Graph Algorithms

Due: Wednesday, August 9th

Directions: Some of the questions on this assignment will appear on the quiz on Wednesday, August 9th.

1. Perform the DFS algorithm on the following graph. Give the pre- and postorder numbers for each vertex.

2. Perform the DFS algorithm on the following graph. Label each edge as a tree edge, forward edge, back edge or cross edge.

3. A *bipartite graph* is a graph $G = (V, E)$ such that V can be partitioned into two sets ($V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$) such that there are no edges between vertices in the same set.
   (a) Design and analyze an algorithm that determines whether an undirected graph is bipartite.
   (b) Prove the following theorem:
   An undirected graph, G, is bipartite if and only if it contains no cycles of odd length.

4. There are three containers: a 10 liter container, a 7 liter container, and a 4 liter container. The 7 and 4 liter containers are initially full of water, while the 10 liter container is empty. There is only one operation: pour the contents of one container into the other stopping only when the pouring container is empty or the receiving container is full.
   - Is there a sequence of pourings that leaves exactly 2 liters in the 4 liter container?
   - Model this as a graph problem:
     - Define the graph.
     - What is the problem that needs to be solved?
     - What algorithm can be used to solve this problem?

5. Design and analyze an algorithm which takes as an input an undirected graph, G, and an edge, $e = (u, v)$, and determines whether G has a cycle containing e.

6. Prove the following theorem:
   In any connected undirected graph, there is a vertex whose removal leaves the graph connected.

7. Is it always possible to make an undirected graph with two connected components connected by adding a single edge?
   - Why or why not? (Proof or counterexample)
   - Does the same hold true for directed graphs (and strongly connected components rather than connected components)? (Proof or counterexample)

8. Design and analyze an algorithm for finding an odd length cycle in a strongly connected directed graph.

9. True or False? (Proof or counterexample) If a depth-first search on a directed graph $G = (V, E)$ produces exactly one back edge, then it is possible to choose an edge $e \in E$ such that the graph without e ($G' = (V, E \setminus \{e\})$) is acyclic.

10. True or False? (Proof or counterexample) If a directed graph $G = (V, E)$ is cyclic but can be made acyclic by removing one edge, then a depth-first search in G will encounter exactly one back edge.

11. Give an example of degree sequence d_1, d_2, d_3, d_4 where all $d_i \leq 3$ and $d_1 + d_2 + d_3 + d_4$ is even, but for which no graph with degree sequence d_1, d_2, d_3, d_4 exists (there is no graph such that the degree of vertex v_i is d_i).

12. Suppose you had a tree with 14 vertices, $A, B, \dots, N$. You wrote down the pre and post orderings for the vertices:

    Preorder: $A, B, D, E, H, L, M, I, F, C, G, J, K, N$

    Postorder: $D, L, M, H, I, E, F, B, J, N, K, G, C, A$

    Then you lost your graph. Can you reconstruct the tree? If so, do.