CS 141, Fall 2017 — Homework 6

Posted: November 6th, 2017 — Due: November 13th, 2017

Name:Zhenxiao Qi

Student ID #:500654348

- You are expected to work on this assignment on your own
- Use pseudocode, Python-like or English to describe your algorithms. Absolutely no C++/C/Java
- When designing an algorithm, you are allowed to use any algorithm or data structure we explained in class, without giving its details, unless the question specifically requires that you give such details
- Always remember to analyze the time complexity of your algorithms
- Homework has to be submitted electronically on Gradescope by the deadline. No late assignments will be accepted

Problem 1. (40 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.

Answer:

This problem is kind of like Activity Selection covered in class. First we sort these intervals by their lower bounds. Our greedy choice is intervals which are overlapped with the first interval. Then we choose one point within the overlapped part. Next we do greedy algorithm among the rest intervals. The time complexity of sorting is $O(n \log n)$ and $O(n)$ for choosing point. So the total time complexity is $O(n \log n)$

Prove of greedy choice property:

Suppose that $A \subseteq S$ is an optimal solution for S, and $\{K\}$ is the first in A. If $\{K\}$ is our greedy choice, then the optimal begins with our solution. If $\{K\}$ is not our greedy choice, which means $\{K\}$ is not all the intervals that are overlapped with the first interval. There will always be some points that should be include in set A. In conclusion, the optimal begins with our greedy choice

Prove of optimal substructure:

Suppose that $A = A' \cup$ {greedy} is not an optimal solution for S, in which case A' is the optimal solution for the sub-problem. Then there is a solution $B = B' \cup$ {greedy} is optimal for s with fewer points in it, but our premise is that A' already have fewest points in it(A' is already optimal). So $A = A' \cup$ {greedy} is an optimal solution for S

Problem 2. (60 points)

In the United States, coins are minted with denominations of 1, 5, 10, 25, and 50 cents. Now consider a country whose coins are minted with denominations of $\{d_1, \ldots, d_k\}$ units. They seek an algorithm that will enable them to make change of n units using the minimum number of coins.

1. (20 pts) The greedy algorithm for making change repeatedly uses the biggest coin smaller than the amount to be changed until it is zero. Provide a greedy algorithm for making change of n units using US denominations. Prove its correctness and analyze its time complexity.

2. (20 pts) Show that the greedy algorithm does not always give the minimum number of coins in a country whose denominations are $\{1, 6, 10\}$.

3. (20 pts) Give dynamic programming algorithm that correctly determines the minimum number of coins needed to make change of n units using denominations $\{d_1, \ldots, d_k\}$. Analyze its running time.

Answer:

1. First we sort the denominations, and find the biggest coin smaller than n, then we take this coins as many as we can and compute how much we need for the rest. Next we use this greed strategy for the rest. The time complexity is $O(n \log n)$ for sorting.
   Prove of greedy choice property:
   Suppose that A is set of denominations which is an optimal solution for S, then we sort A by the value of th elements. Let K be the biggest denomination that is smaller than n. If $K =$ the first element in A, then the optimal solution begins with our greedy choice.Else, we can replace the first element in A with our greedy choice for the reason that our greedy choice takes fewer coins.
   prove of optimal substructure:
   Suppose that $A = A' \cup$ {greedy} is not an optimal solution for S, in which A' is the optimal solution for the sub-problem. Then there will be a solution $B = B' \cup$ {greedy} which is optimal for S, which, however, contradict with the fact that A' is the minimum number of coins(A' is optimal). So $A = A' \cup$ {greedy} is an optimal solution for S.

2. For $n = 15$, the greedy algorithm will choose 10 first and the rest is 5 which is smaller than 6, so next the greedy algorithm will choose five 1 to make up. The total amount of coins is 6. However the minimum number of coins is 5 for two coins of 6 and three coins of 1

3. Suppose that the denominations $D = \{d_1, \ldots, d_k\}$ is sorted. First we make change using d_1 for $i = 0$ to $i = k$, then we make change using d_2 according to the following rule.

$min(i, change) = min(min(i, change - d_i) + 1, min(i - 1, change))$, i means using the first i denominations to make change. The minimum number of coins equals to the minimum between the number of coins using $i - 1$ to make change and that using a combination of them to make change. So the number in (k, n) is the minimum number of coins when make change of n units. This is kind of like a $n * k$ memory table, the time complexity is $O(nK)$.