HACETTEPE UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING
BBM204
PROGRAMMING ASSIGNMENT #5

Subject **:** Path Traveler Problems
Advisor **:**
Submission Date **:** 4.6.2020
Due **:** 18.6.2020
Prog. Language **:** Java (Eclipse will be used to evaluate)

1) DESCRIPTION

Switzerland, being a small country, is composed of different cantons. In these cantons there exist various cities. Suppose that there is a medical cargo company named “Medica” which works in Switzerland and carries various medical tools and drugs for several pharmacies and hospitals across the country. In this assignment, we deal with several graph traversal strategies according to the various requirements.

In this problem, there will be D different nodes (i.e. delivery points – pharmacy or hospital) and E two-way undirected edges. It should be noted that the delivery points will be numbered with integers from 1 to D such that the node 1 will be the starting point for all cargo carriers. At the end of the day, all carriers will end up their trip at S^{th} node (stop node). Moreover, all undirected edges have a time value indicating the required time to complete to travel on the edge.

The owner of the company “Medica” wants to test 4 different path traversal strategies for 4 different carriers. The strategy for each carrier type is described below:

- **1^{st} Carrier:**
  The aim of the first type of carrier is to complete his/her trip as soon as possible in terms of total duration. Thus, the carrier wants to follow the shortest path. Keep in mind that, the driver can use the same edge for only once and the journey lasts when the driver reaches to S^{th} node.

- **2^{nd} Carrier:**
  The second type of carrier does not prefer to drive through long edges such that he/she wants to complete the total trip by driving on the edges which need shorter time. Thus, in this setting, the objective is to minimize the duration of the maximum edge the driver takes in his/her way to the node S. Note that the total duration of the path you choose (the time it takes to go from the starting point to the destination) must be less than the duration of other possible paths. And you must print out the duration of the maximum edge. Keep in mind that, the driver can use the same edge for only once and the journey lasts when the driver reaches to S^{th} node.

- **3^{rd} Carrier:**
  This type of carrier does not pay attention to the time that any edge takes such that all edges are the same for the driver. In this setting, the goal is to visit as few nodes as possible. Strictly speaking, the driver wants to look for a path to the S^{th} node which

requires a minimum number of edges to travel on. Keep in mind that, the driver can use the same edge for only once and the journey lasts when the driver reaches to S^{th} node.

- **4^{th} Carrier:**
  In this regime, when the driver is in a node he/she selects the shortest edge out of that node. Nonetheless, keep in mind that, the driver can use the same edge for only once and the journey lasts when the driver reaches to S^{th} node. Besides, note that, if there exist two shortest edges having an equal length, the driver tends to select the one having the lower edge number. So we can conclude that this type of carrier is greedy.

All these carriers need different methods to travel and you must compute the path for each carrier. Accordingly, we can summarize the objectives you should achieve for each carrier type as follows:

1. The first carrier's driver aims to know the shortest time to arrive at the S^{th} node.
2. The second carrier's goal is to know the "maximum duration edge" on the way to the final destination while having the lowest total duration on the trip.
3. The driver of the third carrier aims at having a minimum number of edges during the trip from beginning to S^{th} node.
4. The fourth carrier wants to know the time to arrive at S^{th} node when its driver employs a greedy algorithm. Moreover, if it cannot arrive at S^{th} node, you can use the value of -1 to print out as a result.
5. Except for the second carrier, you should print out the path that the driver follows.

2) PROBLEM DEFINITION AND DETAILS

Your duty for this assignment is to implement various algorithms for each carrier type and process the input file given by an argument to output the total travel duration on the console.

The information you will require to know about the structure of the input file is given below:

