

Homework 2

January 2020

1 Files on an external drive

Suppose you have an external drive with capacity G , G integer. Given further a list of files $f_1 \dots f_n$ that you would like to save on the drive. Each file has a certain size, file f_i takes up g_i gigabytes. However, the total sum of all g_i is larger than G , so your goal is to find a largest subset of the files that fits on the drive with capacity G .

- (a) How many different ways are there to choose a subset of the files that may or may not fit on the drive?
- (b) Suggest a greedy 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 $O()$? Do you think there could be another algorithm with better time complexity in terms of $O()$? (No formal proof needed, just write down your thoughts.)
- (d) Prove that your greedy algorithm will give you an optimal solution (this sub-problem is more difficult, you may skip it).
- (e) Can you think of a special case of the problem, maybe a special property of the g_i so that you can come up with an algorithm that is faster (in terms of $O()$) than the one you suggested in (b)? If yes, write down the algorithm, give the running time in terms of $O()$ and explain why it is faster.

2 Consultancy business scheduling

Suppose you're running a lightweight consulting business. Your clients are distributed between the East Coast and the West Coast, and this leads to the following question. Each month, you can either run your business from an office in New York (NY) or from an office in San Francisco (SF). In month i , you'll incur an operating cost of N_i if you run the business out of NY; you'll incur an operating cost of S_i if you run the business out of SF. (It depends on the distribution of client demands for that month.) However, if you run the business out of one city in month i , and then out of the other city in month $i + 1$, then you incur a fixed moving cost of M to switch base offices. Given a sequence of n months, a plan is a sequence of n locations: each one equal to either NY or SF such that the i^{th} location indicates the city in which you will be based in the i^{th} month. The cost of a plan is the sum of the operating costs for each of the n months, plus a moving cost of M for each time you switch cities. The plan can begin in either city.

The problem: Given a value for the moving cost M , and sequences of operating costs N_1, \dots, N_n and S_1, \dots, S_n , find a plan of minimum cost.

- (a) How many different plans exist?
- (b) Give an example to show that a greedy algorithm does not correctly solve this problem. Try to choose a small example, i.e., a small n .
- (c) Give an efficient algorithm that takes values for n , M , and sequences of operating costs N_1, \dots, N_n and S_1, \dots, S_n , and returns the cost of an optimal plan. We recommend you use pseudo-code.
- (d) What is the running time of your algorithm in terms of $O()$?