Problem B
Range Order Statistic Queries

Time limit: 3 seconds
Memory limit: 2048 megabytes

Problem Description

Given an array of n integers $a_1, a_2, \ldots, a_n$ and q queries to be performed on these integers. Each query belongs to one of the following two types:

1. 1 idx_i v_i —Change a_{idx_i} to v_i. That is, $a_{idx_i} := v_i$.
2. 2 l_i r_i k_i —Find the k_i-th order statistic of the numbers $a_{l_i}, a_{l_i+1}, \ldots, a_{r_i}$. In other words, sort the numbers $a_{l_i}, a_{l_i+1}, \ldots, a_{r_i}$ in ascending order and output the k_i-th element. **Note that k_i is at most 5 in this problem.**

Can you answer all these queries?

Input Format

The first line of the input contains two integers n and q. The second line of the input contains n space-separated integers $a_1, a_2, \ldots, a_n$. The i-th of the following q lines contains the i-th query in the format as in the problem description.

Output Format

For each query of type 2, output the k_i-th order statistic in one line.

Technical Specification

- $1 \leq n, q \leq 2 \times 10^5$
- $1 \leq a_i \leq 10^9$ for $i = 1, 2, \ldots, n$
- $1 \leq idx_i \leq n$ for $i = 1, 2, \ldots, q$ of query type 1
- $1 \leq v_i \leq 10^9$ for $i = 1, 2, \ldots, q$ of query type 1
- $1 \leq l_i \leq r_i \leq n$ for $i = 1, 2, \ldots, q$ of query type 2
- $1 \leq k_i \leq \min(5, r_i - l_i + 1)$ for $i = 1, 2, \ldots, q$ of query type 2
- It is guaranteed that there is at least one query of type 2 in the input.

Sample Input 1

```
5 4
3 2 1 5 4
2 2 4 1
2 2 4 3
1 3 3
2 2 4 1
```

Sample Output 1

```
1
5
2
```

Sample Input 2

```
6 7
3 6 10 3 5 8
2 1 6 5
2 2 5 3
1 3 1
2 3 6 4
1 6 4
2 4 6 2
2 2 2 1
```

Sample Output 2

```
8
6
8
4
6
```

Hint

Use a segment tree to perform these queries. Since the the value of k_i is at most 5, we can keep the 5 smallest values in each node of the segment tree.

When constructing the segment tree, each internal node has to fetch the 5 smallest values from its left child and 5 from its right child, and select the 5 smallest values among them. This can

be done by merging two sorted arrays and select the first few elements of the merged array. In C++, you can use the 'std::merge' function from the standard library.

Updates and queries can also be performed in a similar way.