

Practice Problems on Greedy Algorithms

September 7, 2004

Below are a set of three practice problems on designing and proving the correctness of greedy algorithms. For those of you who feel like you need us to guide you through some additional problems (that you first try to solve on your own), these problems will serve that purpose. *If anyone would like a help session where I guide you through the process of solving these problems, please let me know.*

The first four problems have fairly straight forward solutions. So if you want to just be sure you understand how to develop a greedy algorithm and prove it is correct (or incorrect) then you should work these problems. The last three problems are harder in both the algorithm needed and in the proof of correctness.

To help you check your work the solutions to all of these are on the course home page (www.classes.cec.wustl.edu/cse441) under “homeworks”. You can also bring your solution/write-up to any of our office hours and we’ll let you know if there were any problems. If you are having trouble and need a lot of guidance the TAs and I will use these practice problems to help you so that you can then do the HW problems on your own.

Practice Problems

1. Given a set $\{x_1 \leq x_2 \leq \dots \leq x_n\}$ of points on the real line, determine the smallest set of unit-length closed intervals (e.g. the interval $[1.25, 2.25]$ includes all x_i such that $1.25 \leq x_i \leq 2.25$) that contains all of the points.

Give the most efficient algorithm you can to solve this problem, prove it is correct and analyze the time complexity.

2. Suppose you were to drive from St. Louis to Denver along I-70. Your gas tank, when full, holds enough gas to travel m miles, and you have a map that gives distances between gas stations along the route. Let $d_1 < d_2 < \dots < d_n$ be the locations of all the gas stations along the route where d_i is the distance from St. Louis to the gas station. You can assume that the distance between neighboring gas stations is at most m miles.

Your goal is to make as few gas stops as possible along the way. Give the most efficient algorithm you can find to determine at which gas stations you should stop and prove that your strategy yields an optimal solution. Be sure to give the time complexity of your algorithm as a function of n .

3. 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.

4. 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.

5. You are given n events where each takes one unit of time. Event i will provide a profit of g_i dollars ($g_i > 0$) if started at or before time t_i where t_i is an arbitrary real number. (Note: If an event is not started by t_i then there is no benefit in scheduling it at all. All events can start as early as time 0.)

Given the most efficient algorithm you can to find a schedule that maximizes the profit.

6. 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. 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.