CS 141, Fall 2018 Assignment 4

Posted: Novemner 10th, 2018 Due:November 24th, 2018, 11:59pm

Problem 1. (25 points) Suppose you are given a set $T = \{(s_1, f_1), \ldots, (s_n, f_n)\}$ of n tasks, where each task i is defined by the start time s_i and a finish time f_i. Two tasks t_i and t_j are $non - conflicting$ if $f_i \leq s_j$ or $f_j \leq s_i$. The *activityselection* problem asks for the largest number of tasks that can be scheduled in a non-conflicting way. The greedy algorithm explained in class considers tasks one by one ordered by the finish time. Ordering by increasing finish time is crucial to prove that this strategy always leads to the optimal solution.
Does the following greedy algorithm compute the optimal solution for activity selection?

1. Compute the number of overlaps for each task
2. Sort the task by the number of overlaps, in increasing order (break ties arbitrarily)
3. Pick the task i with the smallest number of overlaps, schedule it, and remove from further consideration tasks that are overlapping with i
4. Repeat step 3 until all tasks are scheduled

Note that the number of overlaps is NOT updated after step 1. If you think the strategy works, prove that the greedy choice property hold. If not, show an example where this strategy gives a suboptimal solution.

Problem 2. (25 points) Let $I_1, I_2, \ldots, I_n$ be a set of closed intervals on the real line, with $I_i = [a_i, b_i]$. Design an efficient greedy algorithm to compute the smallest set S of points such that each interval contains at least one point. Analyze the time complexity of your algorithm and prove that it always produces the optimal solution.

Problem 3. (25 points) Consider an array of integers,A. We define the absolute difference between two elements,a_i and a_j (where $i \neq j$), to be the absolute value of $a_i - a_j$. Describe an efficient greedy algorithm to find and print the minimum absolute difference between any two elements in the array.

Problem 4. (25 points) Assume you are an awesome parent and want to give your children some cookies. But, you should give each child at most one cookie. Each child i has a greed factor g_i, which is the minimum size of a cookie that the child will be content with; and each cookie j has a size s_j . If $s_j \geq g_i$, we can assign the cookie j to the child i, and the child i will be content. Your goal is to maximize the number of your content children and output the maximum number. Describe an efficient greedy algorithm for the same.
Note: You may assume the greed factor is always positive. Also you cannot assign more than one cookie to a child.

Problem 5. Optional

(25 points) 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