Problem 6 - K-Least Element in the Sequence (100 pts)

Problem Description

Giver is a beginner at programming, so he practices hard everyday for improving his skill. One day, someone asks him a problem: "Given a sequence $a_1, a_2, \ldots, a_n$, please find the k-least element in the sequence." However, Giver is so clever that this problem is too easy for him. Therefore, He tries to modify the problem to make it harder. Now, the problem becomes as follows:

Given a sequence $a_1, a_2, \ldots, a_n$, you are asked to support the following operations.

- Insert i x - insert an integer x before the ith element of the sequence. If $i - 1$ equals the length of the sequence, then insert x at the end of it.
- Delete i - delete the ith element of the sequence.
- Reverse l r - reverse the elements between the interval $[l, r]$ of the sequence.
- Query l r k - find the k-least element between the interval $[l, r]$ of the sequence.

Input

The first line contains two integers n, q ($1 \leq n, q \leq 50000$), representing the length of the initial sequence and the number of operations respectively. The second line contains the initial sequence, which consists of n integers $a_1, a_2, \ldots, a_n$ ($-10^5 \leq a_1, a_2, \ldots, a_n \leq 10^5$).

Each of the next q lines is an operation with the format mentioned above.

It is guaranteed that all operations are valid. Formally, suppose that the current length of the sequence is N, then it must satisfy the following constraints.

- Insert i x: $1 \leq i \leq N + 1$, $-10^5 \leq x \leq 10^5$.
- Delete i: $1 \leq i \leq N$.
- Reverse l r: $1 \leq l \leq r \leq N$.
- Query l r k: $1 \leq l \leq r \leq N$, $1 \leq k \leq r - l + 1$.

Output

For each "Query l r k" operation, print an integer, representing the k-least integer between the lth element and the rth element.

Subtask 1 (20 pts)

- $n, q \leq 3000$.

Subtask 2 (25 pts)

- There are only “Query” operations.

Subtask 3 (5 pts)

- There are only “Insert”, “Delete”, “Reverse” operations.

Subtask 4 (50 %)

- No other constraints.

Sample Input 1

```
5 5
-10 1 4 -3 -5
Query 4 4 1
Query 1 2 1
Query 4 5 2
Query 2 3 2
Query 4 5 1
```

Sample Output 1

```
-3
-10
-3
4
-5
```

Sample Input 2

```
5 5
-2 4 4 -7 2
Delete 5
Delete 1
Insert 2 6
Query 3 3 1
Query 4 4 1
```

Sample Output 2

```
4
-7
```

Sample Input 3

```
5 5
-6 7 3 3 0
Reverse 1 4
Insert 3 -6
Query 3 4 1
Reverse 4 5
Query 1 4 2
```

Sample Output 3

```
-6
-6
```

Hints

1. You may need the data structure like linked list to support efficient modification.
   - Due to the bad performance on interval operation and indexing, you may want to reduce the number of nodes.
   - Try to bind some elements into one node, that is, store a sequence of elements using an sequence in a node.
   - Try to maintain the nodes such that both the number of nodes and the number of elements in a node are not too big. That is, you may need to split a node or rebuild the data structure if one of them is too large.
2. How to reverse a segment that can be represented by a sequence of nodes? what if not, can you make it become the same case? You may need to add a tag on each node and reverse it some time later to preserve a good time complexity.
3. You may need to use the binary search algorithm when finding the k-least element. You may also need to maintain some more information for each node to support efficient query.