Algorithms (CSCI-261-03, CSCI-264), Fall 2023/24
Homework 4, due Friday, Nov 3, 2023, 11:59pm

Problem 1

Given is a sequence of n positive numbers $a_1, a_2, \ldots, a_n$ and a positive integer p. We say that a subsequence $a_{j_1}, a_{j_2}, \ldots, a_{j_k}$, where $j_1 < j_2 < \ldots < j_k$, has *gap* at most p, if $j_{\ell+1} - j_\ell \leq p$ for every $\ell \in \{1, 2, \ldots, k-1\}$. In words, when we circle the elements of the subsequence in the original sequence, every pair of adjacent circled elements will be at most p locations apart. Give an $O(np)$ algorithm that finds the largest sum of an increasing subsequence of $a_1, \ldots, a_n$ with gap at most p.

For example, for input $8, 1, 7, 10, 2, 3, 4, 5$ and $p = 1$, we are looking for an increasing subsequence that consists of adjacent elements: Such subsequence with the largest sum is $1, 7, 10$, with sum 18.

Problem 2

Recall the (unweighted) interval scheduling problem: Given are n intervals $(s_1, f_1), (s_2, f_2), \ldots, (s_n, f_n)$, where $s_i < f_i$ for every $i \in \{1, 2, \ldots, n\}$, and your task is to find a largest subset of non-overlapping intervals. If these intervals represent courses (that is, the i-th course starts at time s_i and finishes at time f_i), then, of course, you'll need extra time to move from one course to the next. Suppose in addition to the intervals, we also have a set of positive numbers $b_{i,j}$ for $i, j \in \{1, 2, \ldots, n\}$: the number $b_{i,j}$ determines how much time it takes to move from the location of the i-th course to the location of the j-th course. What is the largest number of courses we can take in this scenario? Design an $O(n^2)$ algorithm for this problem.

Problem 3

Consider a 2-backpack version of Knapsack. Given are two backpacks of capacities W_1 and W_2. Given are also n items $(w_1, c_1), (w_2, c_2), \ldots, (w_n, c_n)$, where w_i is the weight and c_i the cost of the i-th item. We want to find a set of items that can be split into two parts: one that fits in the first backpack and the other in the second backpack, and the sum of their costs is the largest possible. All the numbers are positive, and W_1, W_2, and all the w_i's are integers. Give an $O(nW_1W_2)$ algorithm that finds a set of items to include in the first backpack and a set of items to include in the second backpack such that for each backpack its weight limit is satisfied and the total cost of these items is the largest possible.

Problem 4

In the Matrix Chain Multiplication problem we are given $a_0, a_1, \ldots, a_n$ that denote the dimensions of n matrices - the i-th matrix A_i is of dimensions $a_{i-1} \times a_i$, and we want to find a parenthesizing that minimizes the number of operations needed to multiply $A_1 A_2 \ldots A_n$. We assume that the number of operations needed to multiply two matrices of dimensions $p \times q$ and $q \times r$ is pqr. In class we will soon discuss how to compute the minimal cost. Give an

$O(n^3)$ algorithm that finds a corresponding parenthesizing (more precisely, use $O(n^3)$ time to find the optimal cost, then $O(n)$ time to find the parenthesizing).

Problem 5 (CSCI-264 only)

Given is a sequence $a_1, a_2, \ldots, a_n$ of numbers. We say that a subsequence $a_{j_1}, a_{j_2}, \ldots, a_{j_k}$, where $j_1 < j_2 < \ldots < j_k$, is *convex* if $a_{j_{i-1}} + a_{j_{i+1}} \geq 2a_{j_i}$ for every $i \in \{2, 3, \ldots, k-1\}$. Give an $O(n^3)$ algorithm that finds the length of a longest convex subsequence.