This assignment should be completed individually. You are allowed to discuss the problems together, but not discuss the answers. You must write the solutions on your own, and list the collaborators you worked with.

**Problem 1. [15 points]**
In this problem, you should assume that $G$, shown below, is represented as a collection of adjacency lists, and that vertices are ordered alphabetically within each adjacency list.

![Graph](image)

**Figure 1:**

a. Draw the tree that is induced by performing breadth-first search starting at node $d$. Indicate clearly the order of actions taken by showing the contents of the lists $L[i]$ with every iteration. Clearly mark the edges that belong to $G$ but not in the tree.

b. Draw the tree that is induced by performing a depth-first-search starting at node $d$. Clearly mark the edges that belong to $G$ but not in the tree.

**Problem 2. [15 points]**
**KT Chapter 1 Problem 8.** The basic question is: Can a man or a woman end up better off by lying about his or her preferences? More concretely, we suppose each participant has a true preference order. Now consider a woman $w$. Suppose $w$ prefers man $m$ to $m'$, but both $m$ and $m'$ are low on her list of preferences. Can it be the case that by switching the order of $m$ and $m'$ on her list of preferences (i.e., by falsely claiming that she prefers $m'$ to $m$) and running the algorithm with this false preference list, $w$ will end up with a man $m''$ that she truly prefers to both $m$ and $m'$? (We can ask the same question for men, but will focus on the case of women for purposes of this question.)

Resolve this question by doing one of the following two things:

(a) Give a proof that, for any set of preference lists, switching the order of a pair on the list cannot improve a woman's partner in the Gale-Shapley algorithm; or

(b) Give an example of a set of preference lists for which there is a switch that would improve the partner of a woman who switched preferences.
Problem 3. [20 points]
KT Chapter] 3 Problem 2. (a) How can you detect cycles in an undirected graph? (b) Modify an algorithm that we already discussed in class to detect whether a given undirected graph contains a cycle. If the graph contains a cycle, then your algorithm should output one. (It should not output all cycles in the graph, just one of them.) Note: The running time of your algorithm should be $O(m + n)$ for a graph with $n$ nodes and $m$ edges.

Problem 4. [20 points]
KT Chapter] 3 Problem 7. Some friends of yours work on wireless networks, and they’re currently studying the properties of a network of $n$ mobile devices. As the devices move around (actually, as their human owners move around), they define a graph at any point in time as follows: there is a node representing each of the $n$ devices, and there is an edge between device $i$ and device $j$ if the physical locations of $i$ and $j$ are no more than 500 meters apart. (If so, we say that $i$ and $j$ are in range of each other.)

They’d like it to be the case that the network of devices is connected at all times, and so they’ve constrained the motion of the devices to satisfy the following property: at all times, each device $i$ is within 500 meters of at least $n/2$ of the other devices. (Well assume $n$ is an even number.)

What they’d like to know is: Does this property by itself guarantee that the network will remain connected? Here’s a concrete way to formulate the question as a claim about graphs.

Claim: Let $G$ be a graph on $n$ nodes, where $n$ is an even number. If every node of $G$ has degree at least $n/2$, then $G$ is connected.

(a) Decide whether you think the claim is true or false. (b) Give a counter-example if it is false, or clearly explain why it is true.

Problem 5. [20 points]
KT Chapter] 3 Problem 9. There’s a natural intuition that two nodes that are far apart in a communication network separated by many hops have a more tenuous connection than two nodes that are close together. There are a number of algorithmic results that are based to some extent on different ways of making this notion precise. Here’s one that involves the susceptibility of paths to the deletion of nodes.

Suppose that an $n$-node undirected graph $G = (V, E)$ contains two nodes $s$ and $t$ such that the distance between $s$ and $t$ is strictly greater than $n/2$.

(a) Show that there must exist some node $v$, not equal to either $s$ or $t$, such that deleting $v$ from $G$ destroys all $s-t$ paths. (In other words, explain that the graph obtained from $G$ by deleting $v$ contains no path from $s$ to $t$.)

(b) Give an algorithm with running time $O(m + n)$ to find such a node $v$.

Problem 6. [10 points - Proof Problem]
KT Chapter] 3 Problem 5. A binary tree is a rooted tree in which each node has at most two children. Show by induction that in any binary tree the number of nodes with two children is exactly one less than the number of leaves.

Problem 7. [Bonus 10 points] Challenge Problem. Solve this problem correctly, and you will get 15 bonus points towards this assignment or the next assignment.

KT Chapter 3 Problem 10. A number of art museums around the country have been featuring work by an artist named Mark Lombardi (1951-2000), consisting of a set of intricately rendered graphs. Building on a great deal of research, these graphs encode the relationships among people involved in major political scandals over the past several decades: the nodes correspond to participants, and each edge indicates some type of relationship between a pair of participants. And so, if you peer closely enough at the drawings,
you can trace out ominous-looking paths from a high-ranking U.S. government official, to a former business partner, to a bank in Switzerland, to a shadowy arms dealer.

Such pictures form striking examples of social networks, which, as we discussed in Section 3.1, have nodes representing people and organizations, and edges representing relationships of various kinds. And the short paths that abound in these networks have attracted considerable attention recently, as people ponder what they mean. In the case of Mark Lombardis graphs, they hint at the short set of steps that can carry you from the reputable to the disreputable, preferences.

Of course, a single, spurious short path between nodes v and w in such a network may be more coincidental than anything else; a large number of short paths between v and w can be much more convincing. So in addition to the problem of computing a single shortest v-w path in a graph G, social networks researchers have looked at the problem of determining the number of shortest v-w paths. This turns out to be a problem that can be solved efficiently. Suppose we are given an undirected graph \( G = (V,E) \), and we identify two nodes v and w in G. Give an algorithm that computes the number of shortest \( v \rightarrow w \) paths in G. (The algorithm should not list all the paths; just the number suffices.) The running time of your algorithm should be \( O(m + n) \) for a graph with \( n \) nodes and \( m \) edges.