1. Short questions

(a) (True or False) During BFS traversal, all nodes stored in the queue are colored gray, and all gray nodes are in the queue.

(b) (True or False) During BFS traversal of a graph, which of the following are possible contents of the queue. Why?
   - Node A ($d = 1$), Node B ($d = 1$), Node C ($d = 3$), Node D ($d = 2$).
   - Node A ($d = 1$), Node B ($d = 1$), Node C ($d = 1$), Node D ($d = 1$).
   - Node A ($d = 1$), Node B ($d = 1$), Node C ($d = 2$), Node D ($d = 2$).
   - Node A ($d = 1$), Node B ($d = 1$), Node A ($d = 2$), Node D ($d = 3$).

(c) For the DFS traversal, what’s the discover time and finish time of a node respectively? (i.e., explain what they represent.)

(d) Explain how DFS can be used in topological sorting, and why the resulting order satisfies the requirement of topological sorting (i.e., for two vertices $u, v$, with a directed edge from $u$ to $v$, $u$ always appear before $v$ in the ordering.)
2. Perform a depth-first search and breadth-first search on the following graphs; whenever there is a choice of vertices (for example, in the outer loop of DFS, or when checking the adjacent nodes of a give node), pick the one that is alphabetically first. (This means that in DFS of the undirected graph, we will start from node A; and for example, when checking the adjacent nodes from H, we will see node D first). Draw the BFS and DFS trees formed by the edges that the search algorithms used to find each vertex. For DFS, give an example of 1) a pair of nodes $u, v$, so that $[d[u], f[u]]$ is entirely within $[d[v], f[v]]$, 2) a pair of nodes $u, v$, so that $[d[u], f[u]]$ and $[d[v], f[v]]$, 2) are disjoint (i.e., no overlap in time).

- undirected graph

- Directed graph
3. Pouring water. We have three containers whose size are 10 pints, 7 pints, and 4 pints, respectively. The 7-pint and 4-pint containers start out full of water, but the 10-pint container is initially empty. We are allowed one type of operation: pouring the contents of one container into another, stopping only when the source container is empty or the destination container is full. We want to know if there is a sequence of pourings that leaves exactly 2 pints in the 7- or 4-pint container.

- Model this as a graph problem: give a precise definition of the graph involved and state the specific question about this graph that needs to be answered.

- Write the pseudocode that return the list of adjacent nodes of a given node in this graph.

```plaintext
// You can assume Node type stores three integers,
// a10: amount of water in 10-pint container
// a7: amount of water in 7-pint container
// a4: amount of water in 4-pint container
list<Node> CalculateAdjacentNode (Node curNode)
```
• What algorithm should be applied to solve the problem? Find the answer by applying the algorithm.
4. Give an efficient algorithm which takes as input a directed graph $G = (V, E)$, and determines whether or not there is a vertex $s \in V$ from which all other vertices are reachable. If the graph is undirected graph, would the algorithm be different?

5. Give an efficient algorithm which takes as input a undirected graph $G = (V, E)$ and determines whether the graph is a tree, i.e., the graph is connected (i.e., every node is reachable from every other node) and there is no cycle.
6. Suppose Dijkstra’s algorithm is run on the following graph, starting at node $B$.

- Draw a table showing the intermediate distance values of all the nodes at each iteration of the algorithm.

- Draw the final shortest-path tree.
7. Consider the following graph, and the problem of finding the minimum spanning tree of the graph.

- Run Prims algorithm; whenever there is a choice of nodes, always use reverse alphabetic ordering (e.g., start from node $H$). Draw a table showing the intermediate values of the cost array.

- Run Kruskals algorithm on the same graph. Show the edges that are added to the tree at each step. Whenever there are two candidate edges of same cost, the tie is broken arbitrarily.