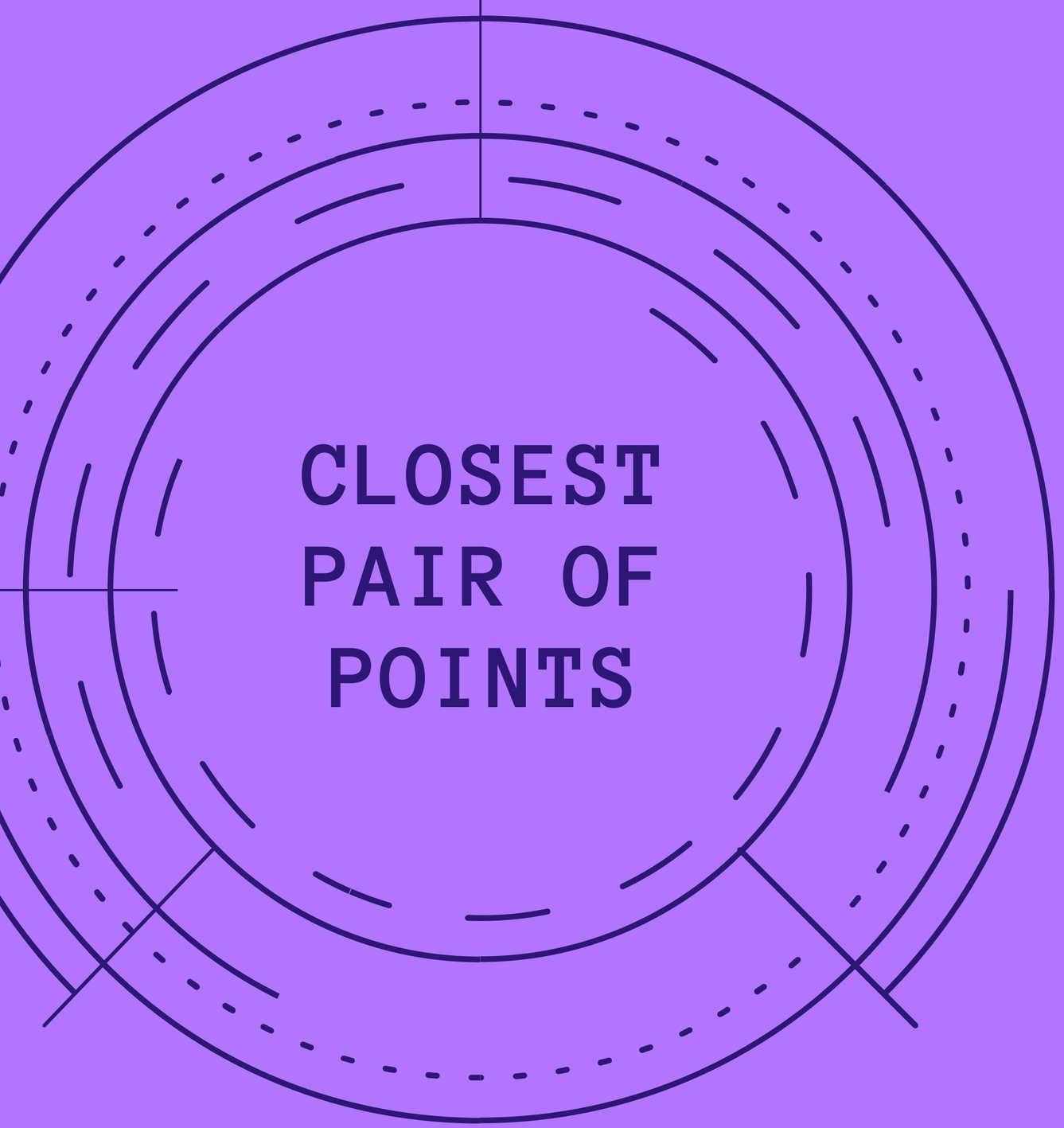
DESIGN AND ANALYSIS OF ALGORITHMS

NATIONAL INSTITUTE OF TECHNOLOGY DELHI

Presented by Nikita Uikey



CLOSEST PAIR OF POINTS

THE PROBLEM STATES-

We are given an array of n points in the plane, and the problem is to find out the closest pair of points in the array. This problem arises in a number of applications.
For example, in air-traffic control, you may want to monitor planes that come too close together, since this may indicate a possible collision.
Recall the following formula for distance between two points p and q.

