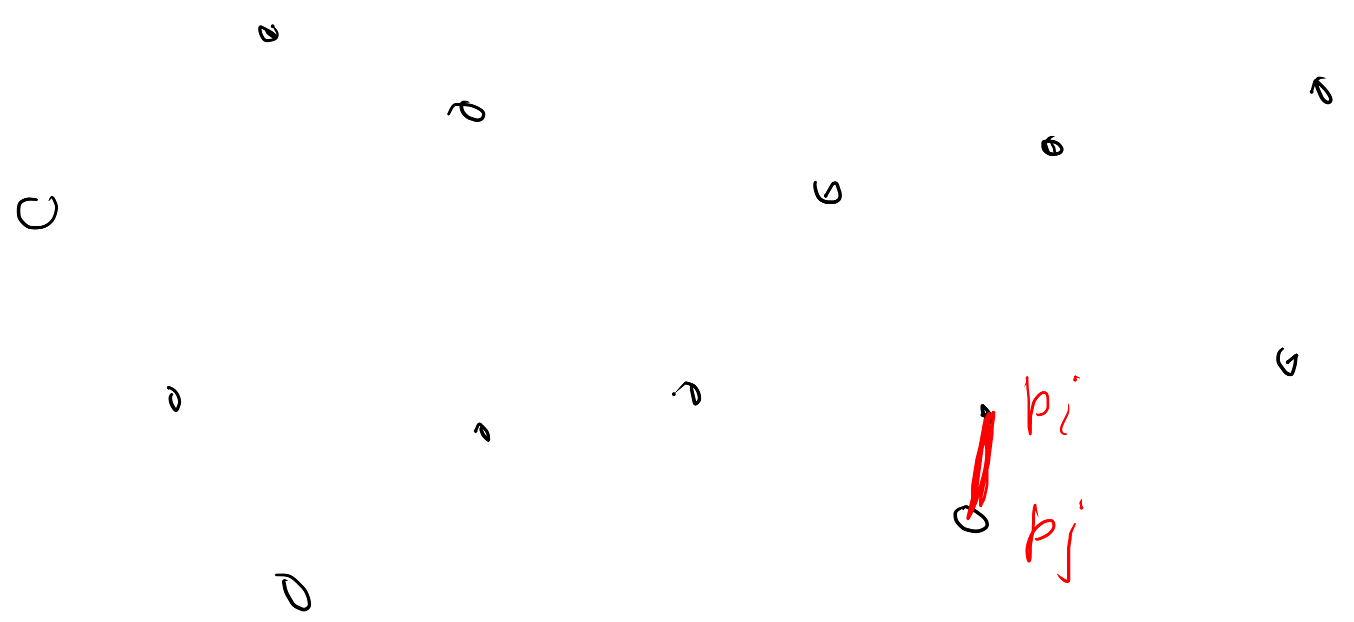
Closest pair problem

Problem: Input: A set of n points $P = \{p_1, p_2, \ldots, p_n\}$

Output: Find a pair (p_i, p_j) such that their distance is minimum.



A naive algorithm

For each pair compute the distance
return the pair with minimum distance.

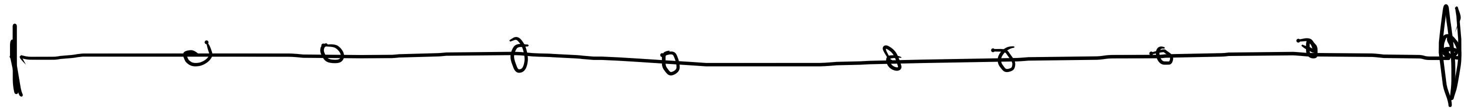
running time $O(n^2)$

can we do better??

1D-version

points are on a line.

Sort the points.

