IOI Training Camp 2017
Team Selection Tests, Day 3

Walled Inversions

You are given an array A of N distinct integers. This is followed by Q updates, which take the form of placing *walls* between some two consecutive elements. After each update, you should find and output the total number of walled inversions: that is, inversions in the array which do not cross any *walls*.
Formally, we associate a wall with the index to its left. That is, a wall between the 2nd and 3rd elements would be called a wall at index 2.
Suppose after u updates, there are walls at the indices $X_1, X_2, \ldots, X_u$ such that $1 \leq X_1 < X_2 < \ldots < X_u < N$. Then the total number of walled inversions is the sum of the number of inversions in each of the segments: Number of inversions in $[1, X_1]$ + Number of inversions in $[X_1 + 1, X_2]$ + ... + Number of inversions in $[X_u + 1, N]$.
The number of inversions in a range $[L, R]$ is the number of (i, j) pairs such that $L \leq i < j \leq R$ and $A[i] > A[j]$.
A pair of indices (i, j) $(i < j)$ crosses a wall placed between indices k and $k + 1$ if $i \leq k$ and $j > k$.
No two walls are placed between the same two elements. And the walls are placed permanently.

Input

The first line of the input contains 2 integers, N and Q.
The next line contains N integers: $A[1], A[2], \ldots, A[N]$.
The i-th of the next Q lines contains an integer, W_i, which denotes that the i-th wall is placed between the indices W_i and $W_i + 1$.

Output

After every update, output the total number of walled inversions, as described above, in a new line.

Constraints

Unless specially mentioned:

- $2 \leq N \leq 10^5$
- $1 \leq Q \leq N - 1$
- $1 \leq A[i] \leq 10^9$
- All elements of A are distinct.
- $1 \leq W_i \leq N - 1$
- W_is are distinct.

Subtasks

Subtask 1 (14 Points):

- $2 \leq N \leq 2000$

Subtask 2 (20 Points):

- $2 \leq N \leq 20000$

Subtask 3 (66 Points):

- No Additional Constraints.

Sample Input 1

```
3 2
1 2 3
2
1
```

Sample Output 1

```
0
0
```

Sample Input 2

```
3 2
3 2 1
2
1
```

Sample Output 2

```
1
0
```

Sample Input 3

```
6 5
3 5 1 6 4 2
3
5
1
2
4
```

Sample Output 3

```
5
3
2
1
0
```

Sample Input 4

```
10 9
3 7 1 9 4 10 8 2 6 5
9
8
7
6
5
4
3
2
1
```

Sample Output 4

```
16
12
6
4
4
2
2
0
0
```

Explanation

In the third sample, after placing the second wall, the array looks like [351|64|2], where "|" denotes a wall. The number of inversions in [351] is 2, because 3 appears before 1, and is to its left, and similarly 5 is before 1. The number of inversions in [64] is 1. The number of inversions in [2] is 0.
Hence, the total number of walled inversions is 2+1+0 = 3, and hence the second output of that sample is 3.

Limits

Time: 2 seconds

Memory: 256 MB