Closest Pair

- Given a collection of n points, find the pair of points that are closest together.
  - 2D point $p = (x, y)$

$O(1)$ - Euclidean distance $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

Brute Force

- for each pair of points find the distance between them

- \# of pairs of points

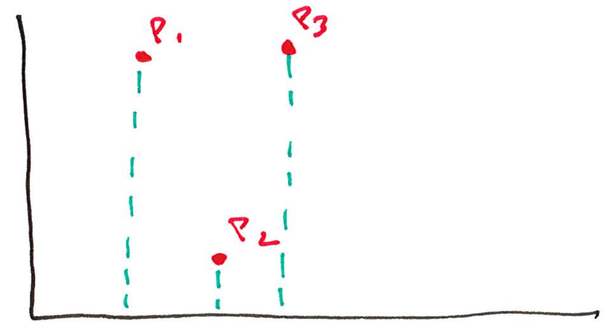
$$\frac{n(n-1)}{2} \in \Theta(n^2)$$

n	n^2	$n \log n$
10	100	10
100	10,000	200
1000	1,000,000	3000

1D Closest Pair

- sort points $\Theta(n \log n)$
- linear pass comparing each point with its neighbors $\Theta(n)$

Scaling to 2D



- can we reduce 2D closest pair to sorting by projecting onto a number line?

  No

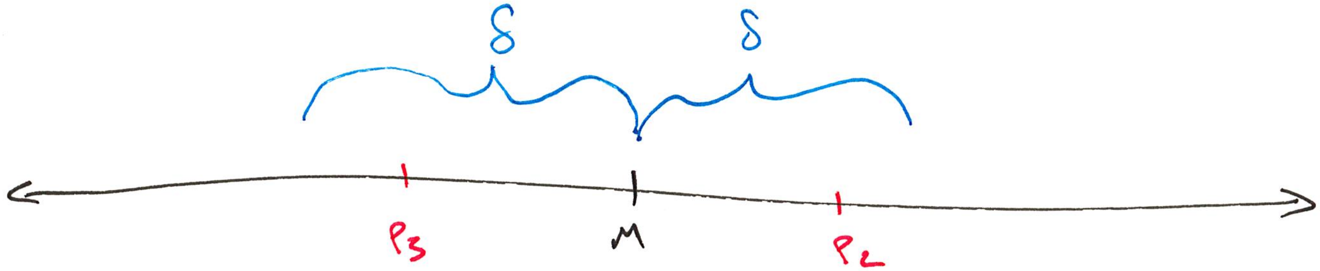
1D closest pair without sorting

- Divide and Conquer

1. Find the median point M and partition the points into two groups L and R
2. Recursively find the closest pair in each half.
   - let δ_L and δ_R be the distance between the closest pair of points in L and R
   - let $\delta = \min(\delta_L, \delta_R)$

The closest pairs of points might contain a point from L and a point from R

- we only need to compare points within δ of M



- $L' = L \cap (m - \delta, m]$ and $R' = R \cap [m, m + \delta)$

3. Extract L' and R' in linear time. Compare each point in L' to each point in R'. If the closest pair among these has distance $< \delta$ return it. Otherwise return the ~~cot~~ closest pair from L or R.

Master Theorem

- works for any recurrence relation of the following form

$$T(n) = aT\left(\frac{n}{b}\right) + f(n) \quad \text{where } f(n) \in \Theta(n^d)$$

$$a > 0$$
$$b > 1$$

- compare $a \; ? \; b^d$

$$T(n) = \begin{cases} \text{if } a < b^d & \Theta(n^d) \\ \text{if } a = b^d & \Theta(n^d \log n) \\ \text{if } a > b^d & \Theta(n^{\log_b a}) \end{cases}$$

- 1D closest Pair

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$$

$a = 2$
$b = 2$
$d = 1$

$2 \; ? \; 2^1$

$\Theta(n \log n)$

Binary Search

$$T(n) = T\left(\frac{n}{2}\right) + O(1)$$

$a = 1$
$b = 2$
$d = 0$

$1 \ ? \ 2^0$

$$\Theta(n^0 \log n)$$

$$\Theta(\log n)$$

Square Matrix Multiplication (Divide and Conquer)

