Notes:

\* You are required to upload your in–class implementations of problems 1-2, to canvas. This is due by 9:50 AM today.

\* You are required to turn in a written report (Word or PDF file) for the homework part (problem 3) of the lab and upload implementations to canvas. These are due by 8:00 AM, November 15, 2017).

Objectives:

\* Implement Depth First Search algorithm on graphs.

\* Implement Topological Sort

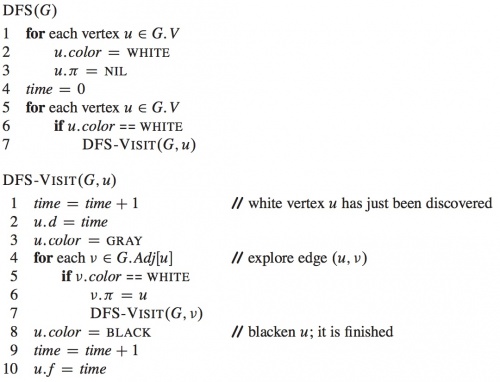
Problems:

1. Write a class DepthFirstPaths.java and implement depth-first search algorithm using the pseudocode given on the next page. You can alternatively download the file DepthFirstPaths.java and implement all the unimplemented methods of the class so that it performs depth first search on graph G.

2. Write a driver program, which reads input file mediumG.txt as an undirected graph and runs the Depth First Search Algorithm to find paths to all the other vertices considering 0 as the source. This driver program should display the paths in the following manner:

0 to ‘v’: “list of all the vertices traversed to go to v from 0, separated by ‘,’ ”. There is another file called tinyG.txt which you can use for your testing purpose.

3. Implement the Topological Sort algorithm for directed graphs and run the algorithm on the directed graph that is provided in the tinyDG.txt file. The pseudocode is provided in the next page.



TOPOLOGICAL-SORT(*G*)

1 call DFS(*G)* to compute finishing times *v.f* for each vertex *v*

2 as each vertex is finished, insert it onto the front of a linked list

3 **return** the linked list of vertices