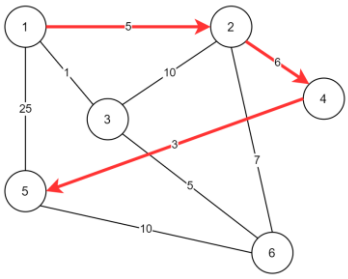
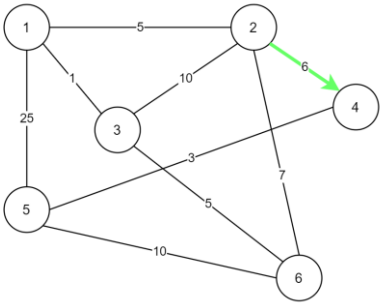
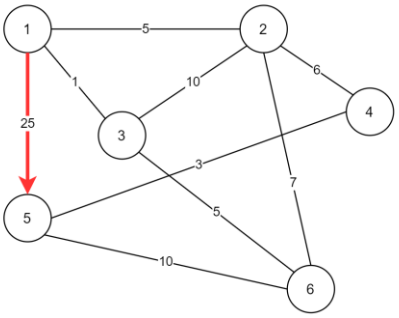
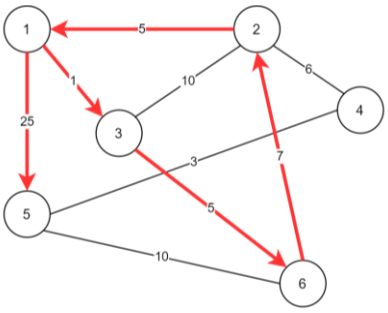
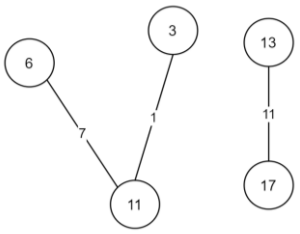
- The input file is composed of many lines.
- There exist four parameters given as integer values in the first line of the input file such as (C D E S).
- The value of C indicates the carrier type (1,2,3,4)
- The value of D represents the number of delivery points (a.k.a nodes)
- The E indicates the total number of edges in the map
- The S indicates the node number of the final arrival node S.
- The remaining lines involve 3 integers delimited by a space character (D_x D_y T_i) to represent the edge/path from delivery point D_x to delivery point D_y with the T_i (i.e. the time to travel this edge)

Your program will be composed of a single code file named "HW5.java" and it needs to take one single input file with the extension of ".txt" as a startup argument. The output will be printed out on the console window.

It is important to know which algorithm was used for each carrier type. We, therefore, want you to specify which algorithm you have implemented and used for each task along with stating why you picked up that algorithm. In other words, you must also briefly introduce the algorithm with its name and the reason why you have chosen it. We need them as comments.

For the evaluation of your assignment, we will test each carrier type with different input files. We will also share some test cases for you to guide you throughout the experiments you conduct.

Some sample input/output pairs are provided below:

1 6 9 5 1 2 5 1 3 1 3 2 10 1 5 25 2 4 6 4 5 3 3 6 5 2 6 7 5 6 10	The output: 14 1->2->4->5	
2 6 9 5 1 2 5 1 3 1 3 2 10 1 5 25 2 4 6 4 5 3 3 6 5 2 6 7 5 6 10	The output: 6	
3 6 9 5 1 2 5 1 3 1 3 2 10 1 5 25 2 4 6 4 5 3 3 6 5 2 6 7 5 6 10	The output: 1 1->5	
4 6 9 5 1 2 5 1 3 1 3 2 10 1 5 25 2 4 6 4 5 3 3 6 5 2 6 7 5 6 10	The output: 43 1->3->6->2->1->5	
2 27 3 27 6 11 7 3 11 1 13 17 11	The output: -1	

3) GRADING POLICY

- Your work will be graded over a maximum of 100 points.
- There are five items you have to do in this experiment and the contribution of each item to your total score will be partial according to the grading policy stated below

Carrier 1 + comment for algorithmic approach	10p
Carrier 2 + comment for algorithmic approach	20p
Carrier 3 + comment for algorithmic approach	25p
Carrier 4 + comment for algorithmic approach	25p
Code design, clean and readable code, input file reading (reading file with startup parameter is mandatory), comments for other codes	20p

NOTES AND RESTRICTIONS:

- We suggest the Eclipse platform for development
- Your experiment will be tested on a Windows PC equipped with the Eclipse platform. Pay attention not to crash during the first phase.
- Your experiment should be submitted before the due date via the “submit system”. Late submissions will be penalized according to the previously announced rules.
- All assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out abstractly. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudo-code) will not be tolerated.
- In short, turning in someone else’s work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.
- Use EXPLANATORY COMMENTS in your source codes.
- CAUTION: Do not start thinking about the assignment lately. The experiment requires less code but a more algorithmic approach.
- You must send only one file, namely “HW5.java”. Nothing more is required!
- You will submit your work from https://submit.cs.hacettepe.edu.tr with the file hierarchy as below:

```
-----(All the input files be located at parent folder of “src”.
    |----src
        ----  HW5.java
```