School of Computing
National University of Singapore
CS2010: Data Structures and Algorithms 2
Semester 2, AY 2015/16

Assignment 4 – **Fun with Graphs**
Assigned – March 26, 2016 (Friday)
Due – April 15th (Friday by 11.59pm), 2016 (in-lab assessment April 18th (Monday), 2016)

1) Objective

- Using weighted undirected and directed graphs data structures
- Applying different graph algorithms
- Last fun with Java for CS2010

2) Four Mini-Assignments in One

Assignment 4 is actually four (4) separate mini-assignments, each based on graphs. A brief summary for each assignment is given here, with further details in the following section.

A4.1) [Shortest and Longest Process Chain]

You are given a graph that represents a scheduling of jobs. There may be more than one starting point and ending point in this graph. Your task is to determine the shortest and longest paths in this graph among all the starting points to their associated ending points.

A4.2) [Communication Network Construction and Cost]

You are given a large network graph where the cost of each edge represents the cost of the cables to connected vertices. Your job is to first compute a network that connects all the nodes with the minimum cable cost. In addition, you also need to compute additional costs of this communication network depending on the node type, e.g. end-nodes require a terminator $5, nodes with 5-19 connections require a switch that is $500, and nodes with 20 or more connections require a router that is $1000. You need to compute the entire cost of the communication network.

A4.3) [Vertex Labeling Based on Distance]

You are given set of V vertices (that represent locations) and their distances on a map. The user can select up to four (4) locations to place hospitals, call these H1-H4. Once these four locations are given, your task is to determine for the remaining V-4 vertices which hospital they would use depending on distance. This can be consider a labeling problem, as each vertex will be labeled with its closest hospital (e.g. H1, H2, .., H4). Additional points if you can do this very efficiently.

A4.4) [Minimum-Maximum Height for Tunnels]

You are given a graph of road tunnels with edges which represent height limit of those tunnels. Your task is to find a path between user-specified locations, *v* and *u*, that uses the maximum tunnel heights. You need to report the minimum height along the path.

A4.1) [Shortest and Longest Process Chain] – [25 points]

All files for this problem are in A4/A4.1

You are given a directed graph that represents an ordering of process flow has shown in Figure 1. You can assume that there are no cycles. There may be more than one starting and ending points in this graph. For example, in the Graph in Figure 1, the starting nodes are 8 and 2, ending nodes are 1, 4, 10, and 21.

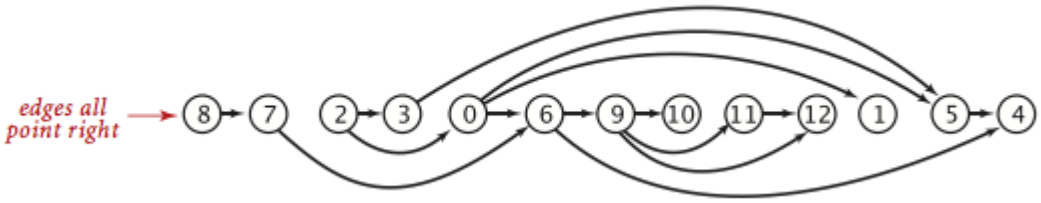

Figure 1: Sample directed acyclic graph.

Your task is to output the shortest and longest path among all the start-end combination.

For example (testcase: a41sampleDAG1.txt)

```
Start points are: 2, 8,
End points are: 1, 4, 10, 12

Shortest process path is (2)
2 to 1 (2):  2->0->1


Longest process path is (5)
2 to 12 (5):  2->0->6->9->11->12
```

or

```
8 to 12 (5):  8->7->6->9->11->12
```

(both answers for longest process path would be considered correct – you only need to find one, but it is OK to print out all if there is a "tie")

A4.2) [Communication Network Construction and Cost] – [25 points]

All files for this problem are in A4/A4.2

In the communication network, the cost is made up of two parts. One cost is from building network cables and the other one is from the equipment for each node. To reduce the budget, you are asked to design a network that can connect all nodes and minimize the total network cables' cost. At the end, you need to give the total cost of the communication network you designed.

2.1 Cost of cable

The cost of network cable is the weight of edges. The graph structure should be decided based on this criteria.

2.2 Cost of equipment for nodes

See figure 2 for the additional cost of nodes on the final network.

End nodes require a small terminator hardware that is $5.

Switch: switches are installed at nodes that have 5 to 19 connections. The cost of one switch is $500.

Router: routers are installed at nodes that have 20 or more connections. The cost of one router is $1000.

