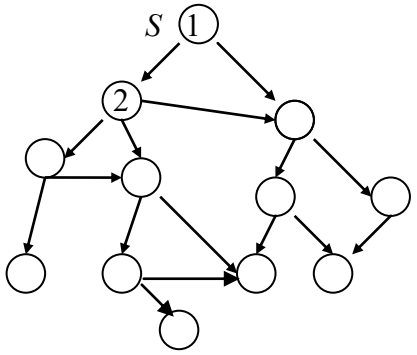
Design and Analysis of Algorithms
CS 575, Spring 2024

Theory Assignment 3.2

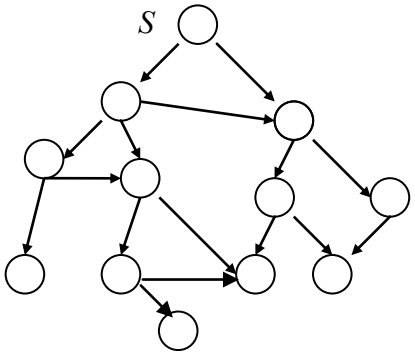
Due on 4/16/24 (Tuesday) at 11:59pm

Remember to include the following statement at the start of your answers with a signature by the side. "I have done this assignment completely on my own. I have not copied it, nor have I given my solution to anyone else. I understand that if I am involved in plagiarism or cheating I will have to sign an official form that I have cheated and that this form will be stored in my official university record. I also understand that I will receive a grade of 0 for the involved assignment for my first offense and that I will receive a grade of "F" for the course for any additional offense.

1. [20 points] In this problem, you will apply different traversal methods on a given directed graph. The start node is denoted by S. When arbitrary decisions on order must be made assume that child nodes are visited from left to right.
   a. [10 points] Perform a breadth-first search on the directed graph below (on the left). (i) *Number the nodes* according to the order in which they are visited (become gray). For example, the first node visited is *S* so *S* is numbered 1. Then, the left child of *S* is visited so it is numbered 2. Show the order numbers inside the circles. (ii) Show the distance of each node to *S* beside each circle. (iii) Show the breadth-first tree.



   b. [10 points] Perform a depth-first search on the directed graph below (on the left). (i) Number the nodes according to the order in which they are visited at the first time (when they become gray). Show the order numbers inside the circles. (ii) Show the depth-first tree(s).



2. [15 points] Please modify the depth first search algorithm (slide 36 and 37 of the graphs basics lecture notes) to find all connected components in an undirected graph. Comment on where you made the modification. Your modified algorithm needs to print out each component ID (starting from 1) and the corresponding vertices. For example, your output for the DFS example (slide 38-41 of the graphs basics lecture notes) will look like the following: Component 1: u, v, y, x. Component 2: w, z.