Closest pair of points

Oct 30

Input: n points: $P_1, \ldots, P_n$; $P_i = (x_i, y_i)$

Output: P_i, P_j s.t. $d(P_i, P_j)$ is minimized

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

ASSUMPTIONS:

(i) Given P_i, P_j can compute $d(P_i, P_j)$ in $O(1)$ time

→ WLOG can ignore the $\sqrt{\ }$ (square root)

$d(P_i, P_j)$ is min $\iff$ $d(P_i, P_j)^2$ is min

" $(x_i - x_j)^2 + (y_i - y_j)^2$

(ii) All the x_i's are distinct
——— y_i's ———

If not (i) "rotate" all points slightly
(ii) Modify the subsequent algo ~~to handle the general case~~

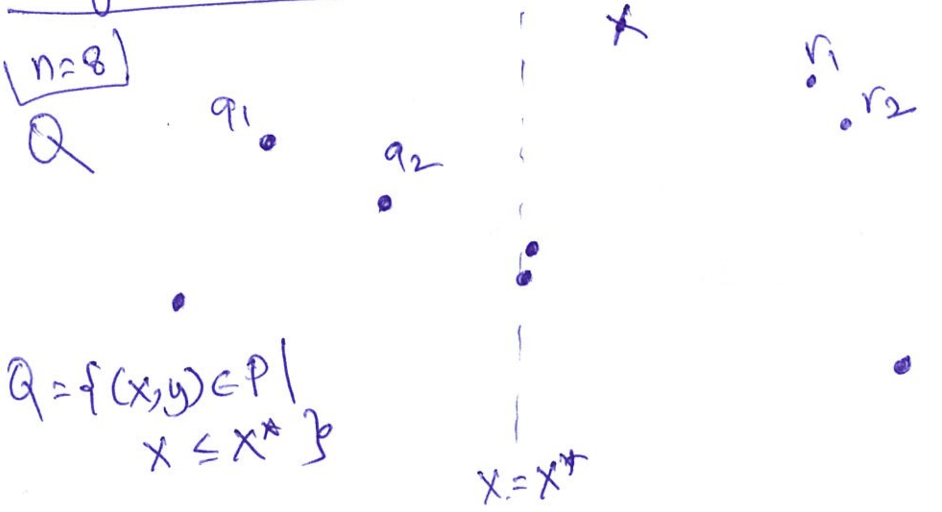
Notation: P be the set of points

P_x : pts in P sorted in increasing order of x values

P_y : ——— y ———

$O(n \log n)$ by sorting

$n = 8$



$Q = \{(x,y) \in P \mid x \le x^*\}$

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

Define $(x^*, y^*) = P_x[\lceil \frac{n}{2} \rceil]$

By recursion find

(i) (q_1, q_2) → closest pair of points in Q

(ii) (r_1, r_2) → ——— R

ASIDE: Given P_x, P_y; compute Q_x, Q_y, R_x, R_y in $O(n)$ time.

Q: How?

$Q_x = P_x[1 : \lceil \frac{n}{2} \rceil]$ $\quad$ $R_x = [\lceil \frac{n}{2} \rceil + 1 : n]$

$\rightarrow$ scan (x, y) in order of P_y if $x \leq x^*$ add (x,y) to Q_y

else add (x,y) to R_y