1 (3). The problem's input is numbers $n, k > 1$ and a list $a_1, \dots, a_n$ of positie integers. Construct an $O(nk)$ algorithm that computes $\max\limits_{0<|i-j|\leqslant k} a_i \times a_j$, i. e. the maximal product of different elements with distance at most k. Try to construct an algorithm that uses $O(k)$ RAM (you can read the input sequence by elements).

Comment: *The problems marked Shen have solutions written in the book "Algorithms and Programming: Problems and Solutions". Try to solve them by yourself and then compare with the author's solution. You shall not include solutions of these problems in your submission.*

2[Shen1.3.2]. Two sequences `x[1]` ... `x[n]` and `y[1]` ... `y[k]` of integers are given. Determine if the second sequence is a subsequence of the first one; that is, if it is possible to delete some terms of the first sequence to obtain the second one. The number of operations should be $O(n + k)$.

3 (3). There is an array of pairs $[(l_1, r_1), \dots, (l_n, r_n)]$. A pair (l_i, r_i) defines a segment $[l_i, r_i]$ on a line. Construct an $O(n \log n)$ algorithm that computes the Jordan measure of the union of the segments $\bigcup\limits_{i=1}^{n} [l_i, r_i]$, i.e. the union is a set of non-intersecting segments, the measure is the sum of their lengths.

4 (5). The input is an array $a := [a_1, \dots, a_n]$ of different numbers. Construct an $O(n \log n)$ algorithm that cuts this array into the list of arrays such that the concatenation of sorted arrays from the list equals to the sorted array a. Moreover, the number of cuts should be maximal. More formally, cut is defined by a sequence of indices $i_1 < \dots < i_k$ and consists of arrays

$$[a_1, \dots a_{i_1}],\ [a_{i_1+1}, \dots a_{i_2}],\ \dots,\ [a_{i_{k-1}+1}, \dots a_{i_k}].$$

In other words, we take the minimal number of continuous non-overlapping subarrays of a that covers a, sort them, and get the sorted array a as the result.

5 (3). A *peak* of an array $[a_1, a_2, \dots, a_n]$ is an element a_i such that

$$a_{i-1} \leqslant a_i \geqslant a_{i+1}$$

for $1 < i < n$ or the only of the corresponding inequalities holds for $i \in \{1, n\}$ ($a_1 \geqslant a_2$, $a_n \geqslant a_{n-1}$). An array a of integers is stored in RAM. Construct an $O(\log n)$ algorithm that finds a peak of a.

6 (6). An array $[a_1, a_2, \dots, a_n]$ of integers is stored in RAM. Construct an $O(n)$ algorithm that cuts the array into three parts $[a_1, \dots, a_i]$, $[a_{i+1}, \dots, a_j]$, and $[a_{j+1}, \dots, a_n]$ such that at least two parts have positive sums of their elements. It is guaranteed that such a cut exists. The output is indices i and j.