

뗏목 제작 (raft)

울창한 두 숲 KOI 숲과 IOI 숲에는 많은 나무들이 우거져 있다. KOI 숲에는 0번부터 $N - 1$ 번까지, 총 N 개의 나무가 왼쪽에서 오른쪽으로 일렬로 줄지어 있으며, IOI 숲에는 0번부터 $M - 1$ 번까지 총 M 개의 나무가 같은 방식으로 일렬로 줄지어 있다. KOI 숲의 i ($0 \leq i \leq N - 1$)번 나무의 높이는 $A[i]$, IOI 숲의 j ($0 \leq j \leq M - 1$)번 나무의 높이는 $B[j]$ 이다.

당신의 부족은 외딴 섬에서 바다의 신께 기도를 드리는 오랜 전통을 가지고 있으며, 그 의식을 치르기 위해 뗏목을 만들어야 한다. 당신은 두 숲에서 나무를 벌목하여 뗏목을 제작하기로 하였다.

뗏목을 타고 섬까지 가는 것은 매우 위험하기 때문에, 당신은 뗏목을 최대한 안정적으로 만들고자 한다. 뗏목은 나무들을 옆으로 이어붙여 만들며, 그 안에 포함되는 직사각형 모양의 면적이 클수록 안정적이다. 구체적으로, 뗏목을 구성하는 나무의 높이가 왼쪽에서부터 $H[0], H[1], \dots, H[L - 1]$ 이라고 할 때, 뗏목의 안정성은

$$\max_{0 \leq s \leq l \leq L-1} (\min(H[s], \dots, H[l]) \times (l - s + 1))$$

의 값으로 정의된다. 즉, 가능한 모든 연속된 구간 $[s, l]$ 에 대해 그 구간에 포함되는 나무들의 최소 높이를 구하고, 그 높이에 구간 너비인 $(l - s + 1)$ 를 곱했을 때의 최댓값을 의미한다.

부족의 전통에 따라, 뗏목을 구성하는 나무들은 다음과 같은 규칙을 따라야 한다.

1. 벌목한 나무는 모두 뗏목에 사용해야 한다.
2. KOI 숲에서 잘라온 나무들은 원래 숲 내에서의 순서를 유지해야 한다. 즉, 원래 숲에서 나무 X 가 나무 Y 보다 왼쪽에 있었다면, 뗏목에서도 X 가 Y 보다 왼쪽에 있어야 한다.
3. IOI 숲에서 잘라온 나무들도 마찬가지로 원래 순서를 유지해야 한다.

KOI 숲의 나무들은 0번부터 $N - 1$ 번까지 N 그룹 모두를 모두 벌목하기로 결정되었다. 그러나, IOI 숲에서 어떤 나무들을 벌목할지는 아직 결정되지 않았다. 당신은 0부터 $Q - 1$ 까지 번호 붙여진 Q 개 질의에 답해야 한다. 질의들은 길이 Q 의 배열 L 과 R 로 표현된다. 질의 k ($0 \leq k \leq Q - 1$)의 답은 IOI 숲에서 번호가 $L[k]$ 이상 $R[k]$ 이하인 나무들을 벌목하는 경우 얻을 수 있는 뗏목의 가능한 최대 안정성이다.

함수 목록 및 정의

여러분은 아래 함수를 구현해야 한다.

```
std::vector<long long> max_stability(std::vector<int> A, std::vector<int> B,
std::vector<int> L, std::vector<int> R)
```

- A : 크기가 N 인 정수 배열
- B : 크기가 M 인 정수 배열
- L, R : 크기가 Q 인 정수 배열
- 이 함수는 크기가 Q 인 정수 배열을 반환한다.

- 반환되는 정수 배열을 C 라 할 때, 0 이상 $Q-1$ 이하의 정수 k 에 대해 $C[k]$ 는 KOI 숲의 모든 나무와 IOI 숲의 번호가 $L[k]$ 이상 $R[k]$ 이하인 나무들을 벌목하는 경우에 뗏목의 가능한 최대 안정성을 나타내는 정수이다.
- 이 함수는 매 테스트 케이스마다 단 한 번 호출된다.

제출하는 소스 코드의 어느 부분에서도 입출력 함수를 실행해서는 안 된다.

