In all problems (throughout this course) when you are asked to give an algorithm you are expected to: (1) give a clear description of the algorithm, (2) prove the algorithm outputs an optimal solution, (3) give the time complexity of the algorithm, and (4) prove that the algorithm has the stated time complexity.

On the top of your homework you must write and sign a statement that you have read and will follow the CS 441/539 collaboration policy. As stated in the policy be sure to acknowledge anyone whom you discussed the problems. Violations of this policy will carry severe penalties.

Core Problems

1. (5 pts) Suppose we want to make change for \( n \) cents, using the least number of coins of denominations 1, 10, and 25 cents. Consider the following greedy strategy: suppose the amount left to change is \( m \); take the largest coin that is no more than \( m \); subtract this coin’s value from \( m \), and repeat. Prove or disprove the correctness of this algorithm.

2. (10 pts) You are given a sequence of \( n \) songs where the \( i \)th song is \( \ell_i \) minutes long. You want to place all of the songs on an ordered series of CDs (e.g. CD 1, CD 2, CD 3, \ldots, CD \( k \)) where each CD can hold \( m \) minutes. Furthermore,

   (1) The songs must be recorded in the given order, song 1, song 2, \ldots, song \( n \).

   (2) No song may be split across CDs.

Your goal is to determine how to place them on the CDs as to minimize the number of CDs needed. Give the most efficient algorithm you can to find an optimal solution for this problem, prove the algorithm is correct and analyze the time complexity.

3. (10 pts) Suppose you are given \( n \) jobs \( j_1, \ldots, j_n \) that need times \( t_1, \ldots, t_n \), respectively, to complete on a single processor. Your goal is to schedule them on this processor in a way that the average completion time is as small as possible; assume that the first job starts at time zero; also assume that once a job is begun it must run to completion. As an example, let there be four jobs with \( t_1 = 15 \), \( t_2 = 8 \), \( t_3 = 3 \), and \( t_4 = 10 \). Then, the (non-optimal) schedule \( j_1, j_2, j_3, j_4 \) gives the average completion time \( (15 + 23 + 26 + 36)/4 = 25 \).

Design an \( O(n \log n) \) greedy algorithm to find an optimal solution for this problem.

4. (15 pts) Consider the following problem. You are given \( n \) events each of which takes one unit and is allowed to start at time \( t = 0, 1, 2, \ldots \). So, for example, an event can be scheduled from 0-1 or from 1-2, and so on. Event \( i \) is specified by an earliest start time \( s_i \) and a maximum delay \( d_i \) where both \( s_i \) and \( d_i \) are integers. Thus event \( i \) can be started at \( s_i, s_i + 1, \ldots, s_i + d_i \). Your goal is to schedule as many events as possible. Give the most efficient algorithm you can to solve this problem.

   \textit{Hint: Suppose several jobs have a start of time} \( t = 0 \), \textit{which one would you pick to schedule?}
5. (15 pts) A ski rental agency has $m$ pair of skis, where the height of the $i$th pair of skis is $s_i$. There are $n$ skiers who wish to rent skis, where the height of the $i$th skier is $h_i$. Ideally, each skier should obtain a pair of skis whose height matches his own height as closely as possible. Your goal is to assign skis to skiers so that the sum of the absolute differences of the heights of each skier and his ski is minimized.

(a) Give the most efficient algorithm you can to obtain an optimal solution to this problem when $m = n$.

(b) Now consider this problem when $m \geq n$. Prove whether or not the following greedy algorithm is optimal.

Let $H$ be the set of heights for the skiers
Let $S$ be the set of ski lengths
Repeat until each skier has skis
   Pick a height $h$ in $H$ and ski length $s$ in $S$ such that $|h-s|$ is the minimum possible
   Match height $h$ to ski length $s$
   Remove $h$ from $H$
   Remove $s$ from $S$

Advanced Problems, required for CS 539T (extra credit for CS 441T)

6. (5 pts) Consider the problem of making change from $n$ cents using the fewest coins when the available coins are quarters, dimes, nickels and pennies. Does the greedy strategy of outputting the largest coin that does not exceed the amount of change that must still be returned yield an optimal solution? Prove your answer is correct.

7. (10 pts) Consider the following scheduling problem. You are given $n$ jobs. Job $i$ is specified by an earliest start time $s_i$, and a processing time $p_i$. We consider a preemptive version of the problem where a job's execution can be suspended at any time and then completed later. For example if $n = 2$ and the input is $s_1 = 2$, $p_1 = 5$ and $s_2 = 0$, $p_2 = 3$, then a legal preemptive schedule is one in which job 2 runs from time 0 to 2 and is then suspended. Then job 1 runs from time 2 to 7 and finally, job 2 is completed from time 7 to 8. The goal is to output a schedule that minimizes $\sum_{j=1}^{n} C_j$ where $C_j$ is the time when job $j$ is completed. In the example schedule given above, $C_1 = 7$ and $C_2 = 8$.

Give the most efficient algorithm you can that computes an optimal preemptive schedule. Be sure to prove that your algorithm is correct and analyze the time complexity of your algorithm.

8. (15 pts) Suppose that we have a set of $n$ lectures to schedule among a large number of lecture halls. Lecture $i$ ($1 \leq i \leq n$) is specified by a start time $s_i$ and finish time $f_i$. The lecture uses the lecture hall during the interval $[s_i, f_i]$. (So a lecture with finishing time $t$ can be scheduled in the same lecture hall as a different lecture with a start time of $t$.) We wish to schedule all the activities using as few lecture halls as possible. Give an $O(n \log n)$ algorithm to find an optimal schedule. (For partial credit give an $O(n^2)$ algorithm.)