CSCI-GA.1170-001/002 Fundamental Algorithms	November 11, 2014

Solutions to Problem 1 of Homework 8 (3(+5) points)

Name: Keeyon Ebrahimi *Due: Wednesday, November 12*

For each of the following suggested greedy algorithms for the ACTIVITY-SELECTION problem, give a simple example of the input where the proposed greedy algorithm fails to compute the correct optimal solution.

(a) (3 points) Select the activity a_i with the shortest duration $d_i = f_i - s_i$. Commit to scheduling a_i. Let S_i' consist of all activities a_j which do not overlap with a_i: namely, either $f_j \leq s_i$ or $f_i \leq s_j$. Recursively solve ACTIVITY-SELECTION on S_i', scheduling the resulting activities together with a_i.

Solution:

This greedy approach will fail if the shortest activity disqualifies 2 other disjoint activities by overlapping both. Here is an example of this type of problem.

i	1	2	3	4	5
si	1	9	10	18	20
fi	10	12	20	22	34

In this example. the first block that will be picked is a_2, for it has the shortest duration of 3. The selection of a_2 disqualifies a_1 and a_3, for they have intersecting start and end times. The next shortest time will then be a_4 with a total time of 4. This will then disqualify a_5, which then in total gives this greedy algorithm a total of 2 activities.

We can see though that is is not the optimal, for the optimal is actually $a_1 \cup a_3 \cup a_5$
This has a total of 3 activities, while the previously suggested greedy algorithm only gives 2 activities. □

(b)* **(Extra Credit; 5 points)** For each activity a_i, let n_i denote the number of activities which do not overlap with a_i (e.g., n_i is the cardinality of the set S_i' defined above for specific a_i). Select the activity a_i with the largest number n_i of non-overlapping activities. Commit to scheduling a_i. Recursively solve ACTIVITY-SELECTION on the n_i activities in S_i', scheduling the resulting activities together with a_i.
(**Hint**: Unlike part (a), you might need a lot of activities for this counter-example. The smallest I know uses $n = 11$ activities. So don't be discouraged if small examples are all bad.)

Solution: ***************** INSERT PROBLEM 1b SOLUTION HERE *************** □

CSCI-GA.1170-001/002 Fundamental Algorithms	November 11, 2014

Solutions to Problem 2 of Homework 8 (11 Points)

Name: Keeyon Ebrahimi *Due: Wednesday, November 12*

Consider the problem of storing n books on shelves in a library. The order of the books is fixed by the cataloging system and so cannot be rearranged. The i-th book b_i, where $1 \leq i \leq n$ has a thickness t_i and height h_i stored in arrays $t[1 \ldots n]$ and $h[1 \ldots n]$. The length of each bookshelf at this library is L. We want to minimize the sum of heights of the shelves needed to arrange these books.

(a) (5 points) Suppose all the books have the same height h (i.e., $h = h_i$ for all i) and the shelves are each of height h, so any book fits on any shelf. The greedy algorithm would fill the first shelf with as many books as we can until we get the smallest i such that b_i does not fit, and then repeat with subsequent shelves. Using either the Greedy Always Stays Ahead or Local Swap method, show that the greedy algorithm always finds the shelf placement with the smallest total height of shelves, and analyze its time complexity.

Solution: ***************** INSERT PROBLEM 2a SOLUTION HERE ***************

□

(b) (6 points) Now assume that the books are not of the same height, and hence the height of any shelf is set to be the height of the largest book placed on that shelf. Show that the greedy algorithm in part (a) doesn't work for this problem. Give an alternative dynamic programming algorithm to solve this problem. What is the running time of your algorithm?

Solution: ***************** INSERT PROBLEM 2b SOLUTION HERE ***************

□

CSCI-GA.1170-001/002 Fundamental Algorithms	November 11, 2014

Solutions to Problem 3 of Homework 8 (12 Points)

Name: Keeyon Ebrahimi — *Due: Wednesday, November 12*

You want to travel on a straight line from from city A to city B which is N miles away from A. For concreteness, imagine a line with A being at 0 and B being at N. Each day you can travel at most d miles (where $0 < d < N$), after which you need to stay at an expensive hotel. There are n such hotels between 0 and N, located at points $0 < a_1 < a_2 < \ldots < a_n = N$ (the last hotel is in B). Luckily, you know that $|a_{i+1} - a_i| \leq d$ for any i (with $a_0 = 0$), so that you can at least travel to the next hotel in one day. You goal is to complete your travel in the smallest number of days (so that you do not pay a fortune for the hotels).

