1. In this problem we will develop a divide-and-conquer algorithm for the following geometric task.

**Closest Pair**

*Input*: A set of points in the plane, \( \{p_1 = (x_1, y_1), p_2 = (x_2, y_2), \ldots, p_n = (x_n, y_n)\} \)

*Output*: The closest pair of points: that is, the pair \( p_i \neq p_j \) for which the distance between \( p_i \) and \( p_j \), that is,

\[
\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2},
\]

is minimized.

For simplicity, assume that \( n \) is a power of two, and that all the \( x \)-coordinates \( x_i \) are distinct, as are the \( y \)-coordinates.

Here’s a high-level overview of the algorithm:

- Find a value \( x \) for which exactly half the points have \( x_i < x \), and half have \( x_i > x \). On this basis, split the points into two groups, \( L \) and \( R \).
- Recursively find the closest pair in \( L \) and in \( R \). Say these pairs are \( (p_L, q_L) \in L \) and \( (p_R, q_R) \in R \), with distances \( d_L \) and \( d_R \) respectively. Let \( d \) be the smaller of these two distances.
- It remains to be seen whether there is a point in \( L \) and a point in \( R \) that are less than distance \( d \) apart from each other. To this end, discard all points with \( x_i < x - d \) or \( x_i > x + d \) and sort the remaining points by \( y \)-coordinate.
- Now, go through this sorted list, and for each point, compute its distance to the seven subsequent points in the list. Let \( (p_M, q_M) \) be the closest pair found in this way.
- The answer is one of the three pairs \( (p_L, q_L), (p_R, q_R), (p_M, q_M) \), whichever is closest.

(a) (10 pts) In order to prove the correctness of this algorithm, start by showing the following property: any square of size \( d \times d \) in the plane contains at most four points of \( L \).

(b) (10 pts) Now show that the algorithm is correct using induction. The only case which needs careful consideration is when the closest pair is split between \( L \) and \( R \).
(c) (10 pts) Write down the pseudocode for the algorithm and show that its running time is given by the recurrence:

\[ T(n) = 2T(\frac{n}{2}) + O(n \log n). \]

(d) (10 pts) Show that the solution to this recurrence is \( O(n \log^2 n) \).

(e) (10 pts) How can the running time be reduced to \( O(n \log n) \)?