

GTU Department of Computer Engineering
CSE 222/505 - Spring 2016
Homework 09

Due date: May 23 2016 – 08:00 AM

Part 1:

1. Use Graph Generation Program (which is uploaded to the system) to generate 20 test graphs with 10, 50, 100, 500 and 1000 vertices and 0.2, 0.5, 0.75, 0.8 sparsity values. The range of the edge weight must be between 1 and 1000. You can visualize the generated graph in the program by clicking “Graph > Show” (visualization of graphs with high numbered vertices can take long time). You can export your graph to a file by using File menu. You may need to convert the file to the format that is used out textbook.

Run shortest path and MST algorithms for the graphs you generated. Find the shortest paths from a source vertex to all other vertices for each graph using the Dijkstra's algorithm. Find the minimum spanning tree using Prim's algorithm. Test the Dijkstra's and Prim's algorithm implementations according to the time and memory complexities by using

- a) the graph ADT implemented using adjacency list
- b) the graph ADT implemented using adjacency matrix.

Note: You should use the implementation of the Dijkstra's algorithm in Listing 10.7 and the implementation of Prim's algorithm in Listing 10.8.

2. Write a method to find the sizes (number of vertices) of the largest (having maximum number of vertices) and smallest (having minimum number of vertices) connected components in a graph passed as a parameter. Note that if the graph is connected both the largest and the smallest connected components will be the same; the whole graph. Write a test method that reads a graph from a file and prints the results. You should test your method by generating several small size disconnected graphs.

RESTRICTIONS:

- Use maven standard Project template
- Can be only one main class in project
- Don't use any other third part library.

GENERAL RULES:

- For any question firstly use course news forum in moodle, and then the contact TA.
- Use maven project management tool. And upload maven project into moodle.
- Code the Project in Java programming language. Java must be 1.8.* or bigger version.
- Any java IDE can be used in coding process.
- Implement all interfaces class
- Add all javadoc documentations for classes, methods, variables ...etc. All explanation must be meaningful and understandable.
- Implement clean code standards in your code;

- o Classes, methods and variables names must be meaningful and related with the functionality.
- o Your functions and classes must be simple, general, reusable and focus on one topic.
- o Use standart java code name conventions.
- Register github student pack and create private project and upload your projects into github.
- Your appeals are considered over your github project process.
- You can submitting assignment one day late and will be evaluated over forty percent (%40).
- Create report which include;
 - o Your name, surname, studentid
 - o Detailed system requirements
- o The Project usecase diagrams (extra points)
- o Class diagrams
- o Problem solutions approach
- o Test cases
- o Running command and results