April 6

Closest pair of points

Input: n points : $p_1, p_2, \ldots p_n$; $p_i = (x_i, y_i)$

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.

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

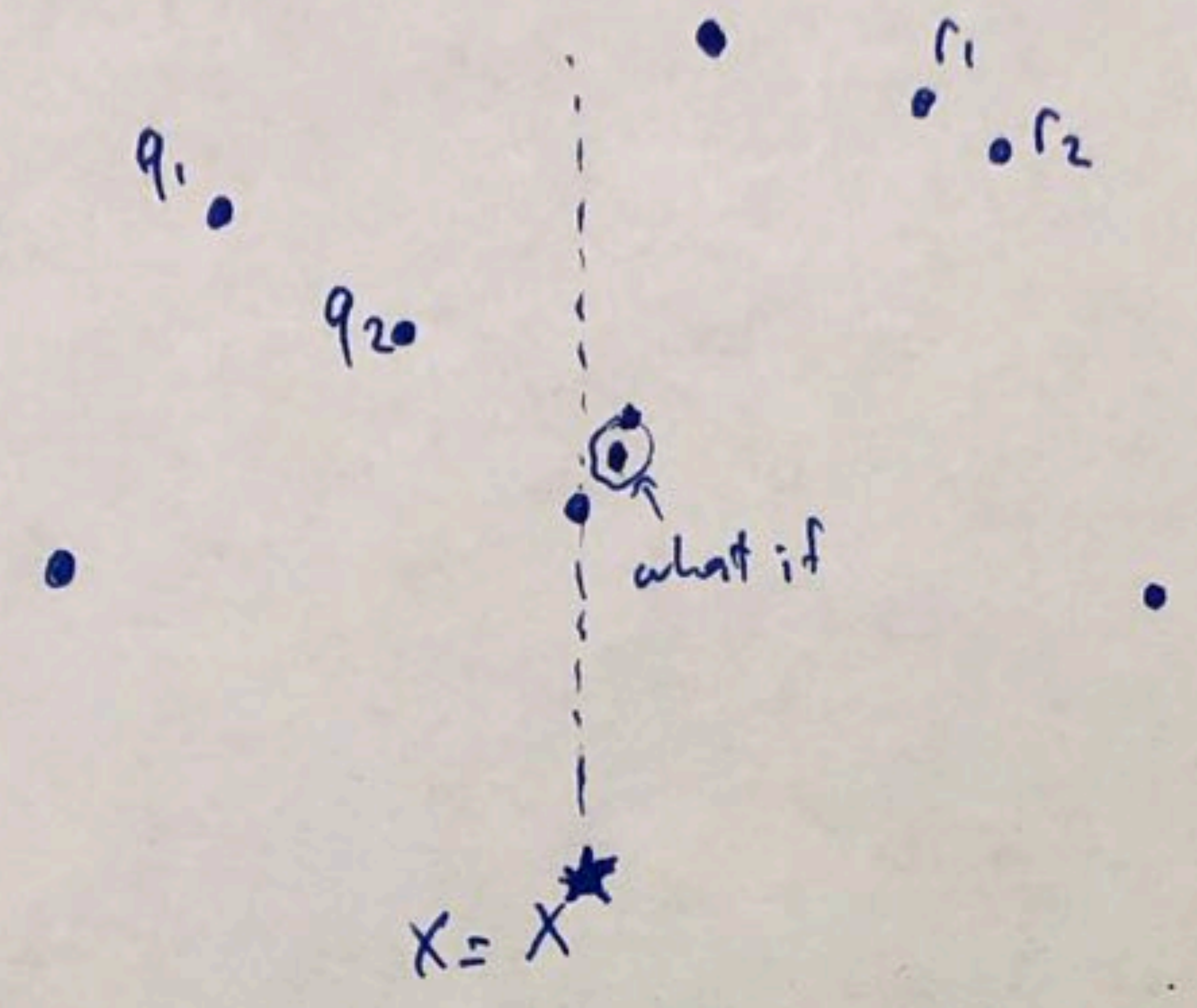
$d(p_i, p_j)$ is min $\Leftrightarrow d(p_i, p_j)^2 \rightarrow (x_i - x_j)^2 + (y_i - y_j)^2$

(ii) All the x_i's are distinct
All the y_i's are distinct
} If not (i) "rotate" all points slightly
(ii) modify the subsequent algo to handle the general case

Notation : P be the set of points

$O(n.\log n)$ by sorting {
P_x : points in P sorted in increasing order of x values
P_y : points in P sorted in increasing order of y values

$n = 8$



Define

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

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

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

By recursion find (i) $(q_1, q_2) \to$ closest pair of points in Q

(ii) $(r_1, r_2) \to$ " " " " " R

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

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

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

else add (x,y) to R_y