

식물 비교 (plants)

식물학자 헤이즐은 싱가포르 식물원의 특별한 전시회를 방문했다. 전시회에서 서로 다른 키의 n 개 식물들이 원 상에 놓여있다. 이 식물들은 시계 방향으로 0부터 $n - 1$ 로 나타내고, 식물 $n - 1$ 다음에 식물 0이 놓여있다.

각 식물 i ($0 \leq i \leq n - 1$)에 대해서, 헤이즐은 식물 i 와 시계 방향으로 다음 $k - 1$ 개의 식물 각각을 비교했고, 이 $k - 1$ 개 식물 중 식물 i 보다 더 큰 식물의 개수를 나타내는 숫자 $r[i]$ 를 적었다. 그래서 각 $r[i]$ 는 어떤 연속적인 k 개의 식물들의 상대적인 키에 따라 결정된다.

예를 들어, $n = 5$, $k = 3$, $i = 3$ 이라고 하자. 식물 $i = 3$ 로 부터 시계 방향으로 $k - 1 = 2$ 개의 식물들은 식물 4와 0이다. 만약 식물 4가 식물 3보다 크고 식물 0이 식물 3보다 작다면, 헤이즐은 $r[3] = 1$ 로 적는다.

헤이즐이 $r[i]$ 들을 정확히 기록했다고 가정한다. 따라서 이 숫자들과 일치하는 서로 다른 키의 식물들의 배치 형태는 적어도 하나 존재한다.

여러분은 q 개 식물 쌍들의 키를 비교해 달라는 요청을 받았다. 불행히도, 여러분은 전시회에 접근하지 못한다. 여러분의 유일한 정보는 헤이즐의 노트북에 기록된 k 와 수열 $r[0], \dots, r[n - 1]$ 이다.

비교해야하는 각각의 서로 다른 두 식물 x 와 y 의 쌍에 대해서, 여러분은 다음 세 가지 상황 중 어떤 일이 일어났는지 결정해야 한다:

- 식물 x 는 식물 y 보다 확실히 크다: 배열 r 과 일치하는 서로 다른 키 $h[0], \dots, h[n - 1]$ 의 임의의 배치에 대해서, $h[x] > h[y]$ 을 만족한다.
- 식물 x 는 식물 y 보다 확실히 작다: 배열 r 과 일치하는 서로 다른 키 $h[0], \dots, h[n - 1]$ 의 임의의 배치에 대해서, $h[x] < h[y]$ 을 만족한다.
- 비교 불가하다: 이전 두 경우가 모두 성립하지 않는다.

Implementation details

여러분은 다음 프로시저를 구현해야 한다:

```
void init(int k, int[] r)
```

- k : $r[i]$ 들을 결정하는 연속적인 식물들의 수
- r : 크기 n 의 배열, 여기서, $r[i]$ 는 시계방향으로 다음 $k - 1$ 개 식물들 중 식물 i 보다 더 큰 식물들의 개수이다.
- 이 프로시저는 `compare_plants`가 호출되기 전에 정확히 한 번 호출된다.

```
int compare_plants(int x, int y)
```

- x, y : 비교되는 두 식물
- 이 프로시저는 다음을 리턴해야 한다:
 - 식물 x 가 식물 y 보다 확실히 크면, 1.
 - 식물 x 가 식물 y 보다 확실히 작으면, -1.
 - 비교 불가하면, 0.
- 이 프로시저는 정확히 q 번 호출된다.

Examples

Example 1

다음 호출을 생각해보자:

```
init(3, [0, 1, 1, 2])
```

그레이더가 `compare_plants(0, 2)`를 호출한다고 하자. $r[0] = 0$ 이기 때문에 우리는 식물 2가 식물 0보다 크지 않다고 추측할 수 있다. 따라서 이 호출은 1을 리턴해야 한다.

다음으로 그레이더가 `compare_plants(1, 2)`를 호출한다고 하자. 위 조건들은 만족하는 모든 가능한 키들의 배치에서, 식물 1은 식물 2보다 작다. 따라서 이 호출은 -1을 리턴해야 한다.

Example 2

다음 호출을 생각해보자:

```
init(2, [0, 1, 0, 1])
```

그레이더가 `compare_plants(0, 3)`를 호출한다고 하자. $r[3] = 1$ 이기 때문에 우리는 식물 0이 식물 3보다 크다는 것을 안다. 따라서 이 호출은 1을 리턴해야 한다.

다음으로 그레이더가 `compare_plants(1, 3)`를 호출한다고 하자. 키의 두 배치 $[3, 1, 4, 2]$ 과 $[3, 2, 4, 1]$ 는 모두 헤이즐의 측정과 일치한다. 식물 1은 식물 3보다 한 배치에서는 작고, 다른 배치에서는 더 크기 때문에 이 호출은 0을 리턴해야 한다.

Constraints

- $2 \leq k \leq n \leq 200\,000$
- $1 \leq q \leq 200\,000$
- $0 \leq r[i] \leq k - 1$ (for all $0 \leq i \leq n - 1$)
- $0 \leq x < y \leq n - 1$
- 배열 r 과 일치하는 식물들의 서로 다른 키의 하나 이상의 배치 형태가 존재한다.

Subtasks

1. (5 points) $k = 2$
2. (14 points) $n \leq 5000, 2 \cdot k > n$
3. (13 points) $2 \cdot k > n$
4. (17 points) `compare_plants`의 각 호출의 정확한 답은 1 혹은 -1 .
5. (11 points) $n \leq 300, q \leq \frac{n \cdot (n-1)}{2}$
6. (15 points) `compare_plants`의 각 호출에 대해서, $x = 0$.
7. (25 points) 추가적 제약이 없음

Sample grader

샘플 그레이더는 다음 형식으로 입력을 읽는다:

- line 1: $n \ k \ q$
- line 2: $r[0] \ r[1] \ \dots \ r[n-1]$
- line $3 + i$ ($0 \leq i \leq q - 1$): `compare_plants`의 i 번째 호출에 대한 $x \ y$

샘플 그레이더는 다음 형식으로 답을 출력한다:

- line $1 + i$ ($0 \leq i \leq q - 1$): `compare_plants`의 i 번째 호출의 리턴 값