Fundamental Algorithms

Homework 5

Due: July 12th, 5:30 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.

Problems To Submit

Problem 1 (25 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 the 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). 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 (25 points)

Recall the frog problem discussed in Homework 4 Problem 1:
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) and find its running time.

Problem 3 (25 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 $s[1 \dots n]$ and $f[1 \dots n]$, 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 (25 points)

Let G be a directed graph on n vertices which is given as input by the adjacency matrix $V[1 \ldots n][1 \ldots n]$. Develop an $O(n)$-time algorithm to check if there is any vertex in G that has edges coming to it from all other vertices of G but no edges going out from it. Justify why your algorithm runs in $O(n)$ time.