~~Closet~~ Closest pair of points

Nov 11

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 as we can consider closest pair of points under $d^2(p_i, p_j) = (x_i - x_j)^2 + (y_i - y_j)^2$]

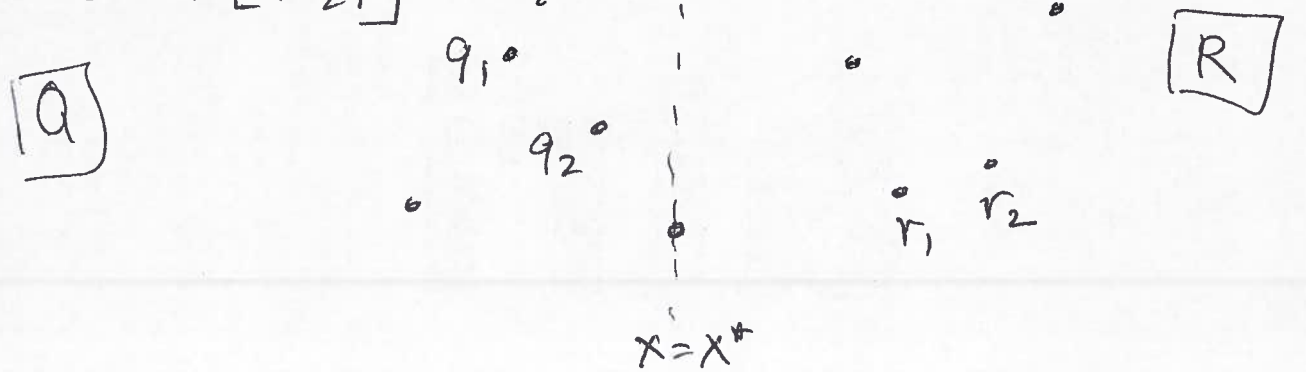
(2) All x_i's are distinct
All y_i's ——— } If not (i) rotate all points slightly

(ii) Subsequent algo can be modified to work without this assumption.

Notation: Given a set of points P

P_x: pts in P in increasing order of x coord
P_y: ——— y ——— } can compute in $O(n \log n)$ time

$(x^*, y^*) = P_x[\lceil \frac{n}{2} \rceil]$ $\quad x_i \leq x^* \leftrightarrow x_i > x^*$



By recursion let

q_1 & q_2 be the closest pair of points in Q

r_1, r_2 ——————— R

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

Obs: The actual smallest distance $\leq \delta$.

Given P_x, P_y

Q: How quickly can you compute Q_x, Q_y, R_x, R_y from P_x, P_y.

→ $Q_x = P_x[1 : \lceil \frac{n}{2} \rceil]$, $R_x = P_x[\lceil \frac{n}{2} \rceil + 1 : n]$ ⟵ $O(n)$

→ Scan each (x, y) in P_y (in order)

if $x \leq x^* \Rightarrow (x, y)$ is in Q_y

o/w ——— R_y.