CS-310, Fall 2021
Assignment 4
Assigned: Nov. 26, Due: Friday, Dec. 10, 11:55PM

November 25, 2021

Total Points = 70

Problem 1. (Points: 10) Suppose a job j_i is given by two numbers d_i and p_i, where d_i is the deadline and p_i is the penalty. The length of each job is equal to 1 minute. All n such jobs are to be scheduled, but only one job can run at any given time. If job j_i does not complete before its deadline d_i, its penalty p_i should be paid. Design a greedy algorithm to find a schedule which minimizes the sum of penalties and prove its optimality.

Problem 2. (Points: 20) A thief enters a store and sees a set I of n items, $I = \{a_1, a_2, \cdots, a_n\}$. Each item has an associated weight w_i and value v_i. Ideally, the thief would like to steal everything in order to gain maximum benefit. However, there is only so much he can carry. The thief has a knapsack with capacity C. The thief now has to determine which items to steal so that their total weight does not exceed C and their total value is maximum.

1. Suppose the items are such that a fraction of an item can be taken, i.e. the thief can take some part of an item (for example, $0.4w_i$ of a_i) and leave the remaining part. This is called the *Fractional Knapsack Problem.* Let $A \subset I$ be the subset of items that the thief steals. Devise an $O(n \log n)$ greedy algorithm to find the subset A such that $\sum_{a_i \in A} w_i < C$ and $\sum_{a_i \in A} v_i$ is maximum.

2. Suppose items can be taken as a whole, i.e. the thief can only take or leave an item; he can not take a fraction of an item. This is called the *Binary Knapsack Problem.* Does your greedy algorithm for the fractional version of the problem (previous question) still find an optimal solution? Justify your answer.

Problem 3. (Points: 20) Coins of a set of denominations (values) are available to a cashier who needs to provide change for a given amount V, such that the number of coins returned in exchange for V is minimum. We assume an infinite supply of each denomination.

1. Given the denomination set $\{1, 5, 10\}$, devise a greedy algorithm to find the minimum number of coins, that can be exchanged for V.

2. Is your greedy algorithm optimal for *any* given set of denominations? Justify.

Problem 4. (Points: 10) The counting sums problem is to count the number of ways a number can be written as the sum of two or more positive integers. For example, we can write 6 as the sum of two or more positive integers in the following ways

$$
\begin{aligned}
5+1&=6\\
4+2&=6\\
4+1+1&=6\\
3+3&=6\\
3+2+1&=6\\
3+1+1+1&=6\\
2+2+2&=6\\
2+1+1+2&=6\\
2+1+1+1+1&=6\\
1+1+1+1+1+1&=6
\end{aligned}
$$

Using a dynamic programming approach, write an algorithm to determine the number of ways 100 can be written as the sum of two or more positive integers.

Problem 5. (Points: 10) Consider the coin change problem: Coins of a set of denominations (values) are available to a cashier who needs to provide change for a given amount V , such that the number of coins returned in exchange for V is minimum. We assume an infinite supply of each denomination. Devise a dynamic programming algorithm which, given a set of denominations $D = \{d_1, \cdots, d_k\}$ and value V, produces the optimal result for the coin change problem. Briefly argue about the optimal substructure property of the problem.