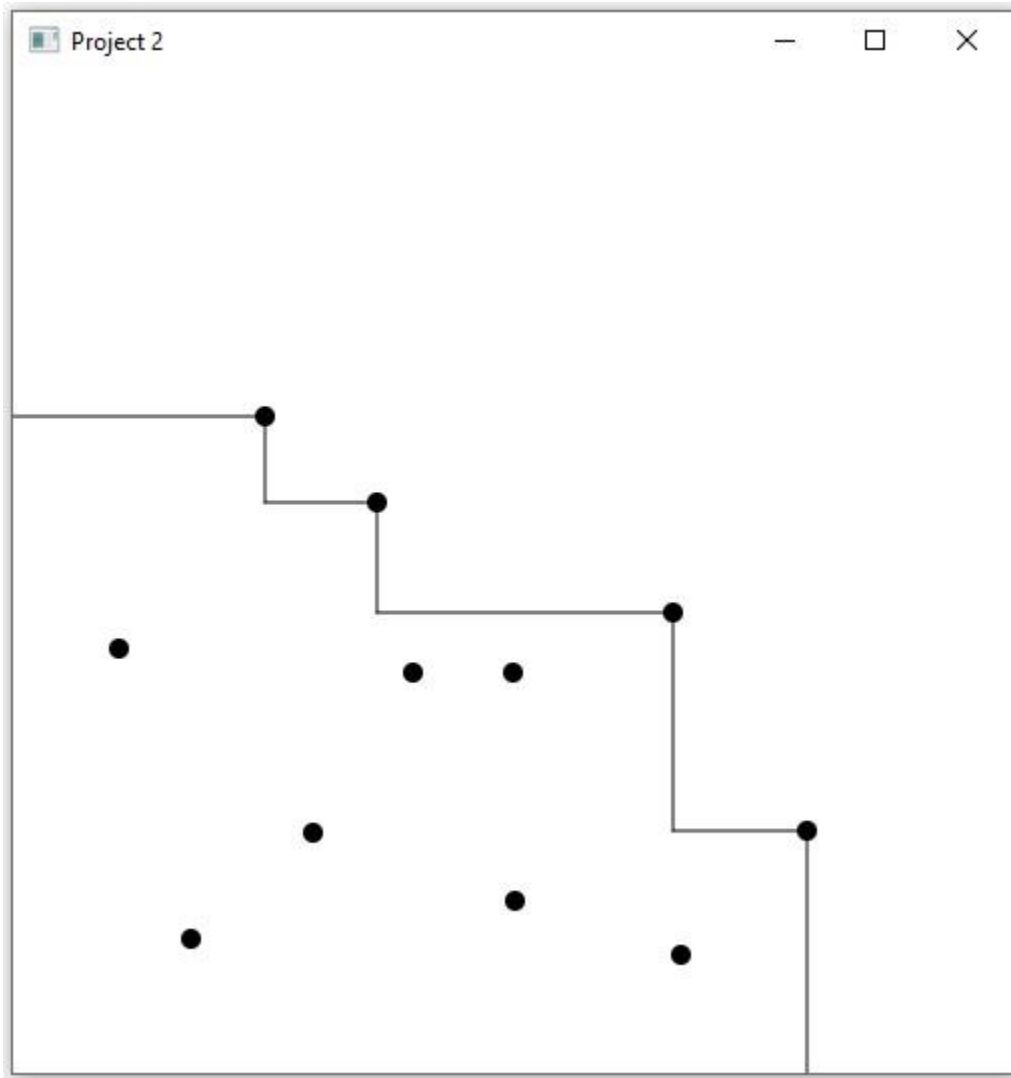


CMSC 315 Data Structures and Algorithms

Programming Project 2

The second programming project involves writing a program to determine the set of maximal points among a set of points displayed in a Java window. Shown below is an example:



A point is considered maximal if there are no other points above it or to the right of it. The maximal points are to be connected by lines as shown in the above example. Initially the set of points is to be read from a file whose name can be hard-coded. The following input file should produce the above window:

```
200.0 300.0
250.0 300.0
330.0 270.0
150.0 380.0
126.0 172.0
397.0 379.0
334.0 441.0
```

```
53.0 288.0
89.0 433.0
182.0 215.0
251.0 414.0
```

After the initial set of points have been read in, the point set should be able to be modified with the mouse. Clicking the left mouse button should add a point at the clicked location and a right click should remove a point. After a point has been either added or removed, a new set of maximal points should be determined and connected with lines.

This program should consist of three classes. The first class should have two instance variables of type `double` that represent the x and y coordinates of the point. It should be an immutable class that implements the `Comparable` interface with the following public methods:

- A constructor that initializes the x and y coordinates of the point
- A method that returns the x ordinate
- A method that returns the y ordinate
- A method that is passed a second point and returns true if the second point is below and to the left of the point on which it is invoked and false otherwise
- A `compareTo` method that compares only the x ordinates of the two points

The second class should extend the `javafx.Pane` class. It should contain the following methods:

- A constructor that is supplied an array list of points that produces the initial point set and determines the maximal points and connects them
- A private event handler that handles mouse clicks that adds a point with a left click, removes a point with a right click and recomputes the maximal point set afterward
- A private method that finds the maximal set and draws the lines that connect them

The method that compares whether a point is below and to the left of another should be used to determine which points are in the maximal set. The `compareTo` method should be used to sort the maximal set of points so that they can be connected. You can use the predefined `sort` method in the `Collections` class to perform the sort. The pane should be 500 pixels by 500 pixels.

The third class defines the scene that contains the pane which is created with the initial set of points that are read in from the file `points.txt` by the main method.

You are to submit two files.

1. The first is a `.zip` file that contains all the source code for the project. The `.zip` file should contain only source code and nothing else, which means only the `.java` files. If you elect to use a package the `.java` files should be in a folder whose name is the package name. Every outer class should be in a separate `.java` file with the same name as the class name. Each file should include a comment block at the top containing your name, the project name, the date, and a short description of the class contained in that file.

2. The second is a Word document (PDF or RTF is also acceptable) that contains the documentation for the project, which should include the following:
 - a. A UML class diagram that includes all classes you wrote. Do not include predefined classes.
 - b. A test plan that includes test cases that you have created indicating what aspects of the program each one is testing
 - c. A Big-O analysis of the time needed to determine the set of maximal points
 - d. A short paragraph on lessons learned from the project