In many of the questions on this assignment you are asked to describe an algorithm. This means to explain in words how your algorithm would solve the problem and then give a brief justification that your method has the claimed running time and is correct. You may give pseudocode but it is not necessary.

1. Consider the following task.

   **Input:** A connected, undirected graph $G$.
   
   **Question:** Is there an edge you can remove from $G$ while still leaving $G$ connected?

   (a) (10 pts) Describe a linear-time ($O(n + m)$) algorithm for solving this question.
   
   (b) (Bonus: 5 pts) Can you reduce the running time of your algorithm to $O(n)$?

2. Consider an undirected graph $G = (V, E)$ with distinct nonnegative edge weights $w_e \geq 0$. Suppose that you have computed a minimum spanning tree of $G$, and that you have also computed shortest paths to all nodes from a particular node $s \in V$.

   Now suppose each edge weight is increased by 1: the new weights are $w'_e = w_e + 1$.

   (a) (10 pts) Does the minimum spanning tree change? Give an example where it changes or prove it cannot change.
   
   (b) (10 pts) Do the shortest paths change? Give an example where they change or prove they cannot change.

3. (10 pts) You are given a strongly connected directed graph $G = (V, E)$ with positive edge weights along with a particular node $v_0 \in V$. Describe an $O((n + m) \lg n)$ time algorithm for finding shortest paths between all pairs of nodes, with the one restriction that these paths must all pass through $v_0$.

4. Here is a problem that occurs in automatic program analysis. For a set of variables $x_1, \ldots, x_n$, you are given some equality constraints, of the form $x_i = x_j$ and some disequality constraints, of the form $x_i \neq x_j$. Is it possible to satisfy all of them? For instance, the constraints $x_1 = x_2, x_2 = x_3, x_3 = x_4, x_1 \neq x_4$ cannot be satisfied.

   (a) (10 pts) Describe an $O(m + n \lg n)$ time algorithm that takes as input $m$
   
   constraints over $n$ variables and decides whether the constraints can be satisfied.
   
   (b) (Bonus: 5 pts) Can you reduce the running time of your algorithm to $O(n + m)$?