Ann, Bob, Charlie (*replace with your names*)
COSC 336
3/19/2020 (*replace with the current date*)

Assignment 8

Instructions.

1. Due date announced on Blackboard.

2. This is a team assignment. Work in teams as in the previous assignments. Submit one assignment per team, with the names of all students making the team.

3. Your programs must be written in Java.

4. Write your programs neatly - imagine yourself grading your program and see if it is easy to read and understand. At the very beginning present your algorithm in plain English or in pseudo-code (or both). Comment your programs reasonably: there is no need to comment lines like "i++" but do include brief comments describing the main purpose of a specific block of lines.

5. You will submit on Blackboard 2 files. The first file should be a .pdf file with the solutions of the Exercise and a short description in English or in pseudocode of the algorithm for the programming task you are required to do and the results that you are required to report. Also include in this file, the Java code so that the grader can make comments.

   Files 2 will contain the java code for the programming task.

   For editing the above document with Latex, see the template posted on the course website.

   assignment-template.tex and

   assignment-template.pdf

   To append in the latex file a pdf file, place it in the same folder and then include them in the latex file with

   ```
   \includepdf[pages=-,pagecommand={},width=\textwidth]{file.pdf}
   ```

   To append in the latex file a .jpg file (for a photo), use

   ```
   \includegraphics[width=\linewidth]{file.jpg}
   ```

Exercise 1.

We have seen that Dijkstra's algorithm can be implemented in two ways: Variant (a) uses an array to store the $dist[]$ values of the unknown nodes, and Variant (b) uses a MIN-HEAP to store these values.

(a) Suppose in your application $m \leq 3n$. Which variant gives a faster runtime? Justify your answer.

(b) Suppose in your application $m \geq n^2/3$. Which variant gives a faster runtime? Justify your answer.

(c) Suppose that your application $m = n^{3/2}$. Which variant gives a faster runtime? Justify your answer.

Exercise 2. Recall that when we do DFS with timing every node u gets 2 numbers that were denoted $u.d$ and $u.f$. $u.d$ is the discovery time and $u.f$ is the finish time.

Show that in a DAG (directed acyclic graph), for any two nodes u and v such that there exists a path from u to v, it holds that $u.f > v.f$.

Hint: There are two cases to analyze. Case 1 is that $u.d < v.d$ (in words, u is discovered before v), and Case 2 is that $v.d < u.d$ (so, v is discovered before u). In both cases, you need to argue that $u.f > v.f$.

Programming Task

Write the program that modifies Breadth First Search (see for example the basic version of BFS in Notes 10) in such a way that given an undirected connected graph G, and a starting node s, it will print for every node v the length of the shortest path from s to v and also the number of shortest paths from s to v. Thus, you'll have two arrays $dist$ and $npath$, and for each vertex v, at the end of the program, $dist[v]$ will be equal to the length of a shortest path from s to v, and $npath[v]$ will be equal to the number of shortest paths from s to v.

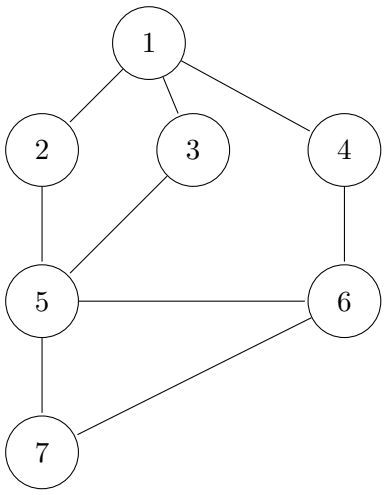
For example, if G_1 is the first graph below, the length of a shortest path from 1 to 7 is 3, and there are three shortest paths from 1 to 7, namely

1 - 2 - 5 - 7,
1 - 3 - 5 - 7 and
1 - 4 - 6 -7).
So, $dist[7] = 3$, and $npath[7] = 3$.

Test your program on the graphs G_1 and G_2 (see the figures) using 1 as the starting node. Your program is required to print the two arrays $dist[]$ and $npath[]$. Report the results you have obtained for these two graphs.

You must use the adjacency list representation of a graph. As in the programming task from assignment 7, for the adjacency list, you **must** use the Java class Adj_List_Graph given in the file Adj_List_Graph.java (see Test_Adj.java for a very simple example of using this class).

graph G_1



graph G_2: