Basic Algorithms CSCI-UA.0310

Homework 7

Due: April 3rd, 11:59 PM EST

Instructions

Please answer each **Problem** on a separate page. Submissions must be uploaded to your account on Gradescope by the due date and time above.
Late submission for this homework is allowed. A late submission will be penalized 10% per day. No submission will be accepted after April 5th, 4 PM EST.

Problems To Submit

Problem 1 (10+10+5 points)

Recall that we designed a dynamic programming approach to solve the problem of making change for n cents using the least number of coins (refer to Problem 9 of ADDITIONAL PROBLEMS set). Now we want to solve this problem using a greedy approach.

(a) Design a greedy algorithm to make change for n cents using the least number of coins among quarters (25), dimes (10), nickels (5), and pennies (1). Fully explain your algorithm.
For example, if $n = 91$, your algorithm must return 6, since we can make change for 91 cents using 6 coins: Take 3 quarters, 1 dime, 1 nickel, and 1 penny. Also, we can show that it is not possible to make change for 91 cents using less than 6 coins.

(b) Show that your algorithm in part (a) outputs a correct result for all positive integers n.

(c) Provide an example of a set of coin denominations for which your greedy approach in part (a) does not output a correct result.
Note that your set of coin denominations must include a penny so that for every positive integer n, it is possible to make change for n cents using your set of coin denominations.

Problem 2 (10+10+5 points)

Recall the frog problem discussed in Homework 5 Problem 3:
Consider the array $A[1 \dots n]$ consisting of n non-negative integers. There is a frog on the last index of the array, i.e. the nth index of the array. In each step, if the frog is positioned on the i^{th} index, then it can make a jump of size at most $A[i]$ towards the beginning of the array. In other words, it can hop to any of the indices $i, \dots, i - A[i]$.

(a) Develop a GREEDY algorithm to determine whether the frog can reach the first index of A.

(b) Show the correctness of your algorithm in part (a).

(c) Find and justify the time complexity of your algorithm.

Problem 3 (13+12 points)

You are given a set $\mathcal{I}$ of n intervals on the real line. The starting and finishing times of these n intervals are given by the arrays $s[1 \ldots n]$ and $f[1 \ldots n]$, where $s[i]$ and $f[i]$ denote the starting time and the finishing time of the i^{th} interval in $\mathcal{I}$, respectively.
We want to color the intervals in $\mathcal{I}$ so that no two overlapping intervals are assigned the same color.

(a) Develop a GREEDY algorithm to compute the minimum number of colors needed to color $\mathcal{I}$ so that overlapping intervals are given different colors.

(b) Show the correctness of your algorithm in part (a) and find its running time.

Problem 4 (10+15 points)

Let P be a set of n points in the plane. The points of P are given one point at a time. After receiving each point, we compute the convex hull of the points seen so far.

(a) As a naive approach, we could run Graham's scan once after receiving each point, with a total running time of $O(n^2 \log n)$. Write down the psuedo-code for this algorithm.

(b) Develop an $O(n^2)$-time algorithm to solve the problem. Write down the pseudo-code of your algorithm and justify that the run-time of your algorithm is $O(n^2)$.