Problem 1.  [10 points - No help question]
The correctness of Prim’s algorithm. Explain how Prim’s algorithm can be used to find a minimum spanning tree of a graph. Why does the algorithm successfully find a minimum spanning tree?

Problem 2.  [20 points]
One of the basic motivations behind the Minimum Spanning Tree Problem is the goal of designing a spanning network for a set of nodes with minimum total cost. Here we explore another type of objective: designing a spanning network for which the most expensive edge is as cheap as possible.

Specifically, let $G = (V, E)$ be a connected graph with $n$ vertices, $m$ edges, and positive edge costs that you may assume are all distinct. Let $T = (V, E')$ be a spanning tree of $G$; we define the bottleneck edge of $T$ to be the edge of $T$ with the greatest cost.

A spanning tree $T$ of $G$ is a minimum-bottleneck spanning tree if there is no spanning tree $T'$ of $G$ with a cheaper bottleneck edge.

(a) Is every minimum-bottleneck tree of $G$ a minimum spanning tree of $G$? Prove or give a counterexample.

(b) Is every minimum spanning tree of $G$ a minimum-bottleneck tree of $G$? Prove or give a counterexample.

Problem 3.  [20 points]
Insertion sort can be expressed as a recursive procedure as follows. In order to sort $A[1..n]$, we recursively sort $A[1..n-1]$ and then insert $A[n]$ into the sorted array $A[1..n-1]$. Write pseudocode for this recursive version of Insertion sort, give a recurrence for the running time of the algorithm, and solve it using recurrence trees.

Problem 4.  [25 points]
Let us say that a graph $G = (V, E)$ is a near-tree if it is connected and has at most $n + 8$ edges, where $n = |V|$. Give an algorithm with running time $O(n)$ that takes a near-tree $G$ with costs on its edges, and returns a minimum spanning tree of $G$. You may assume that all the edge costs are distinct. You have to explain why the running time of your algorithm is $O(n)$.

Problem 5.  [25 points]
Consider the Minimum Spanning Tree Problem on an undirected graph $G = (V, E)$, with a cost $c_e \geq 0$ on each edge, where the costs may not all be different. If the costs are not all distinct, there can in general be many distinct minimum-cost solutions.

Suppose we are given a spanning tree $T \subset E$ with the guarantee that for every $e \in T$, $e$ belongs to some minimum-cost spanning tree in $G$. Can we conclude that $T$ itself must be a minimum-cost spanning tree in $G$? Give a proof or a counterexample with explanation.