Design and Analysis of Algorithms (2022 Spring)
Problem Set 4

4 problems, 25 points each
and 3 exercise problems (do NOT turn in)

Due Date: April 21st, 2022

1. (25 points) Consider a sequence alignment problem over an alphabet of size k, $Z = \{z_1, z_2, \ldots, z_k\}$, with given gap costs $c(z, z') > 0$ for any $z \neq z' \in Z$, and a given mismatch cost $c(-) > 0$. Assume that each of these parameters is a positive integer.

   Suppose you are given two strings $A = a_1 a_2 \ldots a_m$ and $B = b_1 b_2 \ldots b_n$. Give an $O(mn)$ algorithm to find a minimum-cost alignment between A and B, where the cost of aligning z with z' is 0 if $z = z'$, and $c(z, z')$ if $z \neq z'$, and the cost of aligning any letter with a dash is $c(-)$.

2. (25 points) Consider the following variant of interval scheduling, where each job j is associated with not only a time interval, specified with the start time s_j and finish time f_j, but also a value v_j. We want to choose a subset of compatible jobs whose sum of values is maximized. Design a dynamic programming algorithm for this problem and reason about its running time.

3. (25 points) There are n kinds of coins carrying integral values $v_1, v_2, \cdots, v_n$, respectively, and there is unlimited supply to each kind of coins.

   (a) (25 points) Give an algorithm to determine whether it is feasible to make change for a cheque with a given integral value X (i.e., to find a set of coins whose total value is X). Note that each kind of coins can be used multiple times. The running time of your algorithm should be $O(nX)$.

   (b) (optional, 0 points) Modify your algorithm to compute the minimum number of coins used to make change for a given integral value X, if it is feasible.

4. (25 points) Let $A = \{a_1, a_2, \cdots, a_n\}$ be a set of distinct integers, and let $S = a_1 + a_2 + \cdots + a_n$. Assume that the sum S is an even number. Give an algorithm to check whether A can be divided into two subsets such that the sum of the numbers in each subset is exactly equal to $\frac{S}{2}$. Analyze the time complexity of the algorithm.

   Hint: You may want to consider a more general question asking whether there is a subset of numbers from A that sums to a given target $0 \leq T \leq S$.

5. (0 points) This question considers how to divide the chapters in a story into volumes (think the seven volumes of Harry Potter or however many Game of Thrones volumes there will be) so as to equalize the size of the volumes. The input consists of positive integers $x_1, x_2, \ldots, x_n$ and $k \leq n$. Here x_i is the number of pages in chapter i, and k is the desired number of volumes. The problem is determine which chapters go into each of the k volumes so as to minimize the difference between the most number of pages in any volumes, and the least number of pages in any of the volumes. Of course you can not reorder the story. Give an algorithm whose running time is bounded by a polynomial in n.You running time should not depend on the number of pages in the chapters.

   Hint: You may want to first find an algorithm whose running time depends on the number of pages. Then, you can remove this dependency by observing that there are only $O(n^2)$ possible number of pages in a chapter since a chapter is comprised of some consecutive chapters.

6. (0 points) A fisherman catches a golden fish. It speaks like a real human being:

   > *"Put me back, old man, into the ocean. I will pay you a right royal ransom,*
   > *I will give you whatever you ask me."*

   The fisherman asks for an accurate prediction of the stock prices in the next n days (for simplicity, assume that there is only one stock). The golden fish does just that, and p_i is the price on day i. As it goes back to the ocean, the golden fish remarks that the magic will disappear and the predictions will no longer be accurate after one buy and one sell. Now the fisherman wishes to find two days $1 \leq i < j \leq n$ such that buying on day i and selling on day j maximizes the his gain, i.e., to maximize $\frac{p_j}{p_i}$.

   (a) (0 points) Design a polynomial-time algorithm for the fisherman.

   (b) (0 points) What if two buys and two sells are allowed before the magic disappears? Could you design a polynomial-time algorithm for this variant?

7. (0 points) The owners of an independently operated gas station are faced with a following situation. They have a large underground tank in which they store gas; the tank can hold up to L gallons at one time. Ordering gas is quite expensive, so they want to order relatively rarely. For each order, they need to pay a fixed price P for delivery in addition to the cost of the gas ordered. However, it costs c to store a gallon of gas for an extra day, so ordering too much ahead increases the storage cost.

   They are planning to close for a week in the winter, and they want their tank to be empty by the time they close. Luckily, based on years of experience, they have accurate projections for how much gas they will need each day until this points in time. Assume that there are n days left until they close, and they need g_i gallons of gas for each of the days $i = 1, 2, \ldots n$. Assume that the tank is empty at the end of day 0. Give an algorithm to decide on which days they should place orders, and how much to order so as to minimize their total cost.