

## 균형잡힌 수열

다음과 같은 수열을 균형잡힌 수열이라고 정의하자:

- 길이가 1인 모든 수열은 균형잡힌 수열이다.
- 길이  $2k + 1$ 의 수열  $S = [S_0, \dots, S_{2k}]$ 가 다음 조건을 만족하면 균형잡힌 수열이다.
  - $[S_0, S_1, \dots, S_{k-1}]$ 이 균형잡힌 수열이다.
  - $[S_{k+1}, S_{k+2}, \dots, S_{2k}]$ 가 균형잡힌 수열이다.
  - $S_k$ 는 수열  $S$ 의 모든 원소 중 유일한 최댓값이다.

$N$ 개의 정수로 이루어진 수열  $A$ 가 주어진다.  $A[i \dots j]$ 는 수열  $A$ 의  $i$ 번 원소부터  $j$ 번 원소까지로 구성된 길이가  $j - i + 1$ 인 수열이다. 예를 들어  $A = [3, 5, 7, 2, 9]$ 일 때,  $A[1 \dots 3]$ 은  $[5, 7, 2]$ 이고,  $A[4 \dots 4]$ 는  $[9]$ 이다.

$Q$ 개의 쿼리가 주어진다. 각 쿼리는 수열의 특정 원소를 변경하는 연산이다. 이 연산은 누적된다. 초기 상태와 각 쿼리를 수행한 이후,  $0 \leq i \leq j \leq N - 1$ 이며  $A[i \dots j]$ 가 균형잡힌 수열인 정수 두 개의 쌍  $(i, j)$ 의 총 개수를 구하여라.

### 함수 목록 및 정의

여러분은 아래 함수를 구현해야 한다.

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

- $N$ : 수열  $A$ 의 길이.
- $A$ : 길이가  $N$ 인 정수 배열.
- 이 함수는  $0 \leq i \leq j \leq N - 1$ 이며  $A[i \dots j]$ 가 균형잡힌 수열인 정수 두 개의 쌍  $(i, j)$ 의 총 개수를 반환해야 한다.
- 이 함수는 초기에 단 한 번만 호출된다.

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

- 이 함수는  $A[p]$ 의 값을  $v$ 로 변경하는 쿼리를 나타낸다.
- 이 함수는  $A[p]$ 가 변경된 이후  $0 \leq i \leq j \leq N - 1$ 이며  $A[i \dots j]$ 가 균형잡힌 수열인 정수 두 개의 쌍  $(i, j)$ 의 총 개수를 반환해야 한다.
- 이 함수는 `initialize` 함수가 호출되고 나서 총  $Q$ 번 호출된다.

제출하는 소스 코드의 어느 부분에서도 입출력 함수를 실행해서는 안 된다.

## 제약 조건

- $1 \leq N \leq 10^5$
- $0 \leq Q \leq 10^5$
- 모든  $i$ 에 대해  $1 \leq A[i] \leq 10^9$  ( $0 \leq i \leq N - 1$ )
- 모든 `update_sequence` 호출에 대해  $0 \leq p \leq N - 1, 1 \leq v \leq 10^9$

## 서브태스크

번호	배점	제한
1	3	$Q = 0$ , $A$ 는 균형잡힌 수열이다.
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	추가적인 제약조건이 없다.

## 예제

### 예제 1

$N = 4, Q = 0, A = [1, 1, 1, 1]$ 인 경우를 생각해 보자.

그레이더는 아래 함수를 호출한다.

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

$A[i \dots j]$ 가 균형잡힌 수열인  $(i, j)$ 의 목록은  $(0, 0), (1, 1), (2, 2), (3, 3)$ 이므로, 4를 반환해야 한다.

### 예제 2

$N = 12, Q = 0, A = [8, 9, 7, 9, 2, 3, 2, 8, 4, 6, 2, 6]$ 인 경우를 생각해 보자.

그레이더는 아래 함수를 호출한다.

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

호출된 함수는 18을 반환한다.

### 예제 3

$N = 7, Q = 2, A = [1, 3, 4, 4, 2, 1, 6]$ 인 경우를 생각해 보자.

그레이더는 아래 함수들을 순서대로 호출한다.

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

호출된 함수들은 차례대로 7, 9, 8을 반환한다.

## 샘플 그레이더

샘플 그레이더의 입력 형식은 다음과 같다.

- line 1:  $N Q$
- line 2:  $A[0] A[1] \dots A[N - 1]$
- 모든  $1 \leq k \leq Q$ 에 대해:
  - line  $2 + k$ :  $p v$  ( $k$ 번째 `update_sequence`의 인자)

샘플 그레이더는 다음 형식으로 답을 출력한다:

- line 1: `initialize`의 반환값
- 모든  $1 \leq k \leq Q$ 에 대해:
  - line  $1 + k$ :  $k$ 번째 `update_sequence`의 반환값