Balanced Sequence

Let us define a sequence as a **balanced sequence** as follows:

- Every sequence of length 1 is a balanced sequence.
- A sequence $S = [S_0, \ldots, S_{2k}]$ of length $2k+1$ is a balanced sequence if it satisfies the following conditions:
  - $[S_0, S_1, \ldots, S_{k-1}]$ is a balanced sequence.
  - $[S_{k+1}, S_{k+2}, \ldots, S_{2k}]$ is a balanced sequence.
  - S_k is the **unique** maximum among all elements of the sequence S.

A sequence A consisting of N integers is given. $A[i \ldots j]$ denotes the sequence of length $j-i+1$ consisting of elements from index i to index j of sequence A. For example, if $A = [3, 5, 7, 2, 9]$, then $A[1 \ldots 3]$ is $[5, 7, 2]$, and $A[4 \ldots 4]$ is $[9]$.

Q queries are given. Each query is an operation that changes a specific element of the sequence. This operation is cumulative. Find the total number of pairs of integers (i, j) such that $0 \le i \le j \le N-1$ and $A[i \ldots j]$ is a balanced sequence, for the initial state and after performing each query.

Implementation Details

You must implement the following functions.

```
long long initialize(int N, vector<int> A)
```

- N: The length of the sequence A.
- A: An integer array of length N.
- This function should return the total number of pairs of integers (i, j) such that $0 \le i \le j \le N-1$ and $A[i \ldots j]$ is a balanced sequence.
- This function is called exactly once initially.

```
long long update_sequence(int p, int v)
```

- This function represents a query changing the value of $A[p]$ to v.
- This function should return the total number of pairs of integers (i, j) such that $0 \le i \le j \le N-1$ and $A[i \ldots j]$ is a balanced sequence after $A[p]$ is changed.
- This function is called a total of Q times after the `initialize` function is called.

You should not execute any input/output functions in any part of the source code submitted.

Constraints

- $1 \leq N \leq 10^5$
- $0 \leq Q \leq 10^5$
- $1 \leq A[i] \leq 10^9$ for all i ($0 \leq i \leq N-1$)
- $0 \leq p \leq N-1, 1 \leq v \leq 10^9$ for all `update_sequence` calls

Subtasks

No.	Points	Limits
1	3	$Q = 0$, A is a balanced sequence.
2	5	$Q = 0, A[i] \leq 3$
3	12	$A[i] \leq 3, v \leq 3$
4	18	$Q = 0, N \leq 2\,000$
5	26	$Q \leq 10$
6	36	No additional constraints.

Examples

Example 1

Consider the case where $N = 4, Q = 0, A = [1, 1, 1, 1]$.

The grader calls the following function:

```
initialize(4, [1, 1, 1, 1])
```

The list of (i, j) where $A[i \ldots j]$ is a balanced sequence is $(0, 0), (1, 1), (2, 2), (3, 3)$, so it should return 4.

Example 2

Consider the case where $N = 12, Q = 0, A = [8, 9, 7, 9, 2, 3, 2, 8, 4, 6, 2, 6]$.

The grader calls the following function:

```
initialize(12, [8, 9, 7, 9, 2, 3, 2, 8, 4, 6, 2, 6])
```

The called function returns 18.

Example 3

Consider the case where $N = 7, Q = 2, A = [1, 3, 4, 4, 2, 1, 6]$.

The grader calls the following functions in order:

```
initialize(7, [1, 3, 4, 4, 2, 1, 6])
update_sequence(3, 1)
update_sequence(3, 2)
```

The called functions return $7, 9, 8$ respectively.

Sample Grader

The input format of the sample grader is as follows:

- line 1: $N\ Q$
- line 2: $A[0]\ A[1] \ldots A[N-1]$
- For all $1 \leq k \leq Q$:
  - line $2+k$: $p\ v$ (arguments for the k-th `update_sequence`)

The sample grader outputs the answer in the following format:

- line 1: return value of `initialize`
- For all $1 \leq k \leq Q$:
  - line $1+k$: return value of the k-th `update_sequence`