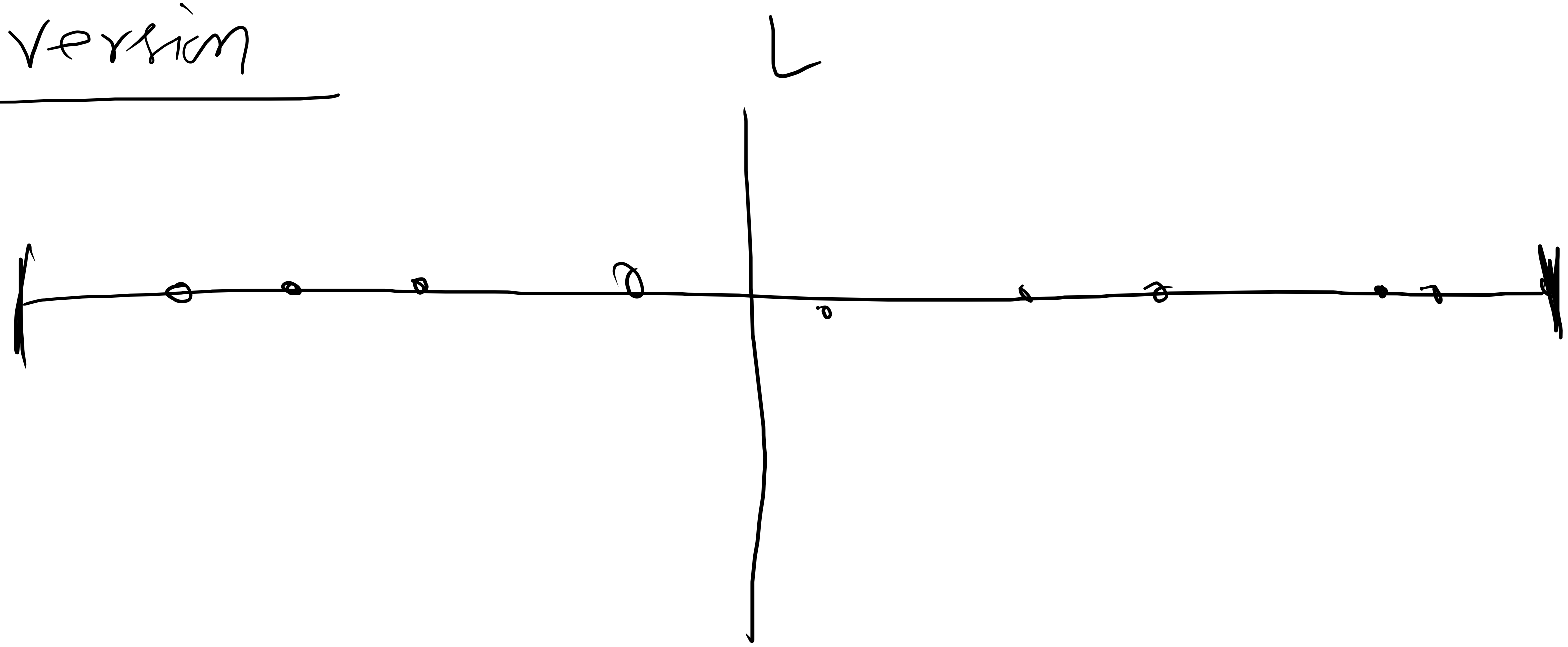


take distance between consecutive pair.

running time: $O(n \lg n) + O(n)$

Applying D&C on 1D version

1 Sort the points



closestpair (P)

If $|P| = 1$ return $\delta_F = \infty$

If $|P| = 2$ return $\delta_F = |p_2 - p_1|$

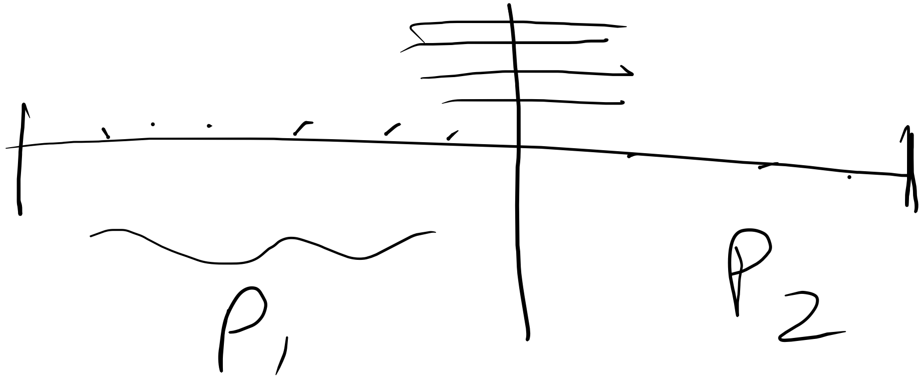
otherwise

L = median (P)

