MCS 401: Computer Algorithms I (Fall 2020)
Homework 2

Due at 2:00pm CDT, Monday, Sep 28

1. (2 points) Consider the following (undirected unweighted) graph $G = (V, E)$ given in the adjacency list representation:

$$\texttt{Adj}[1] = [2, 5]$$
$$\texttt{Adj}[2] = [1, 3]$$
$$\texttt{Adj}[3] = [2, 4]$$
$$\texttt{Adj}[4] = [3, 5]$$
$$\texttt{Adj}[5] = [1, 4]$$

   (a) Draw the graph G. Give the adjacency matrix representation of G.

   (b) Draw the BFS tree and DFS tree for G, starting at node 1. (Assume that when exploring a node, the algorithm iterates over its neighbors from smaller index to larger index.)

2. Let $G = (V, E)$ be a connected (undirected) graph. Fix a vertex $s \in V$. Suppose we run BFS and DFS on G starting at node s. Let T_B and T_D denote the resulting BFS tree and DFS tree respectively. Prove that if $T_B = T_D$, then G must be a tree.

   (Hint: What happens if G has a cycle?)

3. (2 points) Prove that for every n-node tree T, there exists a node u such that each connected component of $T - u$ (removing u and all its edges in T) has at most $n/2$ nodes.

   (You may want to use the sizes of subtrees in your proof. In a tree T, a *subtree* rooted at x consists of x and all of its descendants. Hint: If we remove a node $u \in T$, what is the size of each connected component in the remaining graph?)

4. Consider the following directed (acyclic) graph. How many topological orderings does it have?