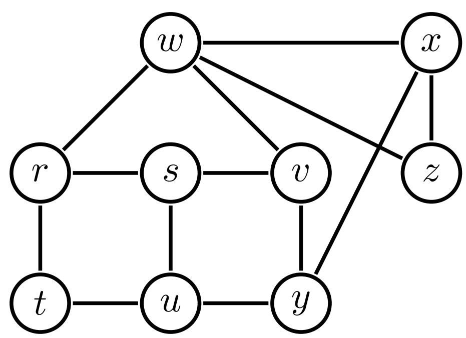
CS 624: Analysis of Algorithms

Assignment 5

Due: Nov. 13, 2023

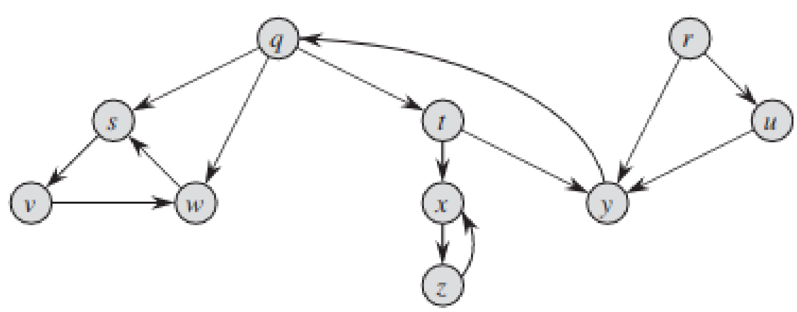
1. Show the d and π values that result from running breadth-first search on the undirected graph in the figure, using vertex u as the source.



2. Give an example of a directed graph $G = (V, E)$, a source vertex $s \in V$, and a set of tree edges $E_\pi \in E$ such that for each vertex $v \in V$, the unique simple path in the graph (V, E_π) from s to v is a shortest path in G, yet the set of edges E_π cannot be produced by running BFS on G, no matter how the vertices are ordered in each adjacency list.

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. 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.



5. Give a counterexample to the conjecture that if a directed graph G contains a path from u to v, and if $u.d < v : d$ in a depth-first search of G, then v is a descendant of u in the depth-first forest produced. The notation u.d in that problem refers to the "discovery time" or (as it is called in my notes) the "start time" for vertex u in the depth-first walk.

6. Give a counterexample to the conjecture that if a directed graph G contains a path from u to v, then any depth-first search must result in $v.d < u.f$.

7. 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

8. 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?