

CS 211 Data Structures and Algorithms Lab

July -- December, 2018

Assignment 6

Total Marks: 10

Due on 19th October

The objective of this assignment is to implement Breadth First Search (BFS) to solve single-source shortest path problem on directed graphs and to implement Depth First Search (DFS) to do topological sorting of a Directed Acyclic Graph (DAG).

Inputs

Your program should accept an input file and an integer as command-line arguments. A typical execution of your program will be `./a.out sample.graph 25`

The input file represents a directed graph. Every node (vertex) in the graph is uniquely labelled with a non-negative integer. Every line in the input file is of the form `x y`, which represents a directed edge from node `x` to node `y`. No directed edge is repeated in the input file. Since topological sorting makes sense only for DAGs, it is guaranteed that the input graph is a DAG.

The second command-line argument is an integer, which represents the label of a vertex in the given graph, which is the source from which the shortest distance of every vertex has to be calculated using BFS. Note that this argument is irrelevant for doing topological sorting and can be ignored for the same.

Your program may create an adjacency list of the input graph which can be used for both Task 1 and Task 2.

Task 1 (5 Marks)

Implement BFS and use it to find the shortest distance of every vertex from the source vertex (second command-line argument). The output file should be named as `'sd.txt'`. Every line in the output file should be of the form `<vertex-label> <shortest-distance-from-source>`. For example, if the shortest distance of vertex with label 35 from the source vertex is 10, then there is a line in the output file which has `'35 10'`. The vertex-distance pair in the output file can be ordered in any fashion. If there is no directed path from the source to a vertex, then the corresponding distance should be denoted by `-1`.

Task 2 (5 Marks)

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

Submission

- The program you submit should output 'sd.txt' and '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/C++ is mandatory for this assignment). Ex: 180040001.c
- Test well before submission. You may use the attached sample input file for testing. The corresponding output file is also attached. We have some hidden inputs with us to test your program. 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/internet at the last minute is never acceptable as an excuse for late submission. Submissions through email or any other means will be ignored.
- Acknowledge the people (other than the instructor and TA) who helped you to solve this assignment. The details of the help you received and the names of the people who helped you (including internet sources, if applicable) should come in the beginning of the main file as a comment. Copying others' programs and allowing others to copy your program are serious offences and deserving penalty will be imposed if found.

Evaluation

- To consider for first evaluation without penalty, you have to submit your program by **19th October (11:59 pm)**. If you submit after **19th October** but on or before **26th October (11:59 pm)**, there will be a penalty of 10% on the marks you deserve otherwise.
- If you do not submit by **26th October**, your program will not be considered for the first evaluation.
- We will do the first evaluation after **26th October**. **Unlike the previous assignments, the evaluation is not based on diff between the correct output and the output produced by your program. For the task 1, the marks you get is proportional to the number of correct vertex-distance pairs. For the task 2, the marks you get is proportional to the number of edges that are in accordance with the sequence of vertices your program gives. An edge (u,v) is in accordance with the sequence your program gives if u comes before v in the sequence. Note that there can be more than one topological sort for a DAG.**