BIM304 - Computer Algorithm Design
Assignment II

- ZIP file format: **YourNameAndSurname_Assignment2.zip**
- You have to upload your ZIP file until on **Sunday, May 01, 2022, until 23.59 to the MERGEN**.

Develop an application that finds the closest pair of a set of points recursively. Assume that there are at least two points in the input and that all the points are distinct.

If the input is:

4
1.0 1.0 → (1.0, 1.0) Point: X: 1.0, Y: 1.0
2.0 2.0 → (2.0, 2.0) Point: X: 2.0, Y: 2.0
4.0 5.0 → (4.0, 5.0) Point: X: 4.0, Y: 5.0
7.0 8.0 → (7.0, 8.0) Point: X: 7.0, Y: 8.0

then the output should be : (2, 2) - (1, 1) = 1.414214

Details

- There are 5 input files. The smallest file has 12 different points, while the largest one has 100,000 points. These files are in the ZIP file provided to you.
    1. SmallerSet.input → it contains 12 point
    2. test_100.input → it contains 100 point
    3. test_1000.input → it contains 1.000 point
    4. test_10000.input → it contains 10.000 point
    5. test_100000.input → it contains 100.000 point

- The format of the data is as follows:
    1. The first element specify the number of points in the input data
    2. Subsequent line contains the x-coordinate of a point followed by the y-coordinate, separated by a single space (check the given input files)

- The output must show the closest pair of points and the distance between them in the given format in example.
- The formula to compute distance between two points a=(x1,y1) and b=(x2,y2) is

$$distance = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

- Your algorithm must fit into the **divide-and-conquer** strategy.
- The running time should be **Θ(n·logn)** where n is the number of points
- Grouping is not allowed in this homework. Please obey the ethical rules.
- You need to write the codes suitable for the methods left blank in the given java files. The files provided to you are:
    1. PointPair.java → Do nothing
    2. TestClosestPointPair.java → Do nothing
    3. **QuickSort.java** → Write the required codes for the methods left blank. Never change the method names and descriptions.
    4. **FindClosest.java** → Write the required codes for the methods left blank. Never change the method names and descriptions.