COT 4400: Analysis of Algorithm
Problem Modeling: Individual Project (100 pts)
Due: November 19, 2013

This project requires you to model the problem below as graph and then use a known graph algorithm to solve the problem. You will use a graph algorithms library rather than implement the known graph algorithm yourself.

You are not allowed to use the Internet or consult any references.
This is an individual project.

1. *Problem Description*:

   This problem is based on the "Jumping Jim" maze problem (from "MAD MAZES: Intriguing Mind Twisters for Puzzle Buffs, Game Nuts and Other Smart People" by Robert Abbott, Bob Adams, Inc. Publishers, 1990). The text of the problem is quoted below. A diagram of the maze is provided separately.

   > Jumping Jim is about to begin his grand performance at the circus, but his jealous enemy, Dastardly Dan, has restrung all the trampolines. The number on each trampoline indicates how tightly strung each one is; in other words, the number indicates how far Jim will have to move (horizontally or vertically, but NOT diagonally) when he bounces off the trampoline. Jim begins his routine by leaping onto the trampoline at the upper left. He must get to the Goal at the lower right, where he will take his bow. How can he get there?
   >
   > The diagram shows all of the trampolines with each represented as a square. Begin on the square at the upper left. That square is marked 3. From there you could, for example, move three squares down to a square marked 2. From there, you might move two squares right to a square marked 4, and from there you could move four squares right to another square marked 2. That path, incidentally, won't get you to the goal.

- The Input: (`input.txt`)
  The input file contains multiple instances. The input file begins with a single positive integer on a line by itself indicating the number of instances following, each of them as described below. This line is followed by a blank line, and there is also a blank line between consecutive instances.

  Each instance begins with a single positive integer on a line by itself indicating the dimension, n, of the maze. The following lines contain the values for each square of the maze. There are n lines which each have n integer values. 0 represents the goals square. For the maze example provided, the input is:

```
7
3 6 4 3 2 4 3
2 1 2 3 2 5 2
2 3 4 3 4 2 3
2 4 4 3 4 2 2
4 5 1 3 2 5 4
4 3 2 2 4 5 6
2 5 2 5 6 1 0
```

  The maximum dimensions for a maze is $n = 50$[1].

- The Output: (<`studentLastName`>`.txt`)
  The output file should contain one line for each instance. There should be a blank line between consecutive outputs.

  Each output should consist of a path from the top left square to the bottom right square. There should be a single line of output which indicates the path Jim should take. Each move should be represented using `N, S, E, W` to represent "north", "south", "east", and "west", respectively. Each move should be separated by a space.

  Example: Suppose your first move takes you 3 places south to the square marked 2 and your second move takes you east to the square marked 4. This sequence of two moves should be displayed on your output as follows:

```
S E
```

  Note that if there is more than one path for Jim to take then any path will suffice.

You are welcome to try figuring out the solution to the puzzle on your own, but that won't get you any points. Your assignment is to model the maze as a graph and to solve the problem using an appropriate graph algorithm that we've encountered in class.

[1] If your algorithm/code is too slow to handle inputs of the prescribed size you will lose points for both your algorithm design grade AND your results grade

2. *Deliverables*:

   Please submit a **hard copy** of all of the items requested below. Please don't send this by email as I'll have to print it out anyway. Please also submit *code*[2] and *typed project report*[3] to Canvas.

   (a) **Problem Modeling [50 pts]**:

   - **[15 pts]**: Explain how you modeled the problem as a graph.
     For full credit your explanation must be descriptive enough that any competent programmer will be able to create a graph of any valid input based on your description and understand the reasoning behind how the graph is created.
   - **[10 pts]**: Draw the resulting graph.
     For full credit you should draw the resulting graph for the instance of the maze provided by the problem statement. This may be hand-drawn
   - **[10 pts]**: Identify the known graph algorithm needed to solve the problem.
   - **[15 pts]**: Prove that this algorithm will actually solve the problem.
     For full credit you must provide a formal proof that this algorithm will produce the correct solution for any valid input to the problem (i.e., you will need to prove that how you modeled the problem as a graph accurately represents the problem and that the algorithm you identified will produce the correct solution on this graph.)

   (b) **Code [20 pts]**:

   Submit a print-out of your code. If you are unable to get your code to compile/run, please state this explicitly.
   Code Requirements[4]:

   - Read the input from a file in the described format.
   - Output the solution in the described format.
   - README file describing how to compile and run your code. If your code requires more than a simple command to compile and run then you must also provide a Makefile and/or shell script.
     A simple command should only include the compiler to be used and the name of a single file. If your command requires any flags or directives then you must provide a Makefile and/or shell script. For example `g++ main.cpp` *is a simple command.*
   - *Optional: You may also submit an executable for the C4 Linux Lab machines*
   - You must use an algorithm from an approved graph algorithms library.

