Nort

Closest pair of points

$x_i, y_i \le \text{poly}(n)$

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

Output: pair p_i, p_j s.t. $d(p_i, p_j)$ is min

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

Assumptions:

1. Given p_i, p_j can compute $d(p_i, p_j)$ in $O(1)$ time.

   $d(p_i, p_j)$ is min $\iff$ $d^2(p_i, p_j)$ is min

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

2. Assume ~~All~~ all x_i values are distinct.

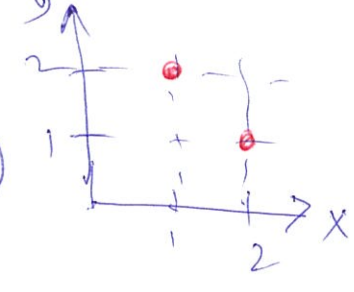
   ——— y_i ———

   If not, (i) (Claim) $\exists$ a rotation of n points that makes x & y co-ords distinct.

   (ii) Can modify the algo we'll see to handle the general case.

Notation: P is the set of points: $P = \{p_1, \dots, p_n\}$

Eg. $P = \{(1,2), (2,1)\}$



P_X is P sorted by x-coord (in incr. order)

Eg. $P_X = \{(1,2), (2,1)\}$

P_Y is P ——— y ———

Eg. $P_Y = \{(2,\underline{1}), (1,\underline{2})\}$

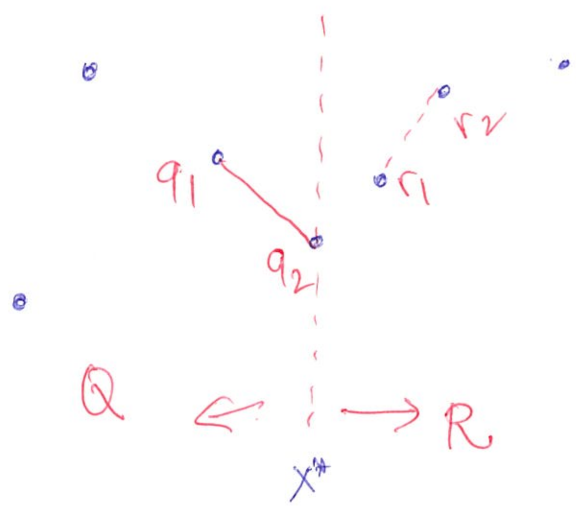
Goal: Do better than $O(n^2)$ time

Towards a divide & conquer algo

(indexing starting from 1)

Step 1: Divide P in equal sized halves Q & R

$n=8$



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

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

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

$$|Q| = \left\lceil \frac{n}{2} \right\rceil$$

$$|R| = n - \left\lceil \frac{n}{2} \right\rceil = \left\lfloor \frac{n}{2} \right\rfloor$$

Step 2: Recursively find closest pairs (q_1, q_2) in Q & (r_1, r_2) in R

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

ASIDE: P_X, P_Y from P in $O(n \log n)$ time.

Q: Given P_X, P_Y; compute Q_X, Q_Y, R_X, R_Y in $O(n)$ time

A: $\underbrace{Q_X = P_X[1 : \lceil \frac{n}{2} \rceil], \; R_X = P_X[\lceil \frac{n}{2} \rceil + 1 : n]}_{O(n)}$

For all $(x,y) \in P_Y$ $\underbrace{\text{if}}_{O(n)}$ $x \le x^*$ add (x,y) to Q_Y

else ———— R_Y