APR 13

closest pair of points

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

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 :

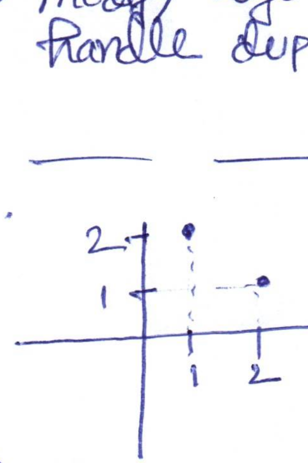
① Given P_i & P_j, we can compute $d(P_i, P_j)$ in $O(1)$ time.
WLOG, ignore the square root.

$$d(P_i, P_j) \text{ is min} \iff d^2(P_i, P_j) \text{ is min}$$
$$= (x_i - x_j)^2 + (y_i - y_j)^2$$

② All x_i values are distinct
All y_i " " "
} if not,
(i) rotate the points
(ii) modify algo to handle duplicates

Notations : P is the set of all points.

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



P_x: Pts in P sorted by x value

P_y: Pts in P " " y "

$P_x = \{(1,2), (2,1)\}$

$P_y = \{(2,1), (1,2)\}$

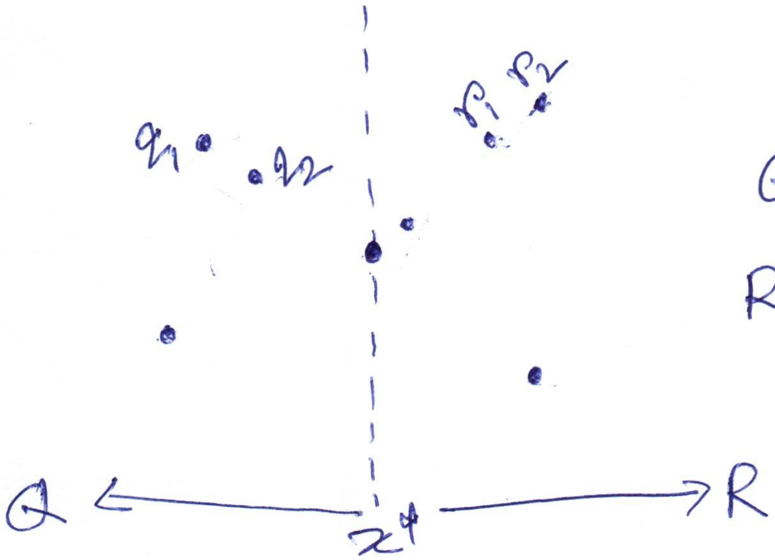
Towards a divide & conquer algo:

$\underline{n=8}$

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

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

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



* let (q_1, q_2) be the closest pair in Q
* let (r_1, r_2) " " " " in R

ASIDE: Given P_x, $\underline{P_y}$, compute Q_x, Q_y, R_x, R_y in $\underline{O(n)}$ time.

* Q_x, R_x in $O(n)$?

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

* Q_y, R_y in $O(n)$?

~~Scan P_y in on~~

$\left.\begin{array}{l} \text{scan } (x,y) \text{ in order of } P_y \\ \text{if } (x \leq x^*), \text{ add } (x,y) \text{ to } Q_y \\ \text{else} \quad " \quad (x,y) \text{ to } R_y \end{array}\right\} O(n)$