COMP 202 - Spring 2021 - Homework #4 Part II

Due date: Wednesday, May 26 at 23:59

HW Description:

This is a two-part assignment, in which you are going to implement 2 graph algorithms:

1. Minimum spanning tree (MST)
2. Shortest path

In this part you will have to implement a shortest path algorithm of your choice. Note that you will also be using the MST part of the assignment that you have implemented in the first part of the homework. The shortest path algorithms you are advised to choose from are:

1. Dijkstra's algorithm
2. Floyd–Warshall algorithm

In this assignment, you will solve several real world network planning problems by mapping them to well-known graph algorithms mentioned above. Using the network planning problem gives you the opportunity to implement new data structures, such as union-find, and apply previously used data structures, such as heaps/priority queues, on real-world problems. Additionally, this assignment comes with more freedom than previous ones, meaning you can choose whichever data-structures you feel appropriate for solving the problem presented.

The tasks to solve:

You will be given a text input in form given below (the headers won't be included, the columns will be separated by spaces)

Network Point A	Network Point B	Cost (millions of dollars)	Latency (milliseconds)
ibm.com	yahoo.com	20	33
yahoo.com	ucsd.edu	19	40
ucsd.edu	ibm.com	10	17
yahoo.com	fsf.org	5	10

Parsing the input will be done for you, so you can start working on an adjacency matrix representation of a graph. You are free to convert it further to any other form you feel more comfortable to use (like adjacency list or edge list). You are free to add additional methods to both HW4.java and Graph.java files. You are free to change the input parameters/arguments to task methods.

Given the inputs, your tasks are to output the following:

1. The total transit time to send a packet between all pairs of computers if all possible network links were built. This requires running Dijkstra's algorithm to find the shortest path from every node in the graph to every other node, or running the Floyd-Warshall algorithm. For example, you would have to identify the fastest way to get from ibm.com to yahoo.com, ibm.com to ucsd.edu, ibm.com to fsf.org, yahoo.com to ibm.com, yahoo.com to ucsd.edu, etc. Note, this is solved using the all-possible-links graph, not the MST from HW4 part I task 2. Moreover, note that this time you are working with latencies.
   No matter what algorithm you choose, in the complexity analysis compare two algorithms in terms of time and space complexity.

2. The total transit time it would take to send a packet between all pairs of computers in the minimum-cost network. This is the same as above, except on the graph generated by the MST method from HW4 part I task 2.

3. The increase in the "total time" required for packet travel in the cheap network (compared to having built all possible links). This is the difference between your solution to line five and line four.

For example, given the input above, the output would be:

```
386
482
96
```

Note that despite the graph being undirected, when the total transit time is calculated you need to add both the transit time from point A to point B and the transit time from B to A, even though they are identical.

Submission Materials:
In all homeworks, you are required to submit two things:

- A pdf file with your pseudocode and asymptotic complexity analysis (time and space). If it is handwritten, make sure it is legible.
- A java implementation that compiles and runs without additional setup (submitted through github classroom **and** via Blackboard).

In your pdf file, you must copy-paste the following statements exactly, and sign it by hand. You can use your phone camera for that. Without this signed statement, your submission will not receive any grade.

I have completed this assignment individually, without support from anyone else. I hereby accept that only the below listed sources are approved to be used during this assignment:

(i) Course textbook,

(ii) All material that is made available to me by the professor (e.g., via Blackboard for this course, course website, email from professor / TA),

(iii) Notes taken by me during lectures.

I have not used, accessed or taken any unpermitted information from any other source. Hence, all effort belongs to me.

Github Submission:
Usage of github classroom and submission details:

1) Accept the invitation using this link (https://classroom.github.com/a/mN4dw9Kb)
2) Choose your student ID from the list, it will be linked to your github account from now on.
3) A personal repository with the starter code will be created. You can directly edit the code on github's page, clone it to your local storage and edit it there, or use the online IDE Repl.it.
4) Make sure your changes are committed and pushed to the repository.
5) Check if you have successfully passed the tests in the "Actions" tab of your repository. Note that these automatic tests help you understand whether or not your solution works up to some level, but passing them does not guarantee that you will receive a full grade.
6) To make sure your code is received and avoid any potential problems on github's side, submit a copy of java files together with the above-mentioned pdf on Blackboard as well. Submit all files as a single .zip file, named as: "ID_NAME_SURNAME.zip"