Divide P in P_1 and P_2 w.r.t. L

δ_L = closestpair (P_1)

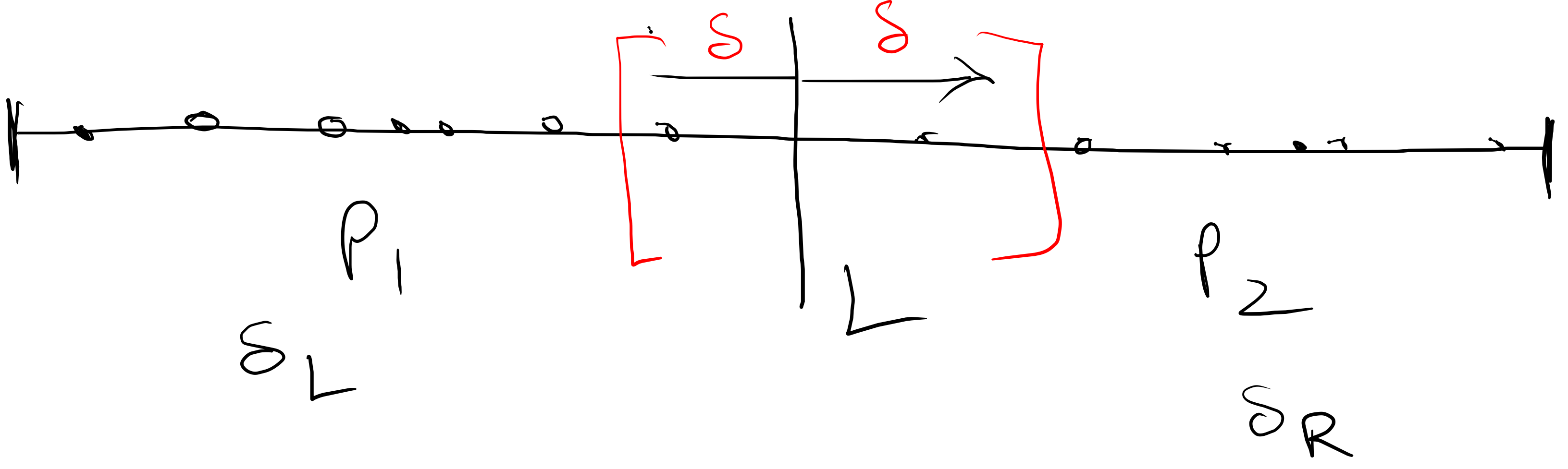
δ_R = closestpair (P_2)



δ_{12} = minimum distance crossing L —— f(n)

return $\delta_F = \min\{\delta_L, \delta_R, \delta_{12}\}$

$$T(n) = 2T(n/2) + f(n)$$ where f(n)



