The closest pair of points in the plane

The problem: Given an array P of n points in the plane, find the closest pair. In case of ties, choose arbitrarily. Assume that the distance between two points p, q is given by the Euclidian distance, $d(p,q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$.

Problems/exercises

1. Formulate the 1D version of the closest pair. How can you solve it, and how fast? Try to extend this solution to the 2D problem: does it work?

   For the remaining problems we consider the 2D version of the probleme.

2. Consider a point $p \in P$. Show that, in order for a point q to be within distance d from p, then both the horizontal and vertical distance between p and q must be smaller than d.

   (Hint: assume, by contradiction, that this was not true, and show this leads to a logical impossibility)

3. Describe how you can find a vertical line L that splits P in half. How long does this take?

4. Show an example where the strip of width d around the middle vertical line L may contain $\Omega(n)$ points. What does this mean for the running time of the algorithm? Write a recurrence.

5. Consider the (refined) divide-and-conquer algorithm which takes as arguments the points in P sorted in two ways: let P_X and P_Y denote the points in P sorted by their x- and y-coordinates, respectively. Furthermore, let L be the vertical line that splits P into two halves, and let P_1 and P_2 be the set of points in P to the left/right of this line, respectively.

   (a) Given P_X and P_Y, how can we find the x-coordinate of line L ?

   (b) Given P_X and P_Y, how can we find P_{1X} (the points in P_1 sorted by their x-coordinates) and P_{2X} (the points in P_2 sorted by their x-coordinates)?

   (c) Given P_X and P_Y, how can we find P_{1Y} (the points in P_1 sorted by their y-coordinates) and P_{2Y} (the points in P_2 sorted by their y-coordinates)?