1.1. **For each task from this block, state for which set of values and which associative operation you need to build a tree of segments. The structure of the tree itself and the algorithm for performing operations cannot be changed.** There is an array a of n integers and two operations: assign a value: $a_i = x$, and one of the following:

(1.1.1) find the minimum on the interval from l to r, and the number of elements equal to this minimum

(1.1.2) find the minimum on the segment from l to r, and the index of the leftmost element equal to this minimum

(1.1.3) find the value of the sum $a_l - a_{l+1} + a_{l+2} - a_{l+3} + ... \pm a_{r-1}$

(1.1.4) find the value of the sum $a_l + 2a_{l+1} + 3a_{l+2} + ... + (r-l)a_{r-1}$

(1.1.5) find on a given segment $[l, r)$ a subsegment $[l_1, r_1)$ ($l \le l_1 \le r_1 \le r$), the sum on which is maximum (it is enough to find this sum, but you can also also find the segment)

1.2. **In tasks from this block, you need to add a new operation to the segment tree.** There is an array a of n integers and two operations: assign a value: $a_i = x$, and one of the following:

(1.2.1) find the minimum i for which $a_i \ge k$

(1.2.2) find all i for which $a_i \ge k$ in time $O(x \log n)$, where x is the size of the output

(1.2.3) find the minimum i on the segment from l to r, for which $a_i \ge k$

1.3. There is a parking for n spaces. Each space can be occupied or free. Operations need to be processed: mark space as occupied/free, and one of the following:

(1.3.1) find the number of free spaces on the segment from l to r

(1.3.2) find k-th free space

(1.3.3) find free space closest to i

1.4. There is a string of n brackets. You need to process requests: 1) change the i-th bracket, 2):

(1.4.1) check if the substring from l to r is a valid brackets sequence

(1.4.2) find the largest prefix of a substring from l to r that is a valid sequence