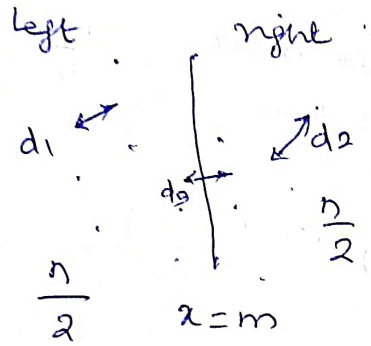
Closest pair problem

Input : set of 'n' points in 2D $\{(x_1, y_1)\ (x_2 y_2) \cdots x_n y_n\}$

Output : Find the closest pair.

Brute force complexity is $O(n^2)$

But with Divide and conquer, a running time of $O(n \log n)$ can be acheived



Approach

- compute median and split the plane of points into 2 regions left and right. $\boxed{O(n)}$

3 possibility exists.

case 1 : 2 pts in left $\boxed{T(n/2)}$

Case 2 : 2 pts in right $\boxed{T(n/2)}$

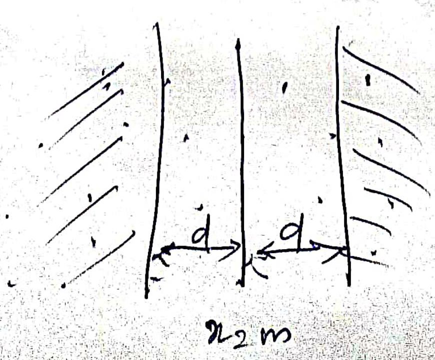
Case 3 : 1 pt in left and 1 pt on right.

1) Find the closest pair in left region (d_1)

2) Find the closest pair in right region (d_2)

3) Find the closest pair between regions or check if there exists a pair such that one pair is on the left region and other point on the right region and the distance b/w them is smaller than $d = \min(d_1, d_2)$

case 3 :



Algorithm

1) Collect all the points in the selected region and sort in ascending order of y-co-ordinates.

Lets it be

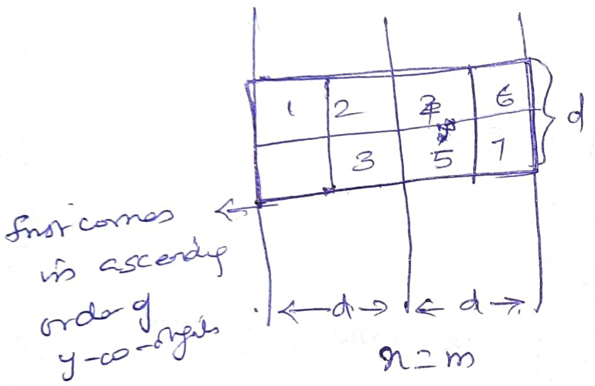
$P_1, P_2, P_3 \cdots P_{50}, P_{51} \cdots$

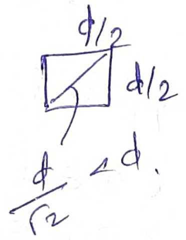
2) For each pt in this list compute the distance between itself and the immediate next 7 points and find the pair with minimum distance d_m and if

$d_m < d$ then d_m is the minimum distance
else d is the minimum distance and its corresponding pts will be closest pair.

No: of computation = 7 x No: of pts.

<u>correctness</u> → of algorithm.





This means that there won't be 2 pts in same little box.

Recursive complexity. sorting along y-coordin.

$$T(n) = \underbrace{O(n)}_{\text{calculate median}} + \underbrace{2T(n/2)}_{\text{left \& Rt}} + O(n\log n)$$

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

$$= O(n\log^2 n).$$

T(n) = time for solving CPP and ~~solving~~ sorting points w.r.t to y coordinate.

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

$$= O(n\log n).$$