Consider the following greedy algorithm: "Each day, starting at the current hotel a_i, travel to the furthest hotel a_j s.t. $|a_j - a_i| \leq d$, until eventually $a_n = N$ is reached". I.e., if several hotels are within reach in one day from your current position, go to the one closest to your destination.

(a) (6 points) Formally argue that this algorithm is correct using the "Greedy Stays Ahead" method.
(**Hint**: Think how to define $F_i(Z)$. For this problem, the name of the method is really appropriate.)

Solution: ***************** INSERT PROBLEM 3a SOLUTION HERE *************** □

(b) (6 points) Formally argue that this algorithm is correct using the "Local Swap" method. More concretely, given some hypothetical optimal solution Z of size k and the solution Z^* output by greedy, define some solution Z_1 with the following two properties: (1) Z_1 is no worse than Z; (2) Z_1 agrees with greedy in the first day travel plan. After Z_1 is defined, define Z_2 s.t.: (1) Z_2 is no worse than Z_1; (2) Z_2 agrees with greedy in the first two days travel plan. And so on until you eventually reach greedy.

Solution: ***************** INSERT PROBLEM 3b SOLUTION HERE *************** □

CSCI-GA.1170-001/002 Fundamental Algorithms	November 11, 2014
Solutions to Problem 4 of Homework 8 (10 points)	
Name: Keeyon Ebrahimi	*Due: Wednesday, November 12*

Recall, Fibonacci numbers are defined by $f_0 = f_1 = 1$ and $f_i = f_{i-1} + f_{i-2}$ for $i \geq 2$.

(a) (2 points) What is the optimal Huffman code for the following set of frequencies which are the first 8 Fibonacci numbers.

Solution: ***************** INSERT PROBLEM 4a SOLUTION HERE *************** $\square$

(b) (4 points) Let $S_1 = 2 = f_0 + f_1$ and $S_i = S_{i-1} + f_i = \ldots = f_i + f_{i-1} + \ldots f_1 + f_0$ (for $i > 1$) be the sum of the first i Fibonacci numbers. Prove that $S_i = f_{i+2} - 1$ for any $i \geq 1$.

Solution: ***************** INSERT PROBLEM 4b SOLUTION HERE *************** $\square$

(c) (4 points) Generalize your solution to part (a) to find the shape of the optimal Huffman code for the first n Fibonacci numbers. Formally argue that your tree structure is correct, by using part (b).

Solution: ***************** INSERT PROBLEM 4c SOLUTION HERE *************** $\square$

CSCI-GA.1170-001/002 Fundamental Algorithms	November 11, 2014

Solutions to Problem 5 of Homework 8 (14 Points)

Name: Keeyon Ebrahimi *Due: Wednesday, November 12*

Little Johnny is extremely fond of watching television. His parents are off for work for the period $[S, F)$, and he wants to make full use of this time by watching as much television as possible: in fact, he wants to watch TV non-stop the entire period $[S; F)$. He has a list of his favorite n TV shows (on different channels), where the i-th show runs for the time period $[s_i, f_i)$, so that the union of $[s_i, f_i)$ fully covers the entire time period $[S, F)$ when his parents are away.

(a) (10 points) Little Johnny doesn't mind to switch to the show already running, but is very lazy to switch the TV channels, and so he wants to find the smallest set of TV shows that he can watch, and still stay occupied for the entire period $[S, F)$. Design an efficient $O(n \log n)$ greedy algorithm to help Little Johny. Do not forget to carefully argue the correctness of your algorithm, using either the "Greedy Always Stays Ahead" or the "Local Swap" argument.

Solution: ***************** INSERT PROBLEM 5a SOLUTION HERE ***************

$\square$

(b) (4 points). Assume now that Little Johnny will only watch shows from the beginning till end (except show starting before S or ending after F), but now he fetches another TV from the adjacent room, so that he can potentially watch up to two shows at a time. Can you find a strategy that will give the smallest set of TV shows that he can watch on the two TVs, so that at any time throughout the interval $[S, F)$ he watches at least one (and at most two) shows. (**Hint**: Try to examine your algorithm in part (a).)

Solution: ***************** INSERT PROBLEM 5b SOLUTION HERE ***************

$\square$