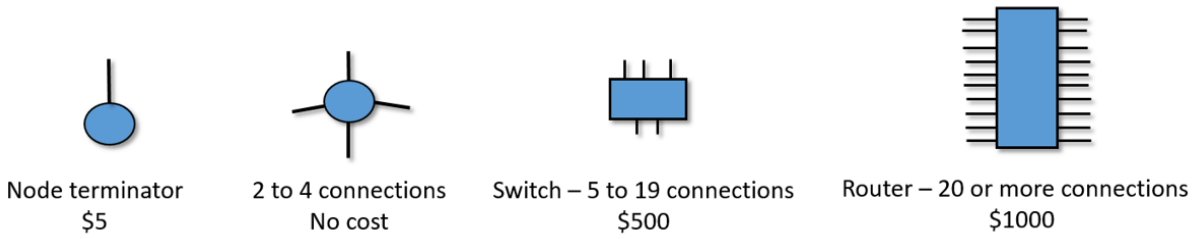

Fig 2. Cost of different types of nodes

2.3 Your Task

Your task is to read a file of graph which consists of all nodes that represent places need to install equipment and cable cost between every two nodes. You then need to compute a communication network which connect all nodes to minimize the cable cost. Finally, you need to compute the total cost of this communication network.

Output the following information:

Cable cost

1) Total number of edges and total cost

2) # of terminal nodes and total cost

3) # of switches and total cost

4) # of routers and total cost

Overall cost: 1+2+3+4

Here is an example, for input A4P4sample.txt, the output is:

```
Number of edges = 30; total cost = $10070
Number of terminal nodes = 27; total cost = $135
Number of switches = 1; total cost = $500
Number of routers = 1; total cost = $1000
Overall cost = $11705
```

A4.3) [Vertex Labeling Based on Distance] – [25 points]

All files for this problem are in A4/A4.3

In a big city there are many hospitals and houses. To save travelling time, residents of each house should always visit the nearest hospital. An ambulance driver would need to know for each house which hospital it is assigned to based on its distance to the nearest hospital.

The city is modelled as V locations (vertices) and E roads (edges). Each location will be either a hospital or a house. Two junctions can be connected by no more than one road. The roads are bidirectional and have only positive weight to describe the length of the road.

A very simple example is shown below of a city with V = 7 and E = 9;

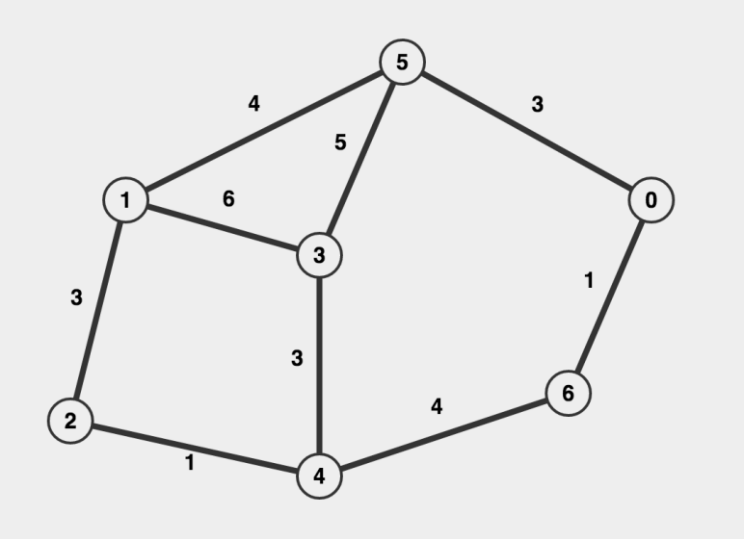

Figure 3. Simple example, edges represent weights.

Your task:

The user can input up to 4 hospital locations.

You then need to output for each hospital location which houses are associated with it and the distance involved. For example, for the figure 3.

```
User input: 0 3

Following 2 houses visit hospital 0:

House 5 visit with shortest distance 3
House 6 visit with shortest distance 1

Following 3 houses visit hospital 3:

House 1 visit with shortest distance 6
House 2 visit with shortest distance 4
House 4 visit with shortest distance 3
```

Note: if the nearest hospital is not unique, assign the house to the nearest hospital with largest index.

To get full mark, attempt to do this as efficiently as possible.

A4.4) [Minimum-Maximum Height for Tunnels] – [25 Points]

All files for this problem are in A4/A4.4

You are given a graph of road tunnels with edges that represent height limit of those tunnels. An example is shown below in Figure 4.

