta = \min\left(d(q_1,q_2),\ d(r_1,r_2)\right)$

<u>ASIDE!</u> Given P_X, P_Y : compute Q_X, Q_Y, R_X, R_Y in $O(n)$

Q: How?

→ $Q_X = P_X[1 : \lceil \frac{n}{2} \rceil]$, $R_X = P_X[\lceil \frac{n}{2} \rceil + 1 : n]$

→ Scan (x, y) in order of P_Y if $x \le x^*$ → (x, y) to Q_Y

else —— R_Y