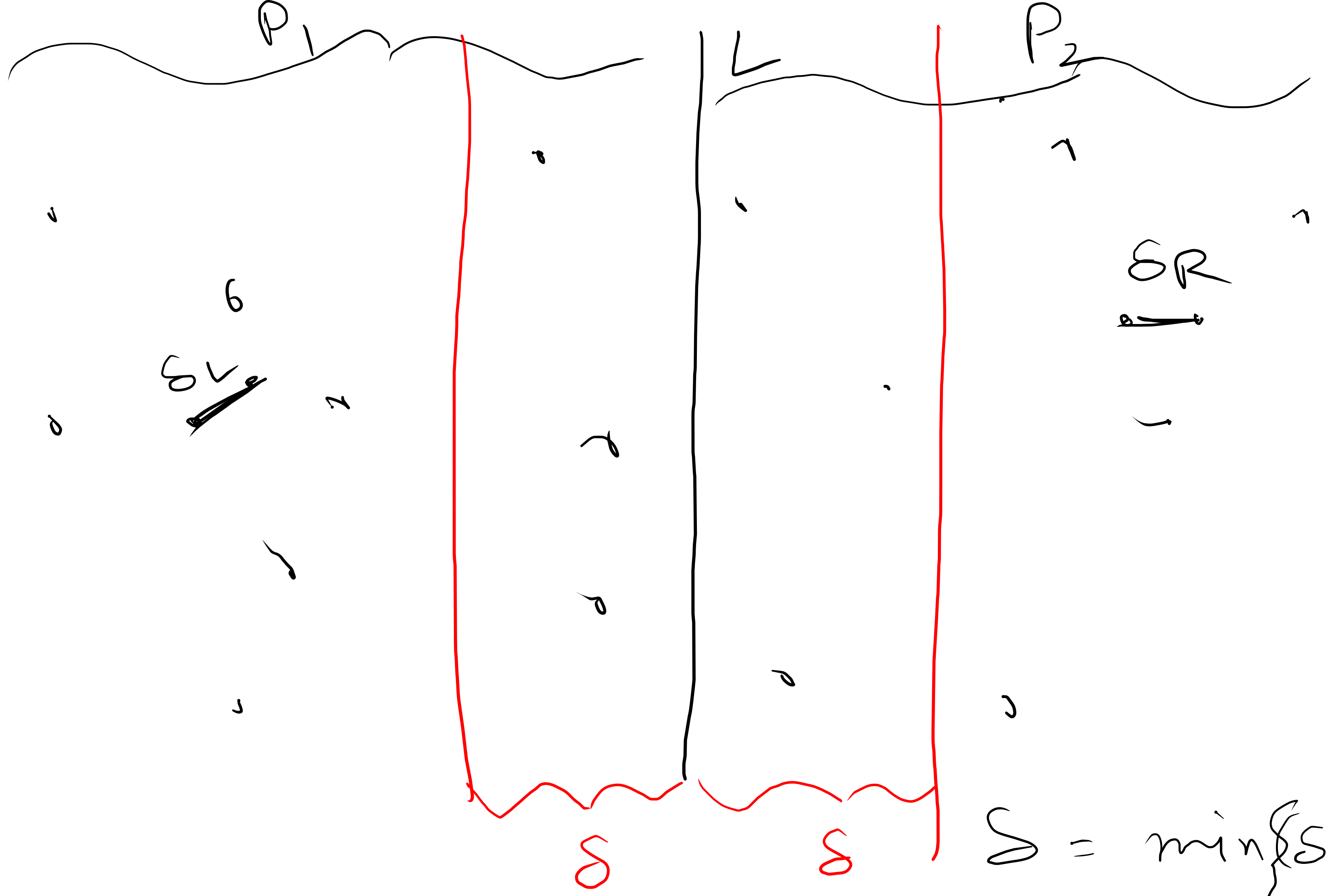
$\delta = \min\{\delta_L, \delta_R\}$

$$T(n) = 2T(n/2) + \theta(n)$$

$$= O(n \lg n)$$

2D-version

closestpair2D(P)