You have a truck whose height is fixed and might not be able to go a certain routes. Your task is to find a route between two nodes that allows the maximum height. However, even along this path, there will still be a minimum that will restrict traffic, so we refer to this problem as the "minimum-maximum height" problem.

Motivation for this problem? Trucks taking paths through tunnels that are not high enough.

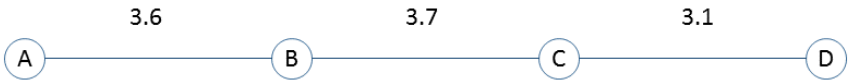

Figure 4: Simple tunnel graph with height restrictions represented as edge weights.

Your task is to read in a graph with V nodes and E edges, where the edges represent the maximum height of a truck allowed on this road. The user can input two nodes (*u* and *v*) and you should provide the path between *u* and *v*, as well as the maximum height allowed (which is the minimum height allowed). See example below in Figure 5:

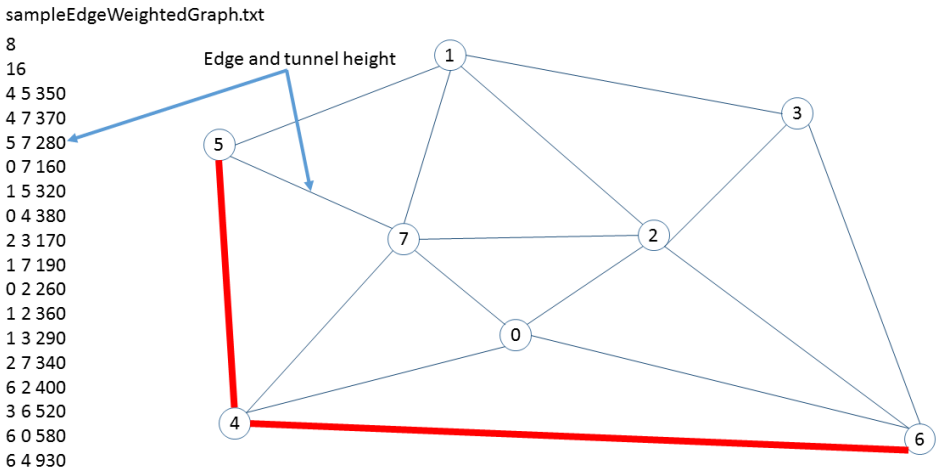

Figure 5: Graph of connected tunnels, with edge weights represent tunnel height limits.

Output:

```
Input:  5 6

Path from 5 to 6:  5-4-6
Maximum Height = 350
```

5) Submission Instruction

What to submit?

Each directory A4.1 - A4.4 has the associate Java file with the main function as well as the data for that mini-assignment. For each mini-assignment, please copy your source code and bin files (class files) and a text-based "README.txt" file that gives a brief descriptions on how you solved the problem in the A4.1-A4.4 directory. The structure should look like this.

A4/A4.1/src/*.java <- All code for your mini-assignment A4.1
A4/A4.1/bin/*.class <- All compiled class files for this mini-assignment
A4/A4.1/readme.txt <- Brief text-based description of how you solved the problem

… (repeat for all mini-assignments)

IMPORTANT: Some additional instructions.

1) Do no create packages! All necessary code should be in the individual directories and added as the default package.

2) Do not import 3rd party APIs, like javafx, etc. Please only use the standard java.* imports.

3) Use the given main file, A4Main1.java, A4Main2.java, A4Main3.java and A4Main4.java.

Submit the following:

1. Submit your code on IVLE before 11.59pm Feb … (late assignments will be deducted -15 points per day).
2. Please put your files in a folder and zip the folder. Use the following convention to name your zipped folder: MatriculationNumber_yourName_Assignment4. For example, if your matriculation number is A1234567B, and your name is Chow Yuen Fatt, for this assignment, your file name should be A1234567B_ChowYuenFatt_Assignment4.zip. As with assignment 2, please name this correctly, points will be deducted for incorrect submission names. We use a script to extract your code for the in-class assessment, in correct file name doesn't work with our script.
3. It is up to you to test to make sure that your Java code in the zipped file is complete. It is recommended that you unzip your Java code in a different location and test it to make sure it runs.

Assessment (April 18th, 2016 – In Lab)

- You will be assessed individually in the lab session (you must attend your assigned lab!)
- You'll have approximately 4-5 minute Q&A session with Dr. Brown or one of the TAs (Hakki, Sixing or Abdel).
- We will run the code directly from your submitted IVLE submission, you won't be able to bring an updated version to the lab.
- We will run your code and then ask you anything about your code we want. Be prepared to answer questions. Please comment your code if you think you will forget by the time we do the assessment.