CS 211 Data Structures and Algorithms Lab
Autumn 2022
Examination

Implement Depth First Search (DFS) and to use it to do topological sorting of a directed acyclic graph (DAG)	
Total marks	*5*
Due date (without penalty)	*11th November (Friday) 5.00pm*
Cut-off date (with penalty - 20%)	*12th November (Saturday) 5.00 pm*
Penalty for violating naming convention(s)	*10%*

Note: Please run the code in linux platform and with gcc compiler. You may use Google Colab before submitting.

Command-line argument:
Your program should receive a file (input file) as a command line argument.

Input:
Your program should accept an input file as a command-line argument. A typical execution of your program will be ***./a.out sample.graph***.

- The input file represents a *directed acyclic graph (DAG)*.
- The first line of the file contains two numbers - first being the number of *vertices n* and the second being the number of *edges m*.
- The vertices are numbered from *0 to n-1*.
- Every other line is of the form x y, which represents a directed edge from vertex x to vertex y.
- No edge is repeated in the input file.
- Your program may create an adjacency list of the input graph which can be used for the task.

Task:
Implement DFS and use it to do a *topological sorting* of the input DAG.

Output:
The output file should be named as ***'ts.txt'***.

- The output file must contain the vertices - one vertex per line - which represents a topological sorting of the input DAG.

- Please note that there can be many topological sortings of the same DAG depending on the order in which the DFS is done. *You can do DFS in any order that you wish and a valid topological sorting is enough to receive full marks*.

Evaluation:

There are two parts for the evaluation:
(i) ***part 1 (max: 1 mark):*** *you either receive 0 mark or 1 mark. To receive 1 marks you have to make sure that the topological sorting produced by your algorithms contains exactly the same set of vertices in the DAG;*
[Note: The given input.graph might also contain Isolated vertices, which means that this type of vertices do not contain any in-coming or out-going edges. These vertices are not present in the input.graph file, but these vertices should be present in the output file ***ts.txt***]

(ii) part 2 (max: 4 marks),*: The marks you obtain will be proportional to the number of edges that is being respected by the sequence of vertices that your program gives as a topological sorting. A sequence of vertices respects a directed edge xy if x comes before y in the sequence.*

We will not be using 'diff' for the evaluation.
[Note: As given *input.graph* might produce different possible topological orders. Thus, the provided output file for testing is one of the possible topological order, which may not be the same as you obtain.]

Submission:
- The program you submit should output ***'ts.txt'*** when run.
- The main file of your program should be named as <roll no>.<extension>, where roll no. specifies your roll no. and the extension depends on the language you choose (Usage of C is mandatory for this assignment). Ex: 210010001.c.
- Do the stress test of your program well before submission.
  (i) You may use the attached sample input files for testing; the corresponding output files are also attached;
- If your program has only a single source file, please submit the file as it is. If your program has multiple source files, please submit your code as a zip file where the name of the zip file should be your roll number. It is important that you follow the ***input/output conventions*** exactly (including the naming scheme) as we may be doing an automated evaluation. ***There will be a penalty of 10% (on the mark you deserve otherwise) if you do not follow the naming conventions exactly.***
- Follow some coding style uniformly. Provide proper comments in your code.
- *Submit only through moodle*. ***Submit well in advance***. Any hiccups in the moodle at the last minute is never acceptable as an excuse for late submission. Submissions through email or any other means will be ignored.
- ***Copying others' programs is a serious offense and a deserving penalty will be imposed if found.***

- If you submit after the due date but on or before the cut-off date, there will be a penalty of 20% on the marks you deserve otherwise.
- ***No Second Evaluation for Exam.***