MTL 342, Design and Analysis of Algorithms, Assignment 2

The points in the assignment will be for correctness and clarity of arguments as well as the readability of the write-up. Unnecessarily long answers might be penalized with negative points. For an algorithm, a proof of correctness and running time analysis are necessary to get full points.

Question 1.1
Let $G = (V, E)$ be a graph and let T be a minimum spanning tree of G. Let e be an edge in T, then show that there exists some cut $(S, V \setminus S)$ such that e is the edge with smallest weight crossing the cut. [2 points]

Question 1.2
Show that a graph has a unique minimum spanning tree if, for every cut of the graph, there is a unique edge with minimum crossing the cut. Show that the converse is not true by giving a counterexample. [3 points]

Question 1.3
You are given two sorted arrays of size n each. You can assume that all the $2n$ entries are different. Design an algorithm to find the median (n^{th} element) by accessing the arrays only $\mathcal{O}(\log n)$ times. In each access, you can ask for k^{th} smallest element of that array. [3 points]

Question 1.4
Let $G = (V, E)$ be an (undirected) graph with costs $c_e \geq 0$ on the edges $e \in E$. Assume you are given a minimum-cost spanning tree T in G. Now assume that a new edge is added to G, connecting two nodes $v, w \in V$ with cost c. Find an algorithm that detects if T is still a minimum spanning tree of the new graph $G' = (V, E \cup \{vw\})$ running in $\mathcal{O}(|E|)$ time. If it is not, find an algorithm running in $\mathcal{O}(|E|)$ time that computes a minimum spanning tree for G'. [$2 + 3$ points]

Question 1.5
Given an array of n objects, you need to decide if there is an object which is present more than $n/2$ times. The only operation by which you can access the objects is a function f, which given two indices i and j, outputs whether the objects at positions i and j in the array are identical or not. Give an $\mathcal{O}(n \log n)$-time divide-and-conquer algorithm for this (where each call to f is counted as one operation). [3 points]

Question 1.6
Consider an n-node complete binary tree T, where $n = 2^d - 1$ for some d. Each node v of T is labeled with a real number x_v. You may assume that the real numbers labeling the nodes are all distinct. A node v of T is a *local minimum* if the label x_v is less than the label x_w for all nodes w that are joined to v by an edge. You are given such a complete binary tree T, but the labeling is only specified in the following implicit way: for each node v, you can determine the value x_v by probing the node v. Show how to find a local minimum of T using only $O(\log n)$ probes to the nodes of T. [4 points]