Assignment 4 - Due on Nov 7th 2021

Github link : https://github.com/kpavan95/CS5343/tree/main/src/com/assignments/assignment4/

Requirements

JDK 8 and above installed

Zip file

As part of the submission, a zip file will be provided which will have 3 items
- Assignment folder: It's a java project which contains the code . Note: you can also go to the above Github link and look at the code.
- assignment4.jar: It's the executable file to run the program. Instructions to run it are provided below in the Execution Instruction section
- This readme file in pdf format

Build Program

Note: the executable jar is already provided as part of the assignment submission so the jar can be downloaded and this part can be skipped.

- Download the source folder submitted or download it from the github link.
- Open terminal and change directory to the root folder
- Create a binary folder to store the .class file

> mkdir bin

- Compile the .java files using the following command

> javac src/com/assignments/assignment4/*.java -d bin/

- change directory to bin folder

> cd bin/

- Create a Executable jar file using the .class files

> jar cfe assignment4.jar com/assignments/assignment4/Assignment4 com/assignments/assignment4/*.class

now the executable assignment4.jar is ready to use.

Execution Instruction

To run the program, enter the following command in terminal where the executable jar is located

> java -jar assignment4.jar

The screenshot below shows the program's execution in different stages

```
Pavans-MacBook-Air:bin pavankumar$ java -jar assignment4.jar
Welcome to Programming assignment 4.
We are going to look into the workings of Dijkstra's algorithm
Printing Graph.
List of Vertices:0,1,2,3,4,5,6,7,8,9
 List of Edges:Graph (From ---(Weight)---> To):
0 ---(22)---> 7
0 ---(23)---> 9
0 ---(8)---> 1
1 ---(27)---> 6
1 ---(17)---> 5
1 ---(12)---> 2
2 ---(14)---> 4
2 ---(11)---> 5
2 ---(9)---> 7
3 ---(28)---> 7
3 ---(25)---> 2
3 ---(24)---> 9
4 ---(15)---> 1
4 ---(11)---> 0
4 ---(15)---> 3
5 ---(7)---> 6
5 ---(6)---> 8
5 ---(6)---> 0
6 ---(3)---> 1
6 ---(1)---> 3
6 ---(13)---> 3
7 ---(23)---> 0
7 ---(14)---> 8
7 ---(12)---> 9
8 ---(6)---> 4
8 ---(16)---> 2
8 ---(9)---> 7
9 ---(19)---> 6
9 ---(22)---> 5
9 ---(18)---> 0
```

Figure 1: Randomly generated Graph

```
Dijkstra initiated

Printing Tree.
List of Vertices:0,1,2,3,4,5,6,7,8,9
 List of Edges:tree (From ---(distance From Source 0)---> To):
0 ---(8)---> 1
1 ---(20)---> 2
6 ---(33)---> 3
2 ---(34)---> 4
1 ---(25)---> 5
5 ---(32)---> 6
0 ---(22)---> 7
5 ---(31)---> 8
0 ---(23)---> 9

Pavans-MacBook-Air:bin pavankumar$
```

Figure 2: Tree Generated using Dijkstra's algorithm