Ex #4: Closest Pair of Points (hard) ①

Input: n points $P = \{p_1, p_2, \dots, p_n\}$

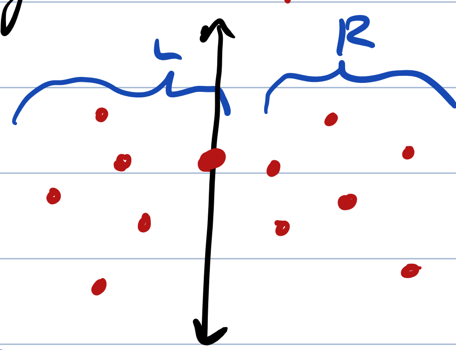
Goal: Find the pair (p_i, p_j) such that $d(p_i, p_j)$ = Euclidean Distance is minimized.

(Assume distinct x and y values for simplicity.)

Step 1: - Create a version of P that is sorted by x-value, call it P_x.
- Create a version of P that is sorted by y-value, call it P_y. $O(n \log(n))$

Step 2: Begin divide-and-conquer.

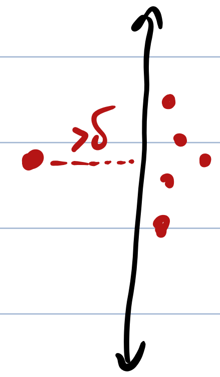
- Split P into left half L and right half R using P_x. $O(1)$



- Form L_x, L_y, R_x, R_y using P_x and P_y. $O(n)$
- Find closest pair in L: (ℓ_1, ℓ_2) and closest pair in R: (r_1, r_2) } recursion.
- Set $\delta = \min(d(\ell_1, \ell_2), d(r_1, r_2))$. $O(1)$
- Now the hard part: how do we combine?

Closest pair could be in L, in R, or have one point in each.

Fact 1: If the closest pair is split across the middle line, then each point has to be within δ of the line.

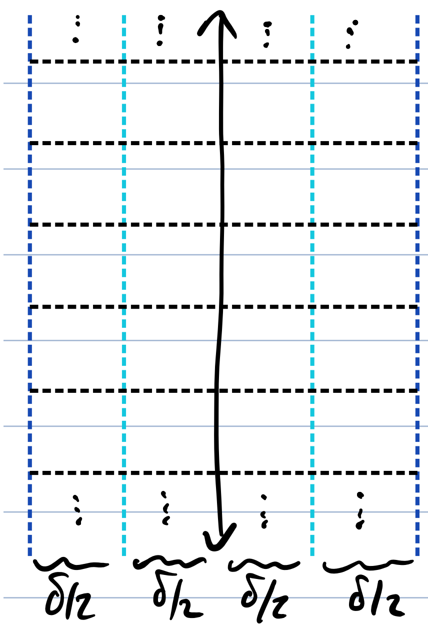


Define S to be just the points within δ of the line. $O(n)$

Note that $S = P$ is possible!

Form S_x and S_y using P_x and P_y. $O(n)$

Here's where it gets really weird! Split up the 2δ-wide vertical strip centered on the middle line into $\delta/2 \times \delta/2$ boxes.



Fact 2: Each box contains at most a single point of S. (Otherwise, those points would be $< \frac{\delta}{2}\sqrt{2} < \delta$ apart, contradicting the fact that δ is min. distance on either side of the line.)

Let's think about S_y, the points in S ordered by y-value.

If you have two points in S_y that are 4 positions apart (e.g., the 10^{th} and 14^{th}), they have to be on different rows of squares.

8 apart $\rightsquigarrow$ empty row between them $\rightsquigarrow > \delta/2$ apart
12 apart $\rightsquigarrow$ 2 empty rows between them $\rightsquigarrow > \delta$ apart

<u>Fact 3</u>: If two points in S are $\leq \delta$ apart, then positions in S_y differ by <u>at most 11</u>.

So, to find the closest pair in S, we don't have to check every pair ($O(|S|^2)$), only the pairs at most 11 apart

$$\left.\begin{array}{ll} s_1 & s_2 \\ s_1 & s_3 \\ & \vdots \\ s_1 & s_{12} \end{array}\right\} 11$$

$$\left.\begin{array}{ll} s_2 & s_3 \\ & \vdots \\ s_2 & s_{13} \end{array}\right\} 11 \qquad = O(11 \cdot |S|) = O(|S|) < O(n)$$

$$\vdots$$

Summary:

- Presort to get P_x, P_y $\quad O(n \log(n))$
- Split in half and form L_x, L_y, R_x, R_y $\quad O(n)$
- Recursively solve on L and R

4

- Find S, S_x, S_y $O(n)$
- Check pairs in S at most 11 apart $O(n)$

$T(n) = O(n \cdot \log(n)) + S(n)$

$S(n) = O(n) + 2 \cdot S(n/2) + O(n) + O(n)$

$\Rightarrow S(n) = O(n \cdot \log(n))$

$\Rightarrow T(n) = O(n \cdot \log(n))$.