Data structures 2018 coursework

In this work the purpose is to use Java and implement graph data structure that can be used in studying relations among two dimensional points. As a starting point you can use two files, T2018.java and Tdata.txt. T2018.java contains an example how to read floating point numbers form a text file and how to write them to the file. Tdata.txt contains example data from which you can create your graph. Implement your graph as a linked structure where a node can be given for instance in the form given below.

```
public class Node {

private float x; //x-coordinate of a point.

private float y; //y-coordinate of a point.

private Node neighbors[]; //Links to the nearest neighbors"

private boolean visited;  //For traversing purposes. Visited or not.

}
```

You can also use other variables and methods that you think are useful.

Your graph implementation should do the following tasks.

1. Create a graph by reading through the file that contains the points on 2D plane. For each point create a node that contains the point. In addition, for each node find another node that contains a point that is closest to the node in question. (Nearest neighbor). This procedure yields two kind of subgraphs. Subgraphs containing two nodes, and sub graphs containing three nodes. (Connected components)
2. For each node of the graph add link to the second closest neighbor
3. Write file BFS.txt, that contains coordinates of the graph nodes in Breadth First order
4. Write file DFS.txt, that contains coordinates of the graph nodes in Depth First order
5. Write file Degrees.txt, that contains the in- and out degrees of the nodes in Depth First order
6. Remove a given node from the graph and write remaining graph to the file DIM.txt in Breadth First order
7. Add more nearest neighbors to the graph nodes, until the graph becomes fully connected. This means that from a random node there exists a path to all other nodes of the graph and back to the starting node. Write your graphs in the file COMP.txt in Breadth First order
8. Create a minimum spanning tree to your graph and write it in the file MSP.txt using Breadth First order
9. In the first task the result was a set of subgraphs of size 2 and 3. What could be the role of 3-node subgraphs in the original graph if we consider the relations of the points?
10. Change a bit the way how you calculate the distances of the points and create a graph that contains the points on the outskirts of the point cloud. See Fig 1. You can determine the definition for outlier by yourself. Write your graph into file OUTLIERS.txt in Breadth First order

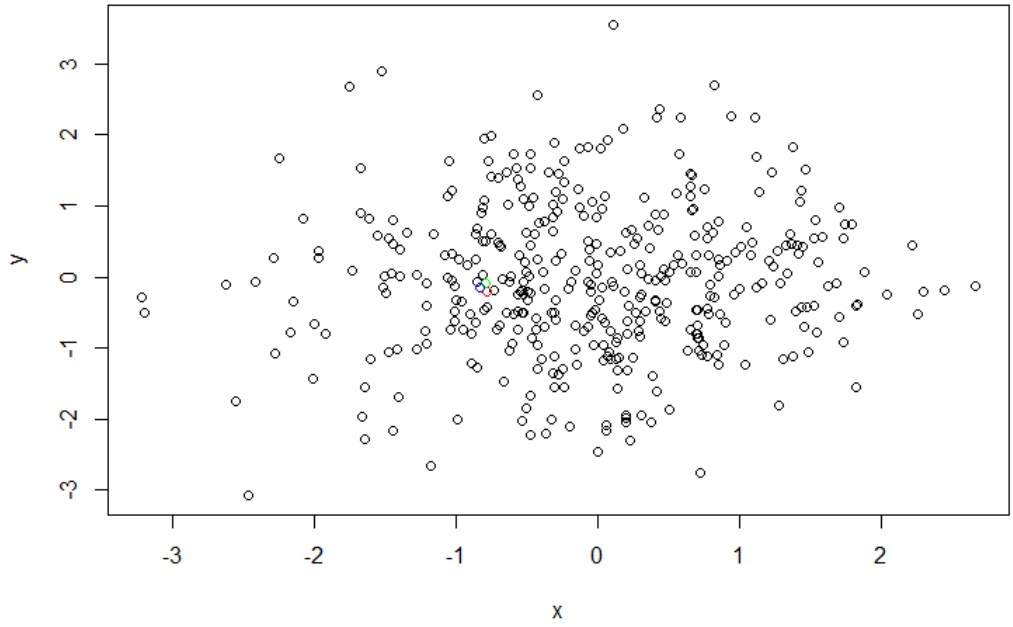

Fig. 1. Points that reside on the outskirts of the point cloud may be ”outliers”. Such points are usually removed from the data.

Returning your coursework

Return your work before 31.1.2019. Use email jyrki.rasku(at)uta.fi and topic Tira2018. Pack your source code in to one file using 7z or zip format. Your work should compile using javac T2018.java and should be run using command java T2018. You can embed your answer to task 9 as a comment to the beginning of your source file. In the returning message list the tasks that you have done.

About grading

A work that does all the tasks mentioned above is worth of ten points. Minimum requirement is to implement tasks 1-3.