BBM204

Take-Home 4

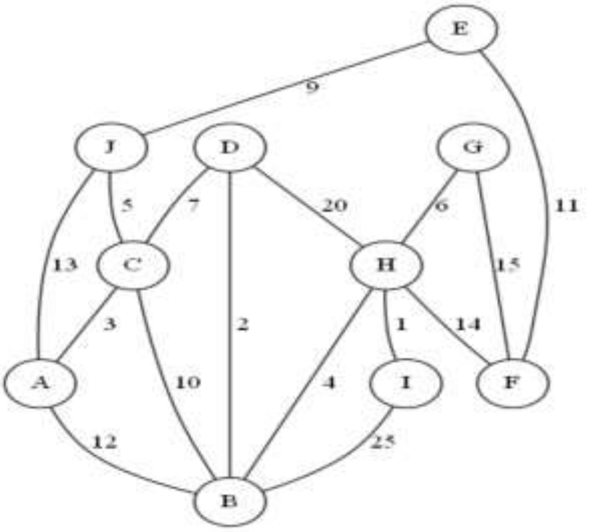
Submission Date: 14.05.2020

Due Date: 17.05.2020 23:59

Programming Language: Java

QUESTIONS

1. Assume that we're running the Prim's algorithm and computing the Minimum Spanning Tree of the following graph by using lazy version of Prim's algorithm starting at node A. Write down the edges of MST graph in the order that both lazy version and eager version of the algorithm of Prim's adds to MST. Use the format (node1, node2) to specify the edges of the MST. Your output should be as shown below. Report file should include your outputs.



Sample Output:

Lazy Version	Eager Version
A-C	A-C

2. Design and write a worst-case generator for edge-weighted graphs with $V=10$ vertices and E edges such that the running time of the lazy version and eager version of Prim's algorithm is nonlinear. In the report you will write, you should prove your answers to the questions given below by showing them on the code you wrote and print the graph you created for the worst case scenario for both eager version and lazy version of Prim's algorithm.
   a. What is necessary to satisfy a worst-case for the eager version of Prim's algorithm?

b. What is necessary to satisfy a worst-case for the lazy version of Prim's algorithm?
c. What is the worst-case complexity for the eager version of Prim's algorithm?
d. What is the worst-case complexity for the lazy version of Prim's algorithm?

You can use following code snippet for print your graph you created.

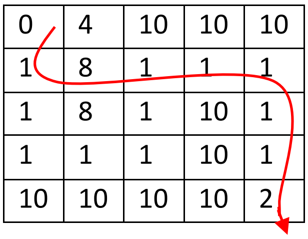
```java
public String toString() {
        StringBuilder stringBuilder = new StringBuilder();

        for(int vertex = 0; vertex < vertices(); vertex++) {
            stringBuilder.append(vertex).append(": ");

            for(Edge neighbor : adjacent(vertex)) {
                stringBuilder.append(neighbor).append(" ");
            }
            stringBuilder.append("\n");
        }

        return stringBuilder.toString();
    }
```

3. Suppose that you have a maze game that is given as N*N positive integers matrix of blocks. maze[0][0] is the source block(upper leftmost block) and maze[N-1][N-1] is the destination block (lower rightmost block). The goal of the game is to control the movements of a *hero* inside a closed 2D generated maze. The hero will begin the game from a source block and the game will end when the hero finds the shortest path from the source block to the destination, where the length of the path is the sum of the integers in the path. The hero can move only in two directions: **right and down.**

- **Sample input**

```
5
0,4,10,10,10
1,8,1,1,1
1,8,1,10,1
1,1,1,10,1
10,10,10,10,2
```

The first line of the input file holds one integer, N, showing the number of rows and columns (N*N). In the following N lines, weights of blocks are given, N integers in every line.



0	4	10	10	10
1	8	1	1	1
1	8	1	10	1
1	1	1	10	1
10	10	10	10	2

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

Maze which is created according to input.txt is shown left-hand side above. The maze shown on the right-hand side shows the index placement of each block. While creating your output, you have to show your path according to these indices. The shortest path of this maze is shown with the red line. And the shortest path is indicated by indices below.

- **Sample Output**

Print the answer which is the shortest path from source block to destination block.

```
Path:0->5 5->6 6->7 7->8 8->9 9->14 14->19 19->24
```

NOTES

- For the second and last question, you have to submit java file. And in your report, you have to prove your answers to the questions by showing them on the code you wrote.
- For the questions given above, you should call the functions you wrote from a single main function. You should read only one input.txt file for the third question and write the output of each question to the same output.txt file. The format of the output file is shown below.
  Output.txt sample format:

```
Q1 output:

//Your output of Q1

Q2 output:

//Your output of Q2
```

- For testing your program command-line parameters should be as shown below:

```
>javac Main.java

>java Main <input.txt><output.txt>
```

- No late submission is permitted.
- Your programs are going to be tested on the dev machine.
- Use submit system that has been configured for you. No other medium except Submit System will be accepted.
- Use UNDERSTANDABLE names for your variables, functions, and classes (please be sure you are obeying name convention).
- Write READABLE SOURCE CODE blocks.
- Use EXPLANATORY COMMENTS in your source codes.
- Your work must be original. Duplicate or very similar works are both going to be considered as cheating.
- **You can use the input specifications and graph classes you use in the course.**
- You will submit your work from https://submit.cs.hacettepe.edu.tr with the file hierarchy as below:

```
Src|--

          |--  Main.java

          |-- .java* (optional)

          |-- output.txt

          |-- report.pdf
```