Algorithms. Assignment 3

Problem 5

During n consecutive days, one job per day is offered to you. On a daily basis you can decide to accept today's job or not. If you do the job of the ith day, you get paid p_i units of money. The payments $p_1, \ldots, p_n$ are all known in advance. However there is a condition: it is forbidden to work on any two consecutive days. The problem is to select a subset of jobs that respects this condition and maximizes the amount of money you earn.

(a) A tempting greedy algorithm is: Choose a job with highest payment. Delete the jobs of the previous and the next day. Keep on following this rule, until every job is either chosen or deleted. – Give a counterexample that fools this greedy rule.

(b) Here is another greedy-like algorithm: Either work on all even days, or work on all odd days. From these two options choose the more profitable one. – Again, give a counterexample that fools this greedy rule.

If you have the time and leisure, feel free to try even more greedy rules that may come to mind ...

Next you want to spend the money (may it be the optimal amount or not) that you have earned. See the next page:

Problem 6

Walking through the town you see a shop that runs a "3 for 2" campaign: If you buy 3 items, you'll get the cheapest one for free. Let us assume that the shop also applies this rule if you buy any number n of items. That is, you can group them into triples, and from every triple you have to pay only the 2 highest prices. (If 3 does not divide n, then 1 or 2 items are left, and you have to pay for them as well.) Incidentally you are interested in n specific items. Their prices are visible. How would you group them in triples, so as to save the maximum amount of money?

(a) Give a fast algorithm that outputs an optimal partitioning into triples. How much time does it take?

Perhaps this was easy, and intuitively you did it right. But now:

(b) Prove that your proposed algorithm guarantees, in fact, an optimal solution. That is, you must show that no other solution can be cheaper.

Advice on Problem 6

It is amazing that the correctness proof is much more challenging than guessing a good algorithm. One can easily choose an awkward approach to the proof, and then get lost in a jungle of unnecessary case distinctions. Therefore we give some extra advice on how to make it simple. (You may ignore the following spoiler if you are already good at proofs.)

Do not explicitly consider the triples, but only the set F of items that you get for free. Sort the set A of all items by descending prices. Then consider how the elements of F may be located in this sorted sequence, and get the optimum by, e.g., some exhange argument.