CS 3150X - Honors Algorithms
Practice Midterm 1

1. Draw the DFS tree of the graph in Figure 1 that will result if the first call to DFS is on vertex A and each vertex with its pre(in)-and post(out)-order numbers. Draw tree edges as solid lines, and back, cross, and forward edges as dashed lines. Put a 'B', 'C', or 'F' on each dashed line according to whether it is a back, cross, or forward edge.
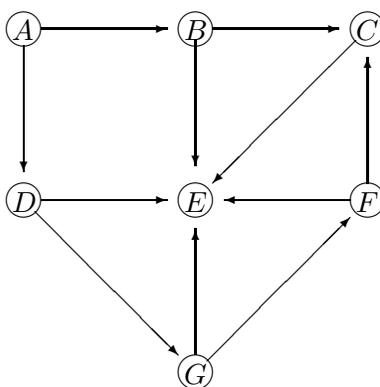

Figure 1: Graph for Problem 1. Assume for all figures that the adjacency lists are ordered so that vertices are ordered in alphabetical order in an adjacency list. (For example, the adjacency list of A has the edge to B before the edge to D.)

2. a) Execute Prim's algorithm on the graph of Figure 2 starting at vertex A. If there are any ties, the vertex with the lower letter comes first. List the edges in the order in which they are added to the tree.

   b) Execute Kruskal's algorithm on the graph of Figure 2 starting at vertex A. Assume that equal weight edges are ordered lexicographically by the labels of their vertices assuming that the lower labeled vertex always comes first when specifying an edge, e.g., (C, E) is before (C, F) which is in turn before (D, G). List the edges in the order in which they are added to the developing forest.

3. True/False

   For each statment below, say whether it is true or false, and give one sentence and/or one picture explanation.
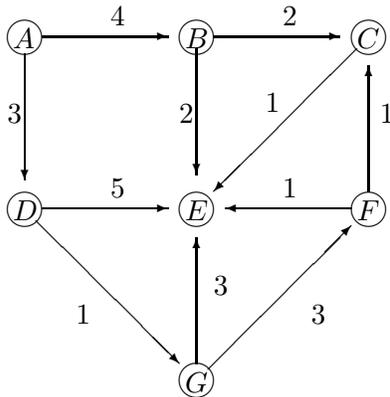
Figure 2: Graph for Problem 2.

a) $T(n) = 4T(n/2) + \log n$ is solvable with the Master Theorem.

b) Suppose T is a shortest paths tree for Dijkstra's algorithm. After adding $c > 0$ to every edge in the graph, T is still a shortest path for the modified graph.

c) It is possible to find a heaviest weight path of exactly k edges from s to each vertex in a directed graph in $O(k|E|)$ time.

d) The heaviest edge in a graph cannot belong to a minimum spanning tree.

4. Give a Θ characterization of the function $T(n)$ that satisfies the recurrence

$$T(n) = T(2n/3) + (\log_2 n)^2$$

Prove your answer correct.

5. A PERT diagram is a business standard for planning and charting a multi-task project. Each task to be performed is modeled as a vertex. The longest and shortest time, short(v) and long(v) are given for each task v. In addition, there are dependencies between the tasks modeled as directed edges, where $v \to w$ if task v must precede task w. For example, in a construction project the electrical and plumbing cannot be done until the framing has been completed.

Given a directed graph for a PERT chart, along with the short and long values for each vertex, give an algorithm that first checks that the diagram is acyclic, and then if so, gives the most optimistic (shortest) and most pessimistic (longest) completion times for the entire project.

6. Consider a positively weighted directed graph. Design the most efficient algorithm you can for finding a minimum weight simple cycle in the graph. Be sure to prove that your algorithm is correct and to argue carefully its running time.