$$T(n) = 8T\left(\frac{n}{2}\right) + \Theta(n)$$

$a = 8$
$b = 2$
$d = 1$

$8 \ ? \ 2^1$

$$\Theta(n^{\log_2 8})$$

$$\Theta(n^3)$$

2D Closest Pair

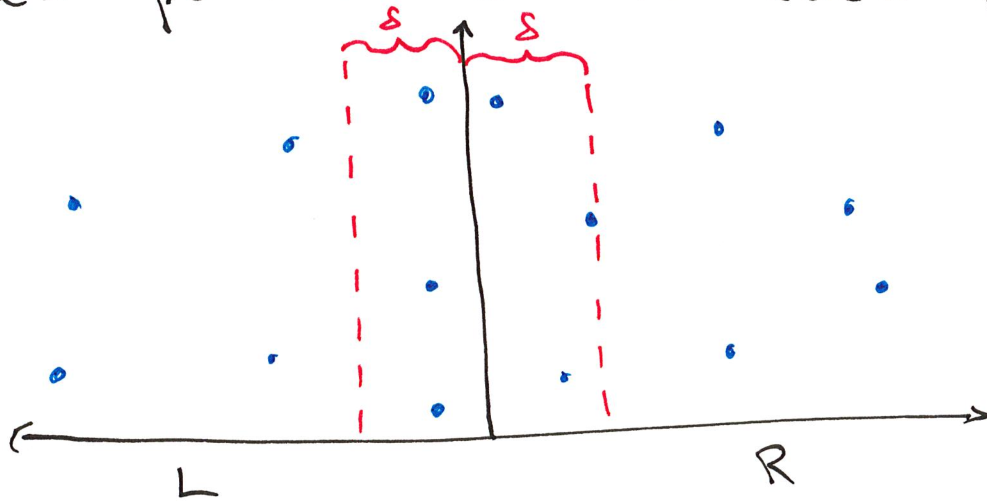
- generalize the 1D solution

1. Take the median m of the x-coordinates. Let ℓ be the line $x = m$. Divide the points into L and R according to whether they are on the left or right of ℓ.

2. Recursively calculate the closest pairs in L and R. Let $\delta = \min(\delta_L, \delta_R)$

3. If there is a closer pair of points $p_L = (x_L, y_L)$ and $p_R = (x_R, y_R)$ these points must be within δ of ℓ.

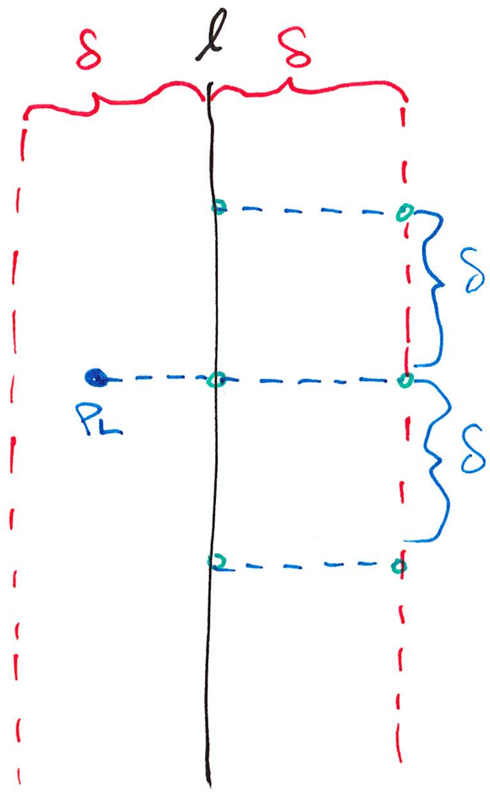
- Define L' to be the points from L within δ of ℓ
- Define R' to be the points from R within δ of ℓ
- compare each point in L' to each point in R'



- In the worst case all points might be within δ of ℓ
- to avoid paying $\Theta(n^2)$-time
  - sort L' and R' by y-coordinate
- process the points from L' and R' in increasing y order

- for point $p_L \in L'$ where $p_L = (x_L, y_L)$ only consider points $p_R \in R'$ such that

$$y_R \in (y_L - \delta, y_L + \delta)$$



how many points from R' can be in this $\delta \times 2\delta$ rectangle? 6

Complexity

$$T(n) = \overset{\text{divide}}{\Theta(n)} + \overset{\text{conquer}}{2T\left(\frac{n}{2}\right)} + \overset{\text{sorting}}{\Theta(n \log n)} + \overset{\text{combine}}{\Theta(n)}$$

$$= 2T\left(\frac{n}{2}\right) + \Theta(n \log n) \in \Theta(n \log^2 n)$$

- Presorting by y-coordinate gets us

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$$

$$\in \Theta(n \log n)$$