



## 植物の比較 (plants)

Hazel は植物学者であり、シンガポール植物園で開催される植物展に来ている。この植物展では、互いに異なる高さの  $n$  本の植物が円形に並べられる。これらの植物は時計回りに  $0$  から  $n - 1$  までの番号が付けられており、特に植物  $n - 1$  と植物  $0$  は隣り合っている。

Hazel は、それぞれの植物  $i$  ( $0 \leq i \leq n - 1$ ) に対して、これに時計回りに続く  $k - 1$  本の植物と比較し、これら  $k - 1$  本の植物のうち植物  $i$  より高いものの数を  $r[i]$  として記録した。従って、各  $r[i]$  の値は、連続した  $k$  本の植物の相対的な高さによって決まる。

例えば、 $n = 5$ ,  $k = 3$ ,  $i = 3$  のとき、植物  $3$  に時計回りに続く  $k - 1 = 2$  本の植物とは、植物  $4$  と植物  $0$  である。植物  $4$  が植物  $3$  より高く、植物  $0$  が植物  $3$  より低い場合には、Hazel は  $r[3] = 1$  と記録する。

Hazel は  $r[i]$  の値を正しく記録したとする。従って、これに矛盾しない植物の高さの状態が少なくとも  $1$  つ存在する。

あなたは  $q$  組の植物のペアについて、そのどちらが高いかを尋ねられた。しかし、あなたは植物展に参加することはできないため、Hazel が記録した  $k$  と  $r[0], \dots, r[n - 1]$  の値だけが情報源である。

それぞれの比較される植物のペア (植物  $x$  と植物  $y$  とする) に対し、以下の  $3$  つの状態のうちどれに当てはまるかを決定せよ。

- 植物  $x$  は植物  $y$  より必ず高い:  $r[i]$  の情報に矛盾しないどのような高さの状態  $h[0], \dots, h[n - 1]$  に対しても、 $h[x] > h[y]$  である。
- 植物  $x$  は植物  $y$  より必ず低い:  $r[i]$  の情報に矛盾しないどのような高さの状態  $h[0], \dots, h[n - 1]$  に対しても、 $h[x] < h[y]$  である。
- 結論づけることはできない: 上の  $2$  つのどちらでもない。

## 実装の詳細

あなたは以下のプロシーダを実装しなさい。

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

- $k$ : それぞれの  $r[i]$  の値を決定するために用いられる連続した植物の数。
- $r$ : 長さ  $n$  の配列であり、 $r[i]$  は、植物  $i$  に時計回りに続く  $k - 1$  本の植物のうち植物  $i$  より高いものの数である。
- このプロシーダは、`compare_plants` のいかなる呼び出しよりも前にちょうど  $1$  回だけ呼び出される。

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

- $x, y$ : 比較される植物の番号.
- このプロシージャは次の戻り値を返なければならない:
  - 1: 植物  $x$  が植物  $y$  より必ず高い場合.
  - -1: 植物  $x$  が植物  $y$  より必ず低い場合.
  - 0: 結論づけることができない場合.
- このプロシージャはちょうど  $q$  回呼び出される.

## 入出力例

### 入出力例 1

次の呼び出しを考える.

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

採点プログラムが `compare_plants(0, 2)` を呼び出したとしよう.  $r[0] = 0$  であることからただちに、植物 2 は植物 0 より高くないことが分かる. 従って、1 を返さなければならない.

次に、採点プログラムが `compare_plants(1, 2)` を呼び出したとしよう. 制約を満たすどのような高さの状態についても、植物 1 は植物 2 より低い. 従って、-1 を返さなければならない.

### 入出力例 2

次の呼び出しを考える.

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

採点プログラムが `compare_plants(0, 3)` を呼び出したとしよう.  $r[3] = 1$  なので、植物 0 は植物 3 より高いことが分かる. 従って、1 を返さなければならない.

次に、採点プログラムが `compare_plants(1, 3)` を呼び出したとしよう. 高さの状態  $[3, 1, 4, 2]$  と  $[3, 2, 4, 1]$  はいずれも Hazel の記録に矛盾しない. ある状態においては、植物 1 は植物 3 よりも低いですが、また別の状態においては、植物 1 は植物 3 よりも高いので、この呼び出しにおいては 0 を返さなければいけない.

## 制約

- $2 \leq k \leq n \leq 200\,000$
- $1 \leq q \leq 200\,000$
- $0 \leq r[i] \leq k - 1$  ( $0 \leq i \leq n - 1$ )
- $0 \leq x < y \leq n - 1$

- 配列  $r$  の情報に矛盾しない, 互いに高さの異なる状態が少なくとも 1 つ存在する.

## 小課題

1. (5 点)  $k = 2$
2. (14 点)  $n \leq 5\,000, 2 \cdot k > n$
3. (13 点)  $2 \cdot k > n$
4. (17 点) それぞれの `compare_plants` の呼び出しに対する正しい答えは 1 か  $-1$  である.
5. (11 点)  $n \leq 300, q \leq \frac{n \cdot (n-1)}{2}$
6. (15 点) それぞれの `compare_plants` の呼び出しについて  $x = 0$ .
7. (25 点) 追加の制約は無い.

## 採点プログラムのサンプル

採点プログラムは以下の形式で入力を読み込む.

- 1 行目:  $n \ k \ q$
- 2 行目:  $r[0] \ r[1] \ \dots \ r[n-1]$
- $3 + i$  行目 ( $0 \leq i \leq q - 1$ ):  $x \ y$  これは  $i$  回目の `compare_plants` の呼び出しに用いられる.

採点プログラムは以下の形式であなたの答えを出力する.

- $1 + i$  行目 ( $0 \leq i \leq q - 1$ ):  $i$  回目の `compare_plants` の呼び出しの戻り値.