



Comparing Plants (plants)

Hazel the botanist visited a special exhibition in the Singapore Botanical Gardens. In this exhibition, n plants of **distinct heights**(එකිනෙකට වෙනස්) are placed in a circle. These plants are labelled from 0 to $n - 1$ in clockwise order, with plant $n - 1$ beside plant 0.

සෑම i ($0 \leq i \leq n - 1$) වෙතින් පැළයක්ම, ඊට පසු clockwise(දක්ෂිණාවර්තව) ඇති පැළ, $k - 1$ ක් සමඟ සසඳනු ලැබේ. එම පැළ අතරින් i වෙතින් පැළයට වඩා උස පැළ ගණන $r[i]$ වේ. Thus, each value $r[i]$ depends on the relative heights of some k consecutive plants.

For example, suppose $n = 5$, $k = 3$ and $i = 3$. The next $k - 1 = 2$ plants in clockwise order from plant $i = 3$ would be plant 4 and plant 0. If plant 4 was taller than plant 3 and plant 0 was shorter than plant 3, Hazel would write down $r[3] = 1$.

Hazel නිවැරදිව $r[i]$ සොයා ඇත. එම නිසා දී ඇති $r[i]$ ට ගැලපෙන, distinct heights(වෙනස් උස ඇති) එක configuration(රටාවක්) හෝ පවතී.

You were asked to compare the heights of q pairs of plants. දැන් ඔබට පැළ රටාව දැක ගත නොහැක. ඔබට ඇත්තේ Hazel ලියා ඇති k හා $r[0], \dots, r[n - 1]$ අගයන් පමණි.

For each pair of different plants x and y that need to be compared, determine which of the three following situations occurs:

- Plant x is definitely taller than plant y : in any configuration of distinct heights $h[0], \dots, h[n - 1]$ consistent with the array r we have $h[x] > h[y]$.
- Plant x is definitely shorter than plant y : in any configuration of distinct heights $h[0], \dots, h[n - 1]$ consistent with the array r we have $h[x] < h[y]$.
- The comparison is inconclusive: neither of the previous two cases applies.

Implementation details

You should implement the following procedures:

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

- k : the number of consecutive plants whose heights determine each individual value $r[i]$.
- r : an array of size n , where $r[i]$ is the number of plants taller than plant i among the next $k - 1$ plants in clockwise order.
- This procedure is called exactly once, before any calls to `compare_plants`.

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

- x, y : labels of the plants to be compared.
- This procedure should return:
 - 1 if plant x is definitely taller than plant y ,
 - -1 if plant x is definitely shorter than plant y ,
 - 0 if the comparison is inconclusive.
- This procedure is called exactly q times.

Examples

Example 1

Consider the following call:

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

Let's say the grader calls `compare_plants(0, 2)`. Since $r[0] = 0$ we can immediately infer that plant 2 is not taller than plant 0. Therefore, the call should return 1.

Let's say the grader calls `compare_plants(1, 2)` next. For all possible configurations of heights that fit the constraints above, plant 1 is shorter than plant 2. Therefore, the call should return -1 .

Example 2

Consider the following call:

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

Let's say the grader calls `compare_plants(0, 3)`. Since $r[3] = 1$, we know that plant 0 is taller than plant 3. Therefore, the call should return 1.

Let's say the grader calls `compare_plants(1, 3)` next. Two configurations of heights $[3, 1, 4, 2]$ and $[3, 2, 4, 1]$ are both consistent with Hazel's measurements. Since plant 1 is shorter than plant 3 in one configuration and taller than plant 3 in the other, this call should return 0.

Constraints

- $2 \leq k \leq n \leq 200\,000$
- $1 \leq q \leq 200\,000$
- $0 \leq r[i] \leq k - 1$ (for all $0 \leq i \leq n - 1$)
- $0 \leq x < y \leq n - 1$
- There exists one or more configurations of **distinct heights** of plants consistent with the array

r .

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) The correct answer to each call of `compare_plants` is 1 or -1 .
5. (11 points) $n \leq 300, q \leq \frac{n \cdot (n-1)}{2}$
6. (15 points) $x = 0$ for each call of `compare_plants`.
7. (25 points) No additional constraints.

Sample grader

The sample grader reads the input in the following format:

- line 1: $n \ k \ q$
- line 2: $r[0] \ r[1] \ \dots \ r[n-1]$
- line $3 + i$ ($0 \leq i \leq q-1$): $x \ y$ for the i -th call to `compare_plants`

The sample grader prints your answers in the following format:

- line $1 + i$ ($0 \leq i \leq q-1$): return value of the i -th call to `compare_plants`.