제약 조건

- $1 \leq N, M \leq 150\,000$
- $1 \leq Q \leq 500\,000$
- 모든 $0 \leq i \leq N - 1$ 에 대해 $1 \leq A[i] \leq 10^9$
- 모든 $0 \leq j \leq M - 1$ 에 대해 $1 \leq B[j] \leq 10^9$
- 모든 $0 \leq k \leq Q - 1$ 에 대해 $0 \leq L[k] \leq R[k] \leq M - 1$

부분문제

1. (10점)

- $N, M, Q \leq 3\,000$

2. (8점)

- $Q \leq 300$

3. (20점)

- 모든 $0 \leq k \leq Q - 1$ 에 대해 다음 두 조건이 성립한다.
 - $L[k] = 0$ 또는 $B[L[k] - 1] < \min(B[L[k]], B[L[k] + 1], \dots, B[R[k]])$
 - $R[k] = M - 1$ 또는 $B[R[k] + 1] < \min(B[L[k]], B[L[k] + 1], \dots, B[R[k]])$

4. (6점)

- 모든 $0 \leq i \leq N - 1$ 에 대해 $A[i] \leq 50$
- 모든 $0 \leq j \leq M - 1$ 에 대해 $B[j] \leq 50$

5. (14점)

- 모든 $0 \leq i \leq N - 1$ 에 대해 $A[i]$ 가 일정하다.

6. (11점)

- 모든 $0 \leq j \leq M - 2$ 에 대해 $B[j] \geq B[j + 1]$

7. (13점)

- 모든 $0 \leq k \leq Q - 1$ 에 대해 $L[k] = 0$

8. (7점)

- 모든 $0 \leq k \leq Q - 1$ 에 대해 $R[k] - L[k]$ 이 동일하다.

9. (11점)

- 추가적인 제약 조건이 없다.

예제 1

$N = 5, M = 4, Q = 2, A = [3, 3, 1, 6, 1], B = [3, 5, 7, 6], L = [0, 0], R = [1, 3]$ 인 경우를 생각해 보자. 그레이더는 다음과 같이 함수를 호출한다.

```
max_stability([3, 3, 1, 6, 1], [3, 5, 7, 6], [0, 0], [1, 3])
```

$L[0] = 0, R[0] = 1$ 인 경우 뗏목의 최대 안정성은 12이다. 12의 안정성을 얻는 방법 중 하나는 다음과 같다:

- 뗏목의 왼쪽부터 KOI 숲의 0,1번 나무, IOI 숲의 0,1번 나무, KOI 숲의 2,3,4번 나무 순서로 구성한다. 이 때 뗏목을 이루는 나무들의 높이는 왼쪽부터 $H = [3, 3, 3, 5, 1, 6, 1]$ 이 된다. $\min(H[0], H[1], H[2], H[3]) = 3$ 이므로 $3 \times 4 = 12$ 의 안정성을 가진다.

$L[0] = 0, R[0] = 3$ 인 경우 뗏목의 최대 안정성은 20이다. 20의 안정성을 얻는 방법 중 하나는 다음과 같다:

- 뗏목의 왼쪽부터 KOI 숲의 0,1,2번 나무, IOI 숲의 0,1,2,3번 나무, KOI 숲의 3,4번 나무 순서로 구성한다. 이 때 뗏목을 이루는 나무들의 높이는 왼쪽부터 $H = [3, 3, 1, 3, 5, 7, 6, 6, 1]$ 이 된다. $\min(H[4], H[5], H[6], H[7]) = 5$ 이므로 $5 \times 4 = 20$ 의 안정성을 가진다.

따라서 함수는 $[12, 20]$ 를 반환하여야 한다.

예제 2

그레이더는 다음과 같이 함수를 호출한다.

```
max_stability([9, 9, 9, 9, 9], [1, 2, 3, 4, 5, 6, 7, 8], [0, 0, 0, 0, 0, 0, 0, 0], [0, 1, 2, 3, 4, 5, 6, 7])
```

함수는 $[45, 45, 45, 45, 45, 45, 45, 49]$ 을 반환하여야 한다.

예제 3

그레이더는 다음과 같이 함수를 호출한다.

```
max_stability([1000000000, 1000000000, 1000000000], [1000000000, 1000000000], [0], [1])
```

함수는 $[5000000000]$ 을 반환하여야 한다.

Sample grader

Sample grader는 아래와 같은 형식으로 입력을 받는다.

- Line 1: $N M Q$
- Line 2: $A[0] A[1] \dots A[N-1]$
- Line 3: $B[0] B[1] \dots B[M-1]$
- Line $4 + k$ ($0 \leq k \leq Q-1$): $L[k] R[k]$

Sample grader는 다음을 출력한다.

- Line $1 + k$ ($0 \leq k \leq Q - 1$): `max_stability`가 반환한 정수 배열을 C 라 할 때 $C[k]$ 의 값

Sample grader는 실제 채점에서 사용하는 그레이더와 다를 수 있음에 유의하라.