

Homework Assignment 1

September 5, 2002

Due Date: September 19

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.

You must submit your homework with a signed cover sheet attached to the front.

Core Problems

- (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.

Either give a counterexample, to prove that this algorithm can output a non-optimal solution, or prove that this algorithm always outputs an optimal solution.

- (10 pts) You are given a sequence of n songs where the i th song is ℓ_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, ..., CD k) where each CD can hold m minutes. Furthermore,
 - (1) The songs must be recorded in the given order, song 1, song 2, ..., song n .
 - (2) All songs must be included.
 - (3) 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.

- (10 pts) You are given n unit-duration jobs where job i has an integer deadline time $d_i \geq 0$ and real-valued penalty $p_i \geq 0$. The n jobs can be scheduled at times $0, 1, \dots, n - 1$ with each job scheduled exactly once and only one job can run at a time. If job i is completed by time d_i then there is no cost for it, but if job i completes after time d_i then a penalty of p_i is incurred. Your goal is to find a schedule which minimizes the total penalty.

Here is a proposed greedy algorithm. Sort the jobs so that $p_1 \geq p_2 \geq \dots \geq p_n$. Let the n possible time slots be initially empty where slot i is the unit-length slot that finishes at time i . We consider the jobs in the order $1, 2, \dots, n$. When considering job j , if any of time slots $1, \dots, d_j$ are available then job j is scheduled in the latest such slot. Otherwise, schedule job j in the latest available time slot (from those after d_j).

Either give a counterexample, to prove that this algorithm can output a non-optimal schedule, or prove that this algorithm always outputs an optimal schedule.

4. (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 skis 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)

5. (10 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.

6. (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.