평균 최대화 (average)

양의 정수로 구성된 길이가 m ($m \geq 2$)인 수열 $x[0]$, $\cdots$, $x[m-1]$이 **막힌 수열**이라는 것은, 이 수열이 아래 조건을 만족한다는 것을 의미한다:

- 1 이상 $m-2$ 이하의 모든 정수 k에 대해, $x[k] > x[0]$이고 $x[k] > x[m-1]$이다.

즉, x의 양 끝 원소가 그 사이에 위치한 모든 원소보다 작다면 x는 막힌 수열이다.

예를 들어 $[3, 7, 8, 4, 2]$과 $[7, 7]$은 막힌 수열이지만, $[5, 8, 4, 6, 7]$와 $[3, 3, 4]$은 막힌 수열이 아니다. 정의에 의해 길이가 2인 모든 수열은 막힌 수열이고, 길이가 1 이하인 수열은 막힌 수열일 수 없다는 점에 유의하라.

길이가 K인 수열 $X[0], \cdots, X[K-1]$이 있을 때, ***들어내기*** 연산은 $X[i], \cdots, X[j]$가 *막힌 수열*인 (i, j)을 골라, 수열에서 $X[i+1], \cdots, X[j-1]$을 제거하는 (즉, 수열을 $X[0], \cdots, X[i], X[j], \cdots, X[K-1]$으로 바꾸는) 연산이다.

$f(X)$를 이러한 들어내기 연산을 원하는 대로 사용하여(사용하지 않을 수도 있고, 여러 번 사용할 수도 있음) 만들 수 있는 최종 수열의 평균의 최댓값이라고 정의하자.

예를 들어, $f([1, 3, 2, 100, 97, 98, 2, 3, 4, 1]) = 43$이며, 들어내기 연산을 아래와 같이 적용하면 된다.

- $i = 0$, $j = 2$를 선택하여 수열을 $[1, 2, 100, 97, 98, 2, 3, 4, 1]$로 바꾼다.
- $i = 5$, $j = 8$을 선택하여 수열을 $[1, 2, 100, 97, 98, 2, 1]$로 바꾼다.
- 최종 수열은 $[1, 2, 100, 97, 98, 2, 1]$이며, 이 수열의 평균은 $(1 + 2 + 100 + 97 + 98 + 2 + 1)/7 = 43$이다.

양의 정수로 구성된 길이가 N인 수열 $A[0], \cdots, A[N-1]$이 주어진다. 여러분은 $A[i], \cdots, A[j]$가 막힌 수열이 되도록 하는 순서쌍 (i, j)가 주어질 때마다, $f(A[i], \cdots, A[j])$의 값을 구하는 프로그램을 작성해야 한다.

참고

길이가 m인 수열 $x[0], \cdots, x[m-1]$의 **평균**은 수열의 원소의 합을 수열의 길이 m으로 나눈 값, 즉 $\frac{x[0]+x[1]+\cdots+x[m-1]}{m}$으로 정의한다.

수열 x에 대해 $x[i], \cdots, x[j]$는 수열 x의 i번 원소부터 j번 원소까지로 구성된 길이가 $j-i+1$인 수열이다. 예를 들어 $x = [3, 5, 7, 2, 9]$일 때, $x[1], \cdots, x[3]$은 $[5, 7, 2]$이고, $x[4], \cdots, x[4]$는 $[9]$이다.

함수 목록 및 정의

여러분은 아래 함수들을 구현해야 한다.

```
void initialize(std::vector<int> A)
```

- 이 함수는 단 한 번만 호출되며, 다른 모든 함수가 호출되기 전에 호출된다.
- A: 크기가 N인 정수 배열.
- 이후 함수 호출에 필요한 전처리나 전역 변수 설정이 있다면, 이 함수에 구현하면 된다.

```
std::array<long long, 2> maximum_average(int i, int j)
```

- $0 \le i < j \le N-1$이고, $A[i], \cdots, A[j]$가 *막힌 수열*이도록 주어진다.
- 이 함수는 $f(A[i], \cdots, A[j]) = s/t$일 때, $[s, t]$를 반환해야 한다.
  - s와 t는 1 이상 10^{18} 이하의 정수여야 한다. 제약 조건을 만족하는 모든 입력에 대해, 정답을 항상 해당 형태의 분수로 표현할 수 있음을 증명할 수 있다.
  - s/t은 기약분수가 아니어도 무방하다.
- 이 함수는 Q번 호출된다.

제출하는 소스 코드의 어느 부분에서도 입출력 함수를 실행해서는 안 된다.

제약 조건

- $2 \le N \le 300\,000$
- $1 \le Q \le 600\,000$
- 모든 i에 대해 $1 \le A[i] \le 10\,000\,000$ $(0 \le i \le N-1)$
- 모든 `maximum_average` 호출에 대해 $0 \le i < j \le N-1$이고, $A[i], \cdots, A[j]$가 *막힌 수열*이다.

부분문제

1. (5점)
   - $N \le 15$
2. (6점)
   - $N \le 50$
3. (13점)
   - $N \le 250$
4. (7점)
   - 모든 i에 대해 $A[i] \le 4$ $(0 \le i \le N-1)$
5. (12점)
   - $N \le 5\,000$
6. (17점)
   - A는 막힌 수열이다.
   - $Q = 1$이며, `maximum_average`$(0, N-1)$가 호출된다. 즉, 수열 A 전체에 대한 답만 구하면 충분하다.
7. (8점)
   - 모든 i에 대해 $A[i] \le 20$ $(0 \le i \le N-1)$
8. (32점)
   - 추가적인 제약 조건이 없다.

예제 1

$N = 10,\ A = [2, 4, 3, 9, 9, 9, 9, 9, 9, 1]$인 경우를 생각해 보자.

그레이더는 다음 함수들을 순서대로 호출한다.

```
initialize([2, 4, 3, 9, 9, 9, 9, 9, 9, 1])
maximum_average(0, 2)
maximum_average(0, 9)
```

$A[0], \ldots, A[2] = [2, 4, 3]$는 들어내기 연산을 통해 초기 상태보다 평균을 크게 만드는 것이 불가능하며, 따라서 최대 평균은 $(2+4+3)/3 = 9/3$이다. 따라서, `maximum_average(0, 2)`는 $[3, 1]$이나 $[9, 3]$ 등을 반환할 수 있다.

반면, $A[0], \ldots, A[9] = [2, 4, 3, 9, 9, 9, 9, 9, 9, 1]$에서 $i = 0, j = 2$ 를 선택하여 4를 제거하면 평균이 $64/10$에서 $60/9$로 증가하며, 이것이 가능한 최대 평균이다. 따라서, `maximum_average(0, 9)`는 $[60, 9]$나 $[20, 3]$ 등을 반환할 수 있다.

Sample grader

Sample grader는 아래와 같은 형식으로 입력을 받는다.

- Line 1: N
- Line 2: $A[0]\ A[1]\ \cdots\ A[N-1]$
- Line 3: Q
- Line $3+k$ $(1 \le k \le Q)$: $i[k]\ j[k]$ (k번째로 호출되는 `maximum_average` 함수의 인자)

Sample grader는 다음을 출력한다.

- Line k: k번째로 호출된 `maximum_average`가 반환한 array의 두 정수

Sample grader는 실제 채점에서 사용하는 그레이더와 다를 수 있음에 유의하라.