Homework 3

February 2020

1 Files on an external drive - continued

Consider problem 1 from Homework 2. Now add the requirement that you would like to maximize the usage of the external drive, that is, you aim to use as much as possible of the G gigabytes.

(a) Does the Greedy strategy from Homework 2 work for this variation of the problem? Find a counterexample, preferably with as few files as possible.

(b) Suggest a Dynamic programming algorithm that solves the problem. First explain your algorithm briefly in words, then present pseudo-code.

(c) What is the running time of your algorithm in terms of $O()$? Is your algorithm a polynomial algorithm?

turn page

2 Making change

Once upon a time, people used coins in stores to buy items.
It still can occur in certain parts of the world.

Given a number of coin values, e.g. 1 SEK, 5 SEK, 10 SEK. Given furthermore a product with the price p and bank-note $B > p$. If a customer pays a product with price p with the the bank-note B, how do you choose the coins so that the change will consist of a minimum amount of coins?
For certain coin sets, as the one above, and assuming that there is an infinite amount of coins of each value, one can come up with a Greedy algorithm to find the minimum number of coins when giving back the change to a customer.
However, this does not work for all coin sets, let us investigate the general case: Given some set of coin values, assume that we have an infinite amount of coins for each value. Further assume that there is a coin value 1 (otherwise the problem of making change may be infeasible for some values of p and B).

1. Consider the following straightforward Greedy strategy: Start with $v = B - p$ and choose the largest coin value that fits, subtract that coin value from v and repeat (possibly using multiple coins of the same coin value).

   Make up a coin system where your Greedy algorithm will fail, that is your Greedy algorithm will make you pay with more coins than necessary for a specific price.

   Answer this question by writing down a price and a set of available coins. You also should give the optimal number of coins and then the number of coins given by your Greedy algorithm.

2. Devise a Dynamic programming (DP) that will work for any coin systems and that will ensure you can pay with the smallest number of coins possible.

   (a) How does the $OPT()$ function look like? Explain what your $OPT()$ expresses.

   (b) What values for initialization are suitable?

   (c) Write down your method as pseudo-code.

3. What is the running time of your DP algorithm in terms of $O()$?