Assignment 5

Due date: 10:00am, Wednesday, the 12th of October 2016.
This assignment is worth of 7% of the total course mark.

General Instructions

You have to do it **individually**.

Paper submissions have to include coversheet including names, student ids, and your tutorial groups such that submissions can get marked.

For Programming submissions upload your files in websubmission **by the deadline**. **No late submissions will be accepted.**
Names and IDs of group members must be specified in a comment in the beginning of all files that you submit.
Hand in your solutions to the box "ADSA" on level 4, Ingkarni Wardli (close to reception) **by the deadline**. **No late submissions will be accepted.**

Exercise 1 *(40 points) Programming Exercises*

For all of the following exercises, write a program, test and and submit the file in the web submission system. Your program should have a main function and inputs such as the path of the graph file are are passed as arguments for the main function.

1. Implement a graph traversal algorithm (in a file called Acyclic.java) that takes a directed graph $G = (V, E)$ as an adjacency matrix, where $G_{i,j} \in \mathtt{Z}$ denotes an existing edge from i (in the ith row) to j (in the jth column) with weight $G_{i,j}$ if it is not equal to 0. Otherwise, if the weight is 0, there is no edge from i to j. You program should find out whether the input graph is acyclic. As the second output, the program should print 0 if the graph does NOT have cycles and it should print 1 otherwise. *[20 marks]*

2. Write another program that takes a graph as above and finds all of the strongly connected components (SCC). Your program should print the size of each SCC separated by a comma. *[20 marks]*

Exercise 2 *Floyd-Warshall Algorithm (20 points)*

1. Implement a program (save in file FloydWarshallLongestSP.java) that uses the Floyd-Warshall algorithm to compute the longest path of all-pairs-shortest paths for a given graph. You can use your implementation of graphs from the previous questions. *[20 marks]*

Paper Based Questions

Exercise 3 *Shortest Path Algorithms (25 points)*

1. Comment out the post-processing part of the Bellman-Ford algorithm, run the code again and see what happens when you use the same test cases above. In your paper-based answer sheet, explain what type of inputs may cause problem, what happens in this case and why. *[10 marks]*

2. You are developing a packet routing program for a network. To select the next server to forward a packet, you would like to choose a router that has the maximum number of shortest paths from the current router (so that you reduce the risk of the packet getting lost). Suppose you are given a network map with distance between each pair of directly connected routers. Design an $O(V + E)$-time algorithm that finds the number of shortest paths between the current router (the source vertex s) and the next router (the target vertex t). *[10 marks]*

3. Use a proper algorithm to find all pairs shortest paths (APSP) for the following graph and write down the final result. *[5 marks]*

$$\begin{bmatrix} 0 & 1 & 3 & 0 & 0 & 3 \\ 3 & 0 & 5 & 1 & 2 & 2 \\ 1 & -1 & 0 & 0 & 2 & 1 \\ 1 & 0 & 1 & 0 & 5 & 0 \\ 0 & 3 & 1 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 & 2 & 0 \end{bmatrix}$$

Exercise 4 *True or False (15 points)*

Is the following statement true or false? Prove or reject by example.

1. Let P be a shortest path from some vertex s to some other vertex t in a directed graph. If the weight of each edge in the graph is increased by one, P will still be a shortest path from s to t. *[5 marks]*

2. Running Dijkstra on a graph always gives correct distances. *[5 marks]*

3. If all edges in a graph have distinct weights, then the length of the shortest path between two vertices is unique. *[5 marks]*

Instructions for programming code

First, type the following command, all on one line (replacing `aXXXXXXX` with your username):

```
svn mkdir --parents -m "ADSA"
https://version-control.adelaide.edu.au/svn/aXXXXXXX/2016/s2/adsa/assignment5
```

Then, check out this directory and add your files:

```
svn co https://version-control.adelaide.edu.au/svn/aXXXXXXX/2016/s2/adsa/assignment5
cd assignment5
```

```
svn add File1.java
add File2.java ...
svn commit -m "assignment5 solution"
```

Next, go to the web submission system at:
https://cs.adelaide.edu.au/services/websubmission/
Navigate to 2016, Semester 2, Adelaide, Algorithm and Data Structure Analysis, then Assignment 5. Click Make A New Submission For This Assignment and indicate that you agree to the declaration. The script will then check whether your code compiles. You can make as many resubmissions as you like. If your final solution does not compile you would not get any marks for this solution.

End of Questions