Exercise II

Remarks: In all the algorithms, always explain their correctness and analyze their complexity. The complexity should be as small as possible. A correct algorithm with large complexity, may not get full credit.

Solve the following 5 questions.

Question 1: Consider an array of numbers. Give an algorithm that finds the maximum contiguous subarray that is increasing.

Question 2: Give an algorithm that finds the maximum size subarray (the entries may not be contiguous) that forms an increasing sequence.

Question 3: Given a gas station with two pumps, and a collection of cars 1, 2, . . . , n with filling time $s_i$ for item $i$ (on both pumps). Find a schedule that assigns cars to the two pumps, so that if the first pump is assigned a sum of $t_1$ and the second a sum of $t_2$, max{$t_1, t_2$} is minimum. Note that you only have to decide which car is assigned to pump 1 and which to pump 2, because the order will not change the sum.

Remark: This will minimize the maximum time that any of the two pumps are ”busy”, hence it is of benefit for the gas station.

Question 4: The minimum distance of an unsorted array array is $\min_{i,j, i \neq j} |A[i] - A[j]|$. Namely, if we sort the array this value is the minimum absolute values of the difference between two contiguous elements. Consider an $n \times n$ matrix $A$. Say that we want to choose one element from every row, $A[1, i], A[2, j], A[3, k], \ldots A[n, p]$. The problem is to find an array with minimum distance under this constraint. Namely, we want to find a minimum distance array under the constraint that the first item belongs to the first row, the second item belongs to the second row, and so on. Give an algorithm for the problem.

Question 5: Given an array of numbers, which may be positive or negative, we consider the problem of finding the subarray (the items do not need to be contiguous) of maximum sum. Give an algorithm for this problem.