Midterm

Name: ____________________

1. Suppose you have a set of jobs $J_1, ..., J_n$ you need to complete, where job J_i will take you t_i minutes to complete with a deadline of d_i. We'll say the lateness of a job is 0 if it's finished before its deadline, otherwise it is the number of minutes past its deadline that it is completed. Your goal is to find an order to complete the jobs such that the summed lateness over all jobs in minimized.

   (a) (5 points) Define the slack s_i of job J_i by $s_i = d_i - t_i$. Show that the algorithm that completes jobs in order of slack (sort the jobs so that $s_1 \leq s_2 \leq s_3$ and then complete jobs in this order) is incorrect. For partial credit, explain to me what would be necessary to show. Hint: two jobs suffice, use the fact that you know how to make an optimal solution in part b.

   (b) (10 points) Show that the algorithm that completes jobs in order of deadlines is correct by means of an exchange argument. To be clear: what I'm looking for here is that you know what an exchange argument is and how to apply it, so full credit will only be given for a correct application of an exchange argument. If you can't get the exchange to work, I will give partial credit if you can explain what you're trying to do.

2. Consider the following problem: we are given a unimodal list of integers $x_1, ..., x_n$ without any repeats. Here unimodal means there is some j such that $x_1, ..., x_j$ is an increasing sequence while $x_{j+1}, ..., x_n$ is a decreasing sequence. The goal is to find this index j.

   (a) (10 points) Determine an $O(\log n)$ run time algorithm that solves this problem. Convince me of its correctness and that it is a $O(\log n)$ algorithm.

   (b) (5 points) (unrelated to this question as a whole) Let $T(n)$ denote the runtime of an algorithm on an input of length n, and suppose $T(n) \leq 3T(\frac{n}{3}) + cn$ for some constant c. Show that $T(n) = O(n \log n)$.

3. We are given a list of positive numbers $x_1, ..., x_n$ and a target value T. Our goal is to choose the sign of each number that that the sum of all numbers evaluates to the target value T. We know that each number x_i satisfies $1 \leq x_i \leq X$. For example, if $x_1 = 3$, $x_2 = 5$, $x_3 = 1$, and $T = 7$, then we should set the sign of x_3 to negative so that $x_1 + x_2 - x_3 = 3 + 5 - 1 = 7 = T$.

   (a) (5 points) Consider the following brute-force algorithm: try every possible assignment of signs for each number. For each assignment, check to see if the resultant sum is T and return the assignment. If no such assignment is found, then announce that no assignment is possible. What is the runtime of this algorithm?

   (b) (10 points) Determine an $O(n^2 X)$ time algorithm that correctly solves this problem. Convince me that the algorithm has $O(n^2 X)$ runtime, and that it is correct. Your algorithm should determine the assignment itself if it is possible.