1. We are given an undirected graph $G = (V,E)$ with edge-costs, as well as a minimum spanning tree $T$ with respect to these edge-costs. A new edge $(u,v)$ with cost $c$ is now added to $G$.

   (i) Show how, in $O(|E|)$ time, you can determine if $T$ still remains an MST.

   (ii) If $T$ is no longer an MST, show how you can construct an MST for the new $G$ in $O(|E|)$ time.

You can assume that $G$ and $T$ are represented using any type of data structure that you choose.

2. We have $n$ tasks, a shared supercomputer, and $n$ servers. Each task $i$ must first be preprocessed on the single supercomputer for $x_i$ units of time, after which it can be processed (as the only task) for $y_i$ units of time on server $i$ which has been dedicated to task $i$. The supercomputer can handle only one task at a time, which it must preprocess completely before moving to another task. We wish to find a schedule so that the time by which the last task completes (the makespan) is minimized; thus, our only “task” (pardon the pun) is to find the optimal way of scheduling the tasks on the supercomputer. Give a polynomial-time algorithm to find this optimal solution, and prove its correctness.

3. In the interval-scheduling problem (where we have to find a maximum-sized subset of compatible jobs), suppose we sort the jobs $i$ in non-decreasing order of their “length” (i.e., finish-time $f(i)$ minus start-time $s(i)$), and then greedily choose jobs in this order. (That is, we scan the jobs in this sorted order, and when we come to the $i$th job in this order, we choose it iff it is compatible with the set of jobs that have already been chosen.) Prove that this heuristic chooses at least half as many jobs as any optimal solution.

4. We are given two sets $X$ and $Y$, each containing $n$ positive reals. Our problem is to permute the elements of $X$, and to permute the elements of $Y$. All possible permutations are allowed; if we permute $X$ as $x_1, x_2, \ldots, x_n$ and $Y$ as $y_1, y_2, \ldots, y_n$, then the profit we obtain is $\prod_{i=1}^{n} x_i^2 y_i$. Develop an $O(n \log n)$-time to do the two required permutes, so as to maximize profit. Prove the correctness of your algorithm.