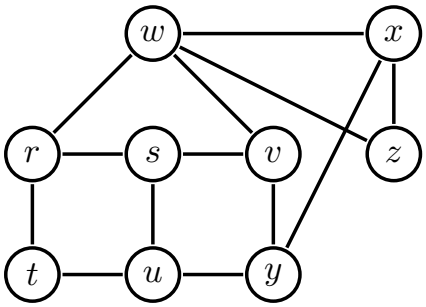
CS 624: Analysis of Algorithms

Assignment 5

Due: Saturday, April 25, 2026

1. Show the d and π values that result from running breadth-first search on the undirected graph in the figure, using vertex u as the source.



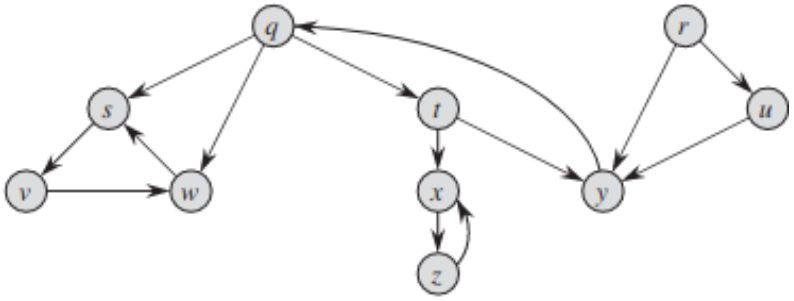
2. There are two types of professional wrestlers: "babyfaces" ("good guys") and "heels" ("bad guys"). Between any pair of professional wrestlers, there may or may not be a rivalry. Suppose we have n professional wrestlers and we have a list of r pairs of wrestlers for which there are rivalries. Give 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. If it is possible to perform such a designation, your algorithm should produce it. You may re-use any algorithm we learned in class as-is (without re-writing all the stages).

3. Show by induction that the number of degree-2 nodes in any nonempty binary tree is 1 fewer than the number of leaves. Conclude that the number of internal nodes in a full binary tree is 1 fewer than the number of leaves. Be sure to carefully state the inductive hypothesis. It will help you in constructing the proof.

4. A directed graph G = (V, E) is said to be semi-connected if, for all pairs of vertices $u, v \in V$ we have $u \rightsquigarrow v$ or $v \rightsquigarrow u$ path (or both).

   Give an efficient algorithm to determine whether or not any directed acyclic graph (DAG) $G = (V, E)$ is semi-connected

5. Show how depth-first search works on the graph below. Assume that the for loop of lines 5–7 of the DFS procedure considers the vertices in alphabetical order, and assume that each adjacency list is ordered alphabetically. Show the discovery and finishing times for each vertex, and show the classification of each edge.



6. Argue that it is not necessary to color the nodes BLACK in DFS. In other words - that it will produce the same results if line 10 in DFS-Visit is removed.

7. Professor Bacon wants to rewrite the strongly connected components algorithm and use the original graph (rather than the transpose) in the second DFS run and scan the vertices in *increasing* finish time rather than decreasing. Does this modified algorithm always produce the correct results?