HW-3: CSC241 (Data Structure and Algorithms)

Due Date: 03/29/2023 (11.59:59 PM)

Special Instruction

Your HW will be graded based on correctness and clarity. Keep you answer precise and to the point. If any question ask for justification of your answer/claim, you may receive a 0 if you merely provide an answer without justification. All sub questions carry equal weights unless specified otherwise. Finally, please check the HW rules at the end.

What to do: Reach to me ASAP if you have any confusion and/or have any emergency that may deter you to submit HW on time. Never hesitate to ask me if any of the previously discussed topics is unclear and you need some more discussion.

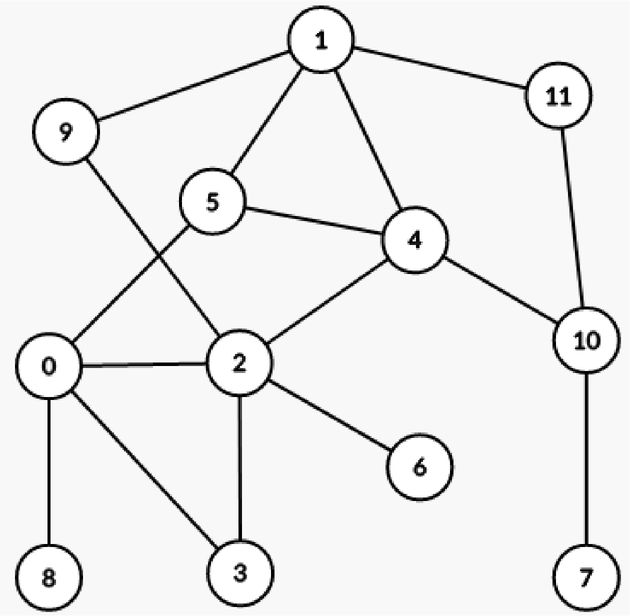
What not to do:

1. Ask to verify your solution
2. Ask to debug/analyze your code

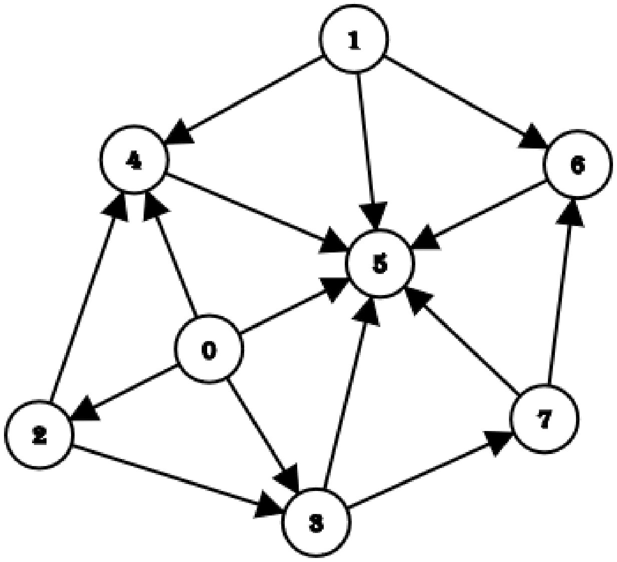
Question 1 (30 Points): Consider the above mentioned graph. Now,

I. Apply BFS considering 1 as the source node (level-0 node). Draw the BFS tree and mark each level. If there are multiple options at any step, pick the node with smaller index.

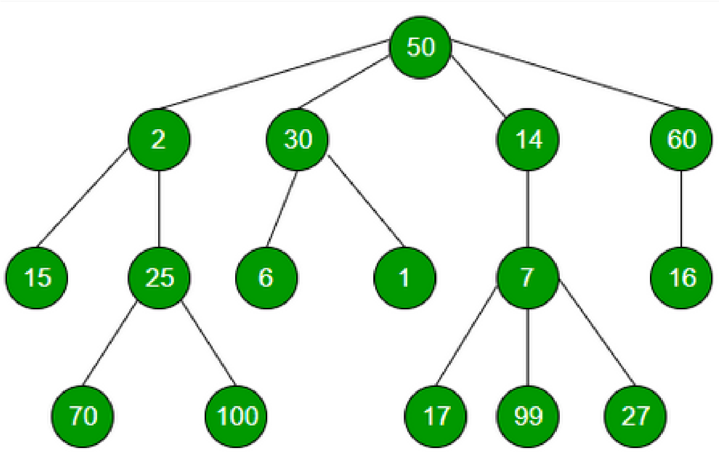
II. Repeat part II considering 6 as the source node (level-0 node).



Question 2 (20 Points): Consider the above mentioned graph with many job dependencies. Each node denotes a job and x -> y means job y is dependent on job x. Determine a feasible schedule by applying topological sort. If there are multiple options at any step, pick the job with smaller index.



Question 3 (20 points): Consider the following tree. All the nodes under the same parent are siblings. For example, 70 and 100 are siblings (25 is their parent). Is it possible to determine the siblings for any node using a queue? Why or why not? Describe you logic and provide a pseudocode with adequate comments.



Special Note: Whatever your answer is, don't describe it using just plain text. Take a node as an example, and describe your answer step-by-step. Make sure your pseudocode contains adequate comments which describe each steps.

Question 4 (30 points): Consider the same graph mentioned in Question 1. Now,

I. Apply DFS considering 1 as the source node. If there are multiple options at any step, pick the node with smaller index. Draw the DFS tree. Also, mention all the back-edges.

II. Repeat part II considering 6 as the source node.

Rules for ALL HW:

1. If any programming problem is given, the code must be written by yourself. DO NOT copy code from anywhere else.

2. You can discuss the problem sets and study together in group, but when it comes to formulating/writing solutions you must work alone independently; i.e., you should be able to explain your answer clearly to anyone else (including the TA and the instructor). Note that this says discuss in group — copying homework solutions from another student, from the Internet, solution sets of friends who have taken this course or one similar to it previously, or other sources will be considered **cheating** and referred to the university. *At the beginning of each submission, you should explicitly list the people you worked with.*