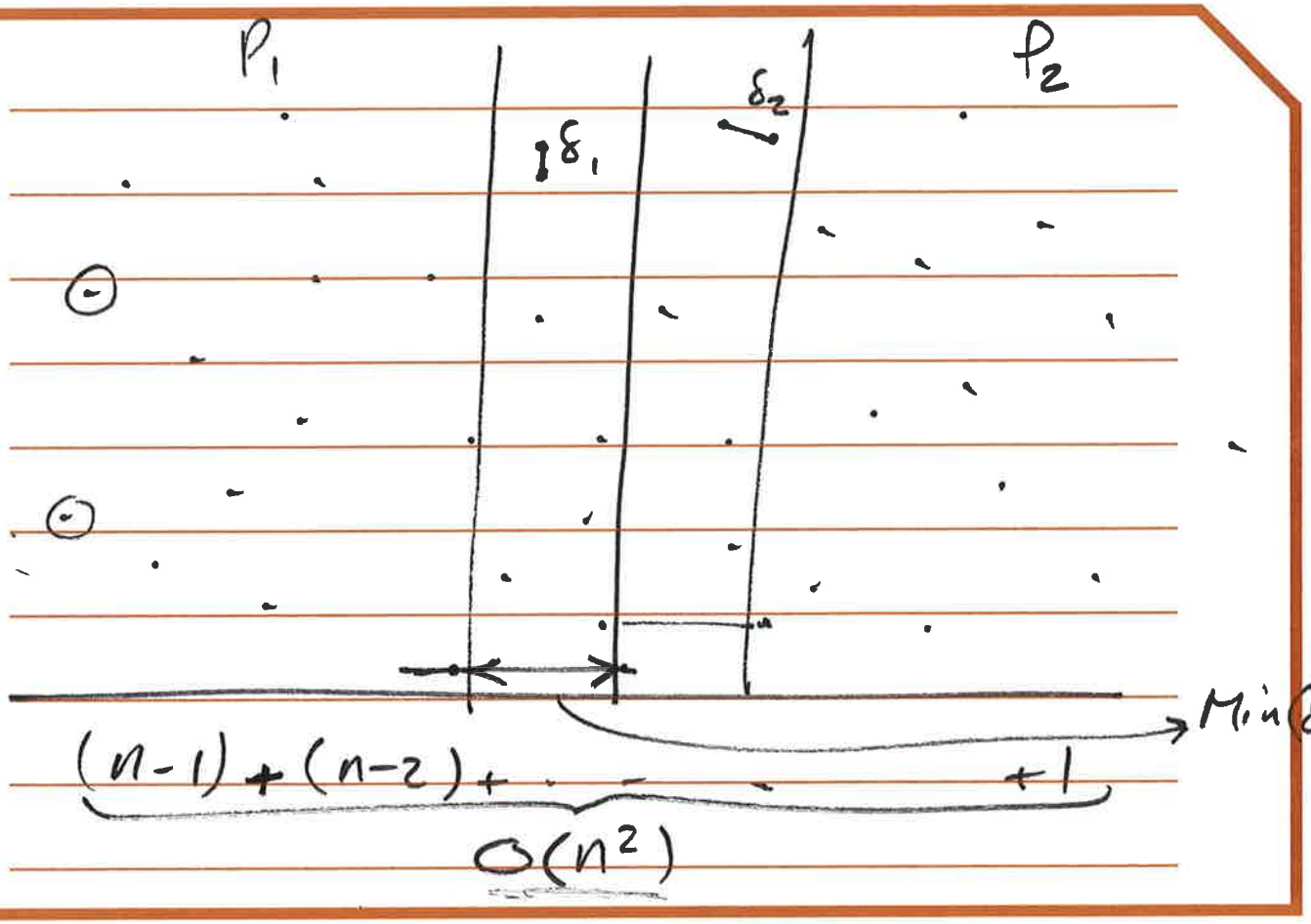
Closest Pair of Points in 2D



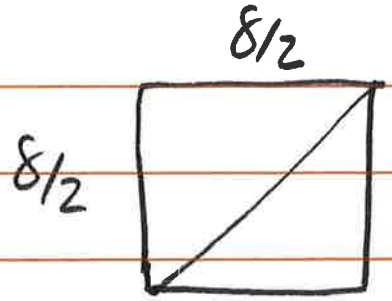
to combine, we need to look at following cases:

