CS 3410 - Homework 02

Due date: see course Schedule and Blackboard.

Overview

Consider pouring a bucket of water on a particular place on the earth. What path will it take as it flows (conceptually) to the ocean? Or, does it get trapped, in a bowl, a sink somewhere? We know that the water will follow the path of least resistance, so it will flow from one region to another if the elevation declines. Suppose we have a map of elevations as shown in the examples below:

	0	1	2	3	4	5	6
0	900	900	900	900	900	900	900
1	900	900	900	900	900	900	900
2	900	900	700	600	500	900	900
3	900	900	800	900	400	900	900
4	900	900	900	900	300	200	100
5	900	900	900	900	900	900	900
6	900	900	900	900	900	900	900

Suppose the water is dropped at 4,2, then the path the water follows is shown in yellow. We will assume that the ocean surrounds the map so that if the water reaches the edge, it will always fall into the ocean.

	0	1	2	3	4	5	6
0	900	900	900	900	900	900	900
1	900	900	900	300	400	900	900
2	900	900	900	900	500	600	900
3	900	900	900	900	900	700	900
4	900	900	900	900	900	800	900
5	900	900	900	900	900	900	900
6	900	900	900	900	900	900	900

Suppose the water is dropped at 4,4. In this case the water gets trapped at position 1,3. In other words, it doesn't make it to the ocean.

	0	1	2	3	4	5	6
0	900	900	900	900	900	200	900
1	900	900	900	900	900	300	900
2	900	900	900	600	500	400	900
3	900	900	900	700	900	900	900
4	600	700	800	900	900	900	900
5	900	900	900	900	900	900	900
6	900	900	900	900	900	900	900

In some cases there may be more than one path that the water can flow, for instance if you start at 4,3 above. In the easier version of this problem, you will just determine if there is a path to the ocean, given a map and starting point. In this case, discovery of either of these paths will answer the question. In the more advanced version, you will have to find the path that ends, for instance, at 0,5. In the more advanced problem, whenever there is a choice (4,3), we will take the path with the largest drop.

	0	1	2	3	4	5	6
0	900	900	900	900	900	900	900
1	900	900	900	300	400	900	900
2	900	900	900	900	500	600	900
3	900	900	900	900	900	700	900
4	900	900	900	900	900	800	900
5	900	900	900	900	900	900	900
6	900	900	900	900	900	900	900

As stated before, if the water reaches an edge, it will always fall off. Thus, it is convenient to think of the ocean outside the boundary of the map as having an infinitely low elevation. If the water is dropped at 2,0, then it will fall off the map. However, if it is dropped at 2,1 then it is trapped.

There are two versions of this assignment. The "B" version has a maximum grade of 85. The "A" version has maximum grade of 100.

B Version

You will write the method, *willFlow* which accepts a map and starting location as input and returns whether the water will reach the ocean. The map is read from a text file whose name is supplied on the command line. The starting point (row,col) is also specified on the command line. You have been supplied with the code below and a sample map.

```
import java.util.*;
import java.io.*;

class WaterFlow
{
   static long count = 0;

   public static void main( String[] args ) throws FileNotFoundException
   {
      // Get file name for map and starting point from command line.
      String fileName = args[0];
      int startRow = Integer.parseInt( args[1] );
      int startCol = Integer.parseInt( args[2] );

      // Create elevation map.
      ElevationMap map = new ElevationMap( fileName );

      // See if the water will flow off map
      boolean mapWillFlow = willFlow( map, startRow, startCol );

      String msg = "Map file: " + fileName + ", Starting point: " + startRow + ", " + startCol;
      System.out.println( msg );
      msg = "Water will fall off the map? " + mapWillFlow;
      System.out.println( msg );

      map.display();
   }

   public static boolean willFlow( ElevationMap map, int startRow, int startCol )
   {
      // You write this method, recursively.
      return false;
   }
}
```

```
// Class represents a 2-d map of elevations
class ElevationMap
{
   int rows=0, cols=0;
   double[][] elevation;

   ElevationMap( String fileName ) throws FileNotFoundException
   {
      // Code to create a map from a text file.
      ...
   }
   // Method to display map.
   void display()
   {
      ...
   }
   // Method to see if a row coordinate is off the map
   boolean isRowOffMap( int row )
   {
      ...
   }
   // Method to see if a column coordinate is off the map
   boolean isColOffMap( int col )
   {
      ...
   }
}
```

A Version

In addition, your method will create a Path object that represents the best path to the ocean. You have been supplied with the code below (WaterFlowA.java).

The driver code is altered:

```
      // Create elevation map.
      ElevationMap map = new ElevationMap( fileName );

      Path path = new Path();

      // See if the water will flow off map
      boolean mapWillFlow = willFlow( map, startRow, startCol, path );


      String msg = "Map file: " + fileName + ", Starting point: " + startRow + ", " + startCol;
      System.out.println( msg );
      msg = "Water will fall off the map? " + mapWillFlow;
      System.out.println( msg );

      if( mapWillFlow ) path.print();

      map.display();
```

In addition, you have been supplied these classes:

```
// Class represents the best path to the ocean.
class Path
{
   ArrayList<Step> steps = new ArrayList<Step>();

   void print()
   {
      System.out.println( "Best Path" );

      for( Step s : steps ) System.out.print( s + ", " );
      System.out.println();
   }
}
// Class represents a step in a path
class Step
{
   int row=-1, col=-1;

   Step( int row, int col )
   {
      this.row = row;
      this.col = col;
   }

   public String toString()
  {
      return "(" + row + ", " + col + ")";
   }
}
```

Deliverables

1. A single java file with the name WaterFlow.java (or WaterFlowA.java).
2. A Word document in this format:
   a. Title page:
      CS 3410 – HW 02
      Name
      Date
   b. An summary of your progress on this homework.
   c. An annotated sequence of testing results. You should conduct at least 6 tests.

3. All text files used for testing that are referred to in the testing results document.
4. Zip all files and submit via Blackboard.