CS2383 – Fall 2024
Assignment 8 – Graphs and related algorithms

Due: Thursday Dec. 5, 10am (class time)

NO LATE ASSIGNMENT ACCEPTED THIS TIME, as I may discuss the solution in class on that day, in preparation for the final exam.

IMPORTANT: individual work please!

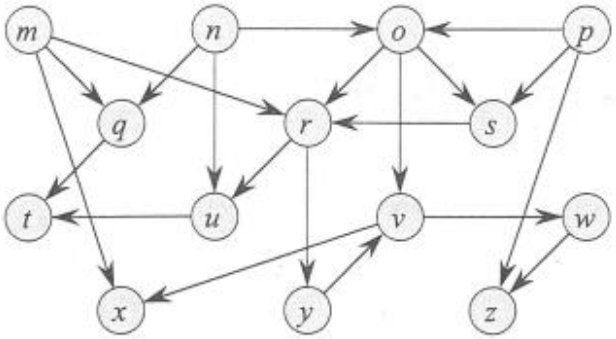
Tasks:

1. Download the 2 Java files located in the Asgn8 folder on D2L. Both are the basic implementations of an undirected graph, one using adjacency lists and one using an adjacency matrix. Make the necessary modifications to make them implement directed graphs instead. Change the names of the classes to DiGraphAdjListImplem and DiGraphAdjMatrixImplem.

2. In the code you had in Question #1 above, add the following method in each of the classes: "public Graph transpose()" where "Graph" is replaced by the exact class name (i.e., DiGraphAdjListImplem and DiGraphAdjMatrixImplem). This method should create a new graph object (and return it) containing the same vertices as the original graph, but with all edges reversed. For example, if there is an edge from vertex 2 to vertex 4, then the new graph should have the edge from vertex 4 to vertex 2 instead. Use a main method to test your work, using the toString method provided. What is the runtime of your algorithms?

3. There are two types of professional wrestlers: "babyfaces" ("good guys") and "heels" ("bad guys"). Between any pair of professional wrestlers, there may or may not be a rivalry. Suppose we have n professional wrestlers and we have a list of r pairs of wrestlers for which there are rivalries. Give an O(n+r)-time algorithm that determines whether it is possible to designate some of the wrestlers as babyfaces and the remainder as heels such that each rivalry is between a babyface and a heel. Note: just provide the idea of how to solve this problem. No need to provide the code for it.

4. Show how the topological sort would work on the example graph below (see next page). Consider the vertices in alphabetical order, and assume that each adjacency list is ordered alphabetically. Although normally the graph would first be turned into one using numbers from 0 to n-1, together with a symbol table to keep the mapping, for this assignment work directly with the graph as is.



Submission: submit everything on paper, including a printout of all your code. Also upload your Java files in D2L: those after Question #2 is completed. You do not need to submit the files with the main methods that you used for testing: we will run the code with our own main files. Therefore, it is important to keep the exact same class names and method names as the ones specified above.