1- both points are in P_1

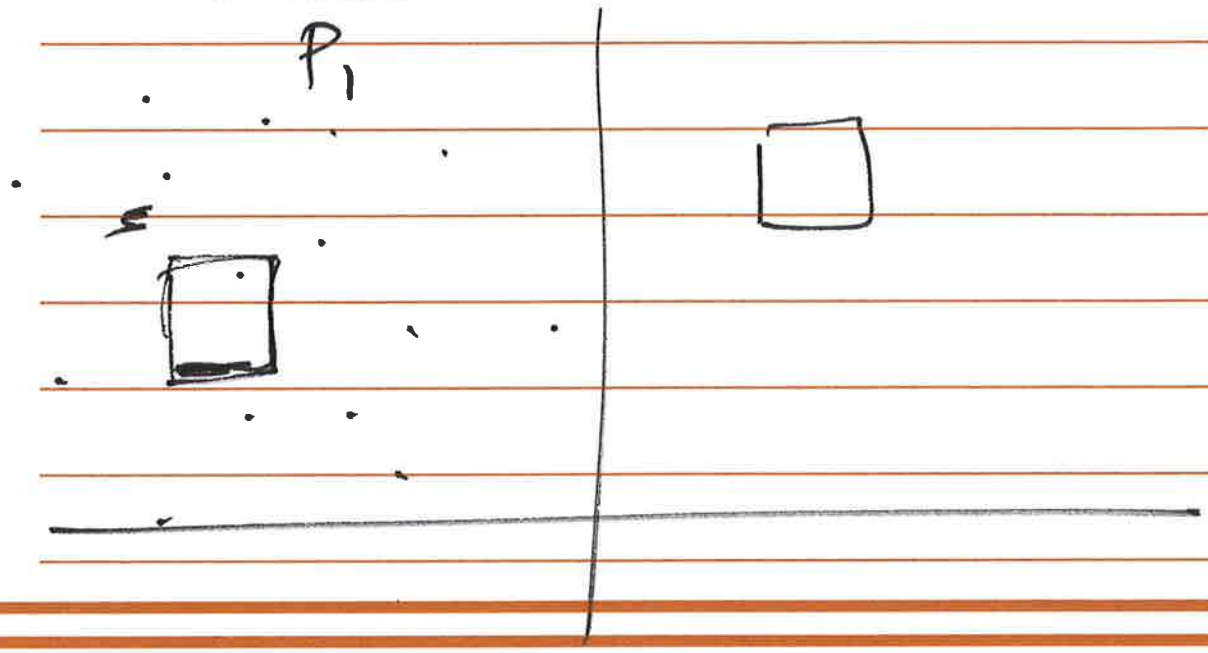
2- " " " " P_2

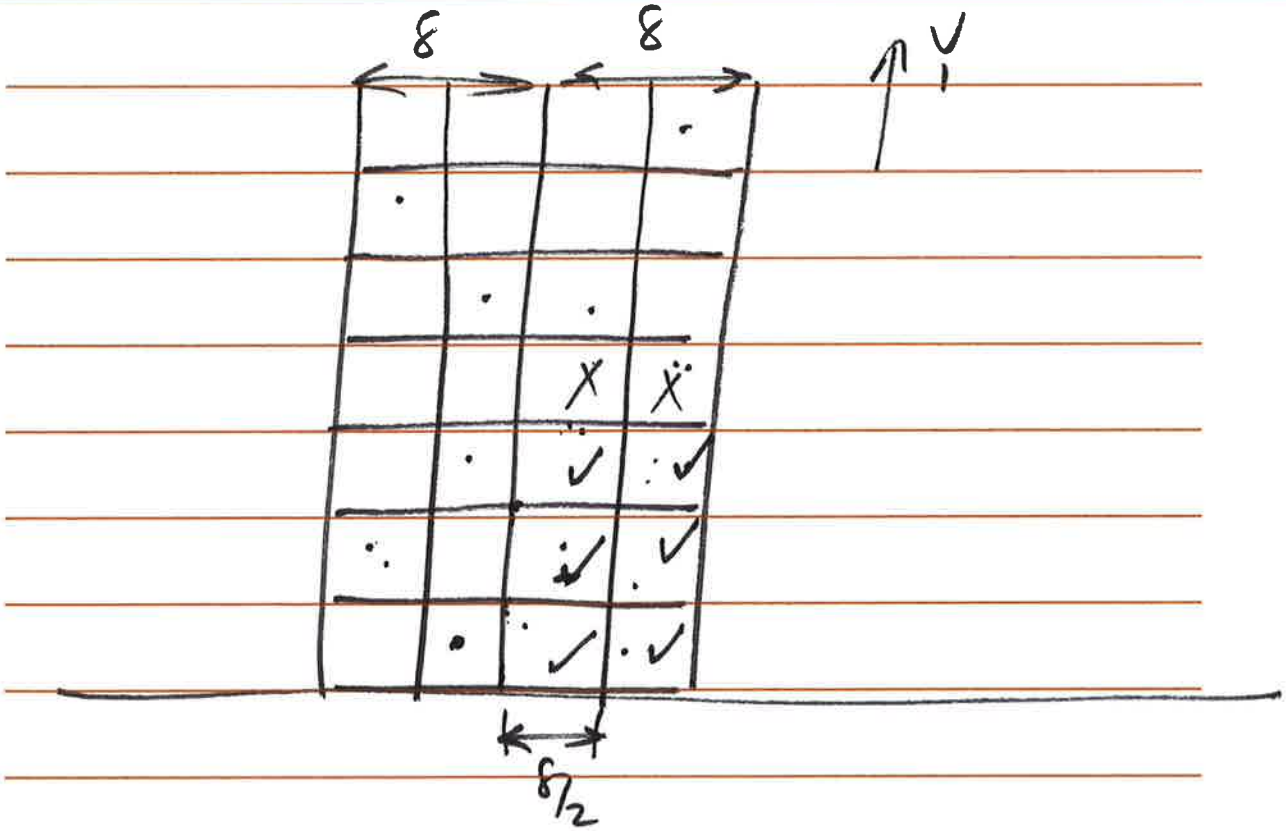
→ 3- one pt. in P_1 & one pt. in P_2

$\delta = \text{Min}(\delta_1, \delta_2)$



$\frac{\delta\sqrt{2}}{2} < \delta$





Implementation

closest_pair (P)

O(n log n) {
- Construct P_x : list of pts sorted by x-coord.
- " P_y : list of pts sorted by y-coord.

(p_0, p_1) = closest_pair_Rec (P_x, P_y)

closest_pair_Rec (P_x, P_y)

if $|P| \leq 3$ then
trivial case, solve it directly

else

O(1) → Construct Q_x ... left half of P_x

O(n) → " Q_y ... list of pts in Q_x sorted by y-coord.

O(1) Construct R_x ... right half of P_x

O(n) → " R_y ... list of pts in R_x sorted by y-coord.

(q_0, q_1) = closest-pair-rec (Q_x, Q_y)

(r_0, r_1) = " " (R_x, R_y)

$O(1)$ → $\delta = \min(d(q_0, q_1), d(r_0, r_1))$

$O(n)$ → $S_=$ = points in P within distance δ of L.

$O(n)$ Construct S_y .. set of pts in S sorted by y-coord.

$O(n)$ For each pt. $s \in S_y$, compute distance from s to each of next 11 pts. in S_y. Let (s, s') be pair w/ min distance

$O(1)$ if $d(s, s') < \delta$ then
Return (s, s')
else $d(q_0, q_1) < d(r_0, r_1)$ then
Return (q_0, q_1)
else Return (r_0, r_1)
end if

$a = 2$

$b = 2$

$n^{\log_b a} = n^{\log_2 2} = n^1 = n$

$f(n) = O(n)$ ←

Case #2 $\Rightarrow \Theta(n \lg n)$

Overall complexity:

$\underbrace{O(n \lg n)}_{\text{Driver}} + \underbrace{O(n \lg n)}_{\text{recursive sol.}} = O(n \lg n)$