CSC236 Fall 2016

Warm UP 07

In week 07, we are going to see one more D&C algorithm, *closestPairOfPoints*: assume there are n points in the 2D plane. The algorithm finds a pair of points that their distance is shortest compare to that of any other pair of points. To find the distance between two points $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$, we use the Euclidean distance: $d(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

A) Develop a brute-force algorithm, *closestPairOfPoints1*. All you need to do is to calculate the distance of each point to every other points and keep the minimum distance. What is the time complexity of this algorithm?
B) Try to develop a divide and conquer algorithm, *closestPairOfPoints2*, in which the plane is divided to two halves; find the closest pair in each and choose the minimum. Consider also the case that each point of the closest pair may be in a different half. What is the time complexity of this algorithm? Prove it.

Do it!

○

[1] Assume *n* pairs of points in the 2D plane are given. Determine the closest pair in an efficient way.