6331 - Algorithms, Autumn 2016, CSE, OSU
Homework 5
Instructor: Anastasios Sidiropoulos

Problem 1. The set of valid coins in the US consists of 25, 10, 5, and 1 cent (i.e. the quarter, the dime, the nickel, and the penny). Given a certain amount k in cents, we wish to find a set of coins of minimum size with total value that sums up to k. For example, for $k = 103$ the answer should be $25, 25, 25, 25, 1, 1, 1$ (that is, four quarters and three pennies).

Consider the following greedy algorithm for this problem: Start by taking as many 25-cents coins as possible, then add as many 10-cent coins as possible, then add as many 5-cent coins as possible, and finally add as many 1-cent coins as possible. Is this algorithm correct? Justify your answer.

Problem 2.

(a) You are given a set of n items of sizes $a_1, \ldots, a_n \in \mathbb{N}$, and a bin of size $B \in \mathbb{N}$. Your goal is to find a maximum cardinality subset of items that all fit inside the bin. That is, you want to find a set of distinct indices $I = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$, such that

$$a_{i_1} + \ldots + a_{i_k} \leq B,$$

maximizing k.

For example, if the sizes are $a_1 = 7$, $a_2 = 4$, $a_3 = 5$, and the bin size is $B = 10$, then the optimum solution is $I = \{2, 3\}$ (that is, picking the second and the third item).

Design a greedy algorithm for this problem. The running time of your algorithm should be polynomial in n.

(b) Suppose that instead of maximizing k, we want to maximize the total size of the items in the bin; that is, we want to maximize the quantity

$$\text{size}(I) = a_{i_1} + \ldots + a_{i_k}.$$

Show that your greedy algorithm does not work in this case.

(c) Suppose that the maximum item size is at most 100; that is, $a_1, \ldots, a_n \in \{1, \ldots, \mathbf{100}\}$. Design a dynamic programming algorithm for picking a set of items that all fit in the bin, maximizing $\text{size}(I)$. The running time of your algorithm should be polynomial in n.