$\delta = \min\{\delta_L, \delta_R\}$

$\delta_L =$

$\delta_R =$

for each p in P_1 and for each q in P_2

compute their distances

$\delta_{12} =$ minimum of them

$\delta_F = \{\delta_L, \delta_R, \delta_{12}\}$

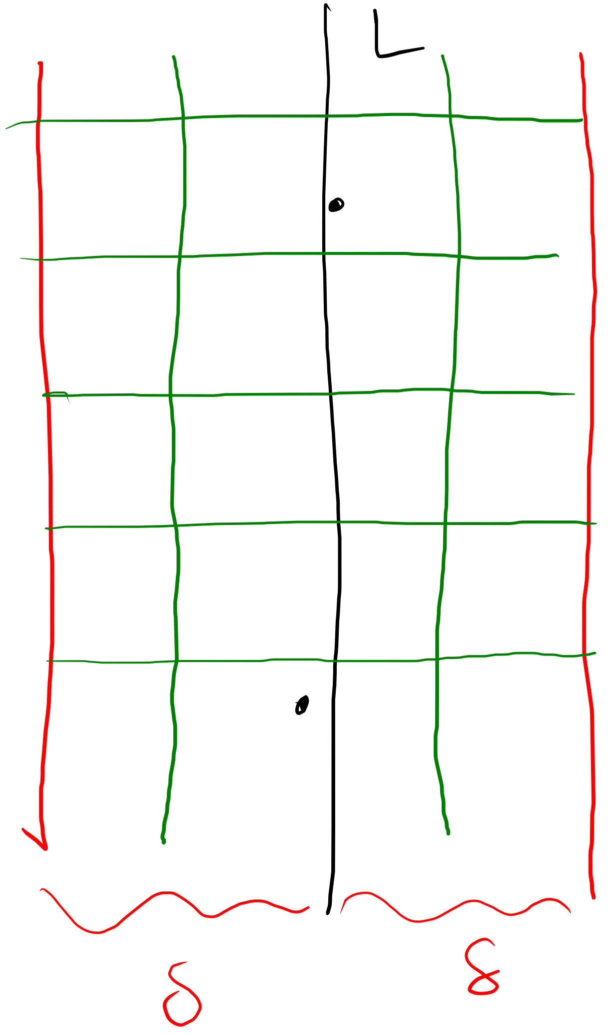
$T(n) = 2T(n/2) + O(n^2)$

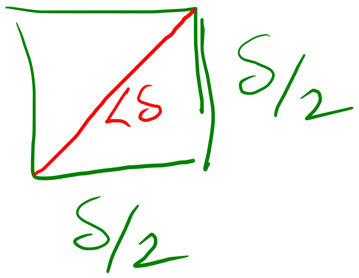
$= O(n^2)$

no improvement

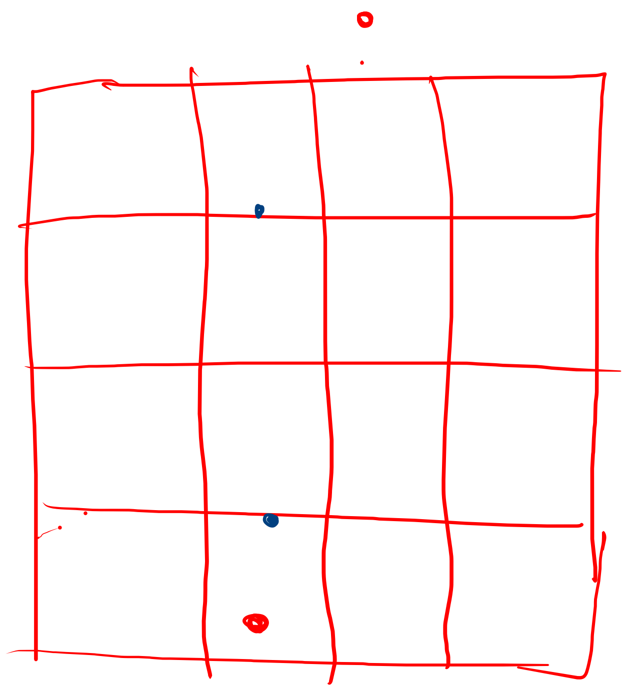
Question: How many points are there in a single box?

Ans: At most 1





$<\delta$



4x4

The final algorithm

P_x : Sorted in x order
P_y : " " y order

closestpair2D (P)
construct P_x and P_y — $O(n\log n)$
(p_0^*, p_1^*) = closest-pair-rec (P_x, P_y) — $O(n\log n)$

Q : first $n/2$ points in P_x
$R \leftarrow$ remaining points in P_x

closest-pair-rec (P_x, P_y) — $T(n)$
base condition. $\Big\}\ \theta(1)$

Q_x : Q sorted in x direction
Q_y : Q " in y "
R_x : R " " x direction
R_y : R " " y "