DIVIDE AND CONQUER

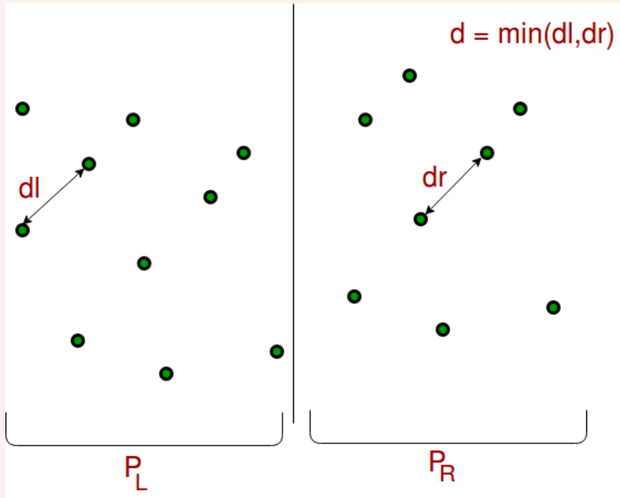
This technique can be divided into the following three parts:

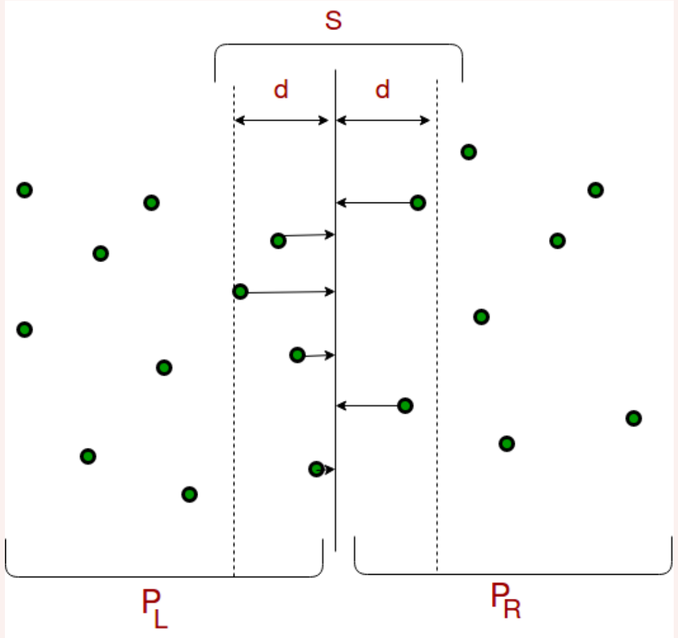
1. Divide: This involves dividing the problem into some sub problem.
2. Conquer: Sub problem by calling recursively until sub problem solved.
3. Combine: The Sub problem Solved so that we will get find problem solution.

DIVIDE AND CONQUER APPROACH

Following are the detailed steps of a O(n (Logn)^2) algortihm.
Input: An array of n points P[]Output: The smallest distance between two points in the given array.As a pre-processing step, the input array is sorted according to x coordinates.
1) Find the middle point in the sorted array, we can take P[n/2] as middle point.
2) Divide the given array in two halves. The first subarray contains points from P[0] to P[n/2]. The second subarray contains points from P[n/2+1] to P[n-1].

3) Recursively find the smallest distances in both subarrays. Let the distances be dl and dr. Find the minimum of dl and dr. Let the minimum be d.





4) From the above 3 steps, we have an upper bound d of minimum distance. Now we need to consider the pairs such that one point in pair is from the left half and the other is from the right half. Consider the vertical line passing through P[n/2] and find all points whose x coordinate is closer than d to the middle vertical line. Build an array strip[] of all such points.

5) Sort the array strip[] according to y coordinates. This step is O(nLogn). It can be optimized to O(n) by recursively sorting and merging.
6) Find the smallest distance in strip[]. This is tricky. From the first look, it seems to be a O(n^2) step, but it is actually O(n). It can be proved geometrically that for every point in the strip, we only need to check at most 7 points after it (note that strip is sorted according to Y coordinate). See this for more analysis.
7) Finally return the minimum of d and distance calculated in the above step (step 6)

BE INSPIRED

“

THANK YOU