   For full credit your code must follow all of the listed requirements.

   (c) **Results [30 pts]**:

   Submit your code (in addition to your code you may submit an executable) to Blackboard to be run on a set of inputs devised by the grader. Your grade will be based on whether your code produces the correct output for each of the inputs devised by the grader.

   (d) **Penalty [-10pts]**: If you choose to implement the known graph algorithm yourself then a 10 point penalty will be assessed.

   (e) **Extra Credit [10pts]**: Extend your implementation to solve the "Jumping Jim's Encore" maze problem (from "MAD MAZES: Intriguing Mind Twisters for Puzzle Buffs, Game Nuts and Other Smart People" by Robert Abbott, Bob Adams, Inc. Publishers, 1990). Note that you are only eligible to earn the extra credit if you are using a known graph algorithm implemented by an approved graph algorithms library (i.e., you have not taken the 10 point penalty above).

---

[2] Your code must compile and run on the C4 Linux Lab machines

[3] Project report and code will be checked for plagiarism

[4] Failure to follow these requirements may result in a 0 for the code portion of your grade AND a 0 in the results portion of your grade

Extra Credit

This problem is based on the "Jumping Jim's Encore" maze problem (from "MAD MAZES: Intriguing Mind Twisters for Puzzle Buffs, Game Nuts and Other Smart People" by Robert Abbott, Bob Adams, Inc. Publishers, 1990). The text of the problem is quoted below. A diagram of the maze is provided separately.

> Our story so far: In Maze 7, Dastardly Dan had tried to sabotage Jumping Jim's act by restringing all his trampolines. But Dan's actions had the opposite effect. The audience was so delighted by Jim's leaping about trying to reach the goal, that his act became the most popular of the circus.
>
> The circus owner decided to commission Dastardly Dan to create another, even harder maze for Jim to solve. Dan added more trampolines, restrung them, and painted large numbers on each to indicate how far Jim will move when he bounces off each trampoline. The painted numbers would allow the audience to study the maze and try to find a solution before Jim did.
>
> Dan also added a new rule that Jim had to follow. Certain of the numbers were painted in red and enclosed in circles. When Jim begins his act, he can move only vertically or horizontally through the maze of trampolines; he cannot move diagonally. However, if he lands on a red number in a circle, he must then start moving only diagonally; now he can't move vertically or horizontally. Jim must continue moving diagonally until he again lands on a red number. He then switches back to moving only vertically or horizontally. And he switches each time he lands on a red number.
>
> Here's an example to show how that works. Jim begins on the 4 at the northwest corner of the maze. From there he might move south four squares to the red 3. Now he must start moving diagonally. He might go three squares northeast to a 4. On the next move he would continue moving diagonally. He could move four squares southeast to a red 1. That red number would cause him to switch back to moving only horizontally or vertically.
>
> Can you find a route that would let Jim land on the trampoline marked GOAL?

- The Input: (`input.txt`)
  The input file contains multiple instances. The input file begins with a single positive integer on a line by itself indicating the number of instances following, each of them as described below. This line is followed by a blank line, and there is also a blank line between consecutive instances.

  Each instance begins with a single positive integer on a line by itself indicating the dimension, n, of the maze. The following lines contain the values for each square of the maze. There are n lines which each have n integer values. 0 represents the goals square. If it is a red square it has a 'R' after the integer. For the maze example provided, the input is:

```
 8
 4   2   2R  4   4   3R  4   3R
 3   5   3   4   2   3   5   2R
 4   3   2   5R  2   2   5   2
 7   1   4   4   4   2   2   3
3R   2   2   4   2   5   2   5
 2  3R   2   4   4   2   5   1R
 6   2   2   3R  2   5   6   3
 1   2R  5   4   4   2   1R  0
```

  The maximum dimension for a maze is $n = 50$[5].

- The Output: (`<studentLastName>.txt`)
  The output file should contain one line for each instance. There should be a blank line between consecutive outputs.

  Each output should consist of a path from the top left square to the bottom right square. There should be a single line of output which indicates the path Jim should take. Each move should be represented using `N, S, E, W, NE, NW, SE, SW` to represent "north", "south", "east", "west", "northeast", "northwest", "southeast", and "southwest", respectively. Each move should be separated by a space.

  Example: Suppose your first move takes you 4 places south to the square marked 3R and your second move takes you northeast to the square marked 4. This sequence of two moves should be displayed on your output as follows:

```
S NE
```

You are welcome to try figuring out the solution to the puzzle on your own, but that won't get you any points. Your assignment is to model the maze as a graph and to solve the problem using an appropriate graph algorithm that we've encountered in class.

[5] If your algorithm/code is too slow to handle inputs of the prescribed size you will lose points for both your algorithm design grade AND your results grade