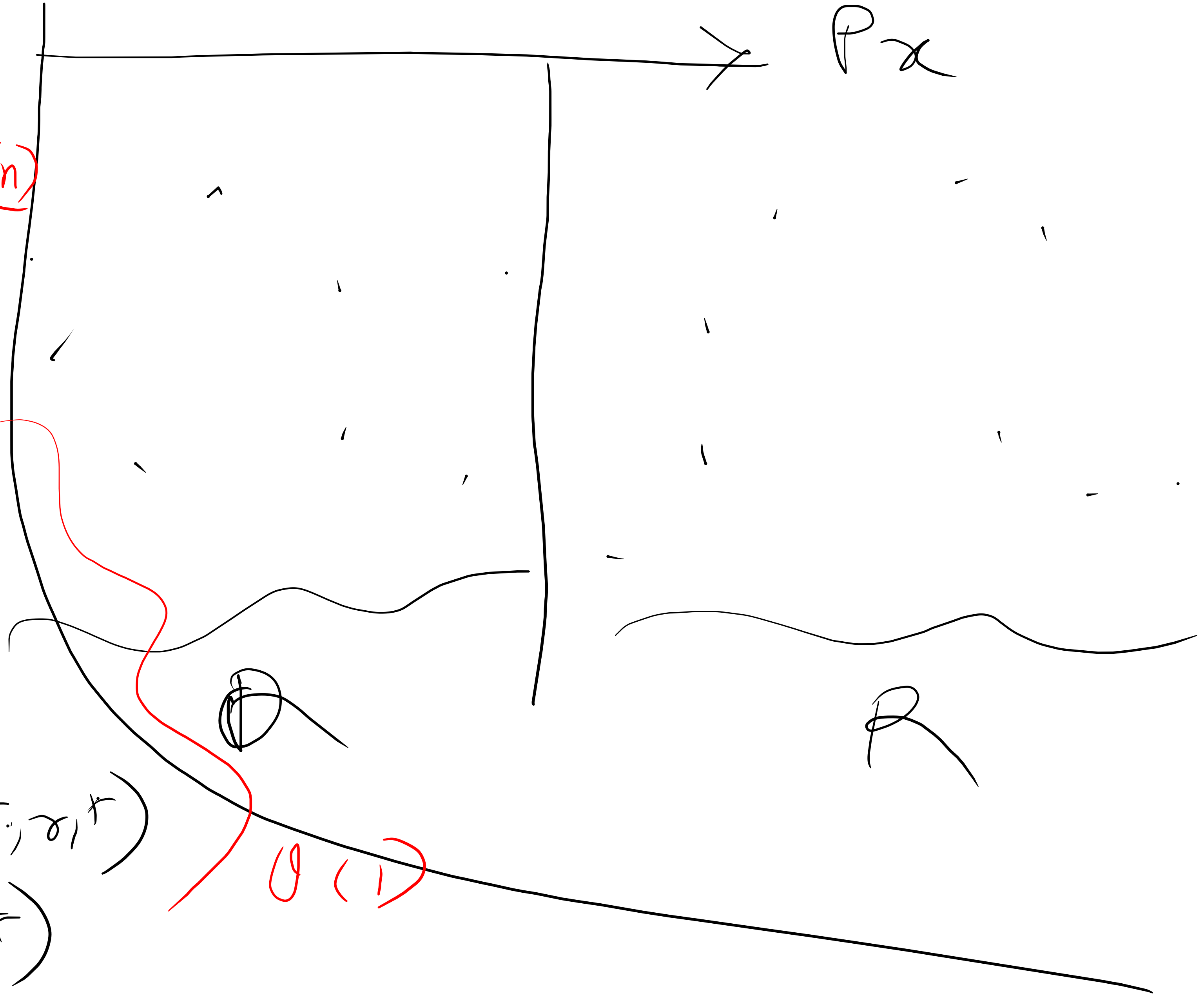
construct Q_x, Q_y, R_x, R_y — $O(n)$
(q_0^*, q_1^*) — closestpairrec (Q_x, Q_y) — $T(n/2)$
(r_0^*, r_1^*) — " " " (R_x, R_y) — $T(n/2)$
$\delta = \min\{d(q_0^*, q_1^*), d(r_0^*, r_1^*)\}$ — $\theta(1)$
$x^* = \max$ x-coordinate of a point in Q — $\theta(1)$
$L = \{(x,y) \mid x = x^*\}$
S = points in P within δ distance of L — $O(n)$

construct S_y — $O(n)$

for each point $s \in S_y$
compute distances from
s to each of the next
15 points in S_y — $O(n)$

let δ_{12} be minimum
$\delta_{12} = d(q_2^*, r_2^*)$

If $d(q_2^*, r_2^*) < \delta$ then
return (q_2^*, r_2^*)
else if $d(q_0^*, q_1^*) < d(r_0^*, r_1^*)$
return (q_0^*, q_1^*)
else return (r_0^*, r_1^*)



$T(n) = 2T(n/2) + O(n)$
$= O(n \lg n)$