

2026 Canadian Computing Olympiad
Day 1, Problem 3
Beyond Counting

Time Limit: 5 seconds

Memory Limit: 1024 MB

Andy Jiang is studying data structures. One day, his friend Austin Zhu gave him a task on trees.

Austin provided a tree with N vertices, numbered from 1 to N . Each vertex i has a value A_i .

For each query, Austin asked Andy to consider a path between two vertices s_i and t_i , and compute how many times a given value x_i appears on that path.

Andy glanced at the problem and thought that this task was too easy for him.

Instead of just counting occurrences, Andy decided to challenge himself further. For each query, he wants to know how the frequency of x_i compares to other values on the same path.

Formally, for each query (s_i, t_i, x_i) :

- Consider the simple path from s_i to t_i .
- Let $\text{cnt}(y)$ be the number of occurrences of value y on this path.

Andy defines the *rank* of x_i as

$$1 + |\{y \mid \text{cnt}(y) > \text{cnt}(x_i)\}|.$$

That is, one plus the number of distinct values that appear more frequently than x_i on the path. Note that it is possible the value x_i does not appear on the path, i.e. $\text{cnt}(x_i) = 0$. In this case, you should return 1 plus the number of distinct values on the path.

In some test cases, the queries are given in an encoded form as described below.

Help Andy compute the rank of x_i for each query.

Input Specification

The first line contains three positive integers N , Q , and T ($1 \leq N, Q \leq 10^5$, $T \in \{0, 1\}$).

The second line contains N integers A_1, A_2, \dots, A_N ($1 \leq A_i \leq 10^9$).

The next $N - 1$ lines each contain two integers u_i, v_i ($1 \leq u_i, v_i \leq N$), representing the i -th edge.

Each of the next Q lines contains three integers $\hat{s}_i, \hat{t}_i, \hat{x}_i$ ($1 \leq \hat{s}_i, \hat{t}_i \leq N$, $1 \leq \hat{x}_i \leq 10^9$), describing the i -th query.

Let $\text{last}_0 = 0$. For each query $i = 1, 2, \dots, Q$, the actual parameters are defined as:

$$\begin{aligned} s_i &= ((\hat{s}_i + \text{last}_{i-1} \times T - 1) \bmod N) + 1, \\ t_i &= ((\hat{t}_i + \text{last}_{i-1} \times T - 1) \bmod N) + 1, \\ x_i &= ((\hat{x}_i + \text{last}_{i-1} \times T - 1) \bmod 10^9) + 1. \end{aligned}$$

After computing the answer to the i -th query, set

$$\text{last}_i = \text{answer to the } i\text{-th query.}$$

It may also be useful to note that “mod” corresponds to the % operator in most programming languages, indicating the remainder after division. For example, $5 \bmod 3 = 2$ and $17 \bmod 4 = 1$.

The following table shows how the 25 available marks are distributed:

| Marks Awarded | Bounds on N, Q | Bounds on T | Additional Constraints |
|---------------|-------------------------|---------------|-------------------------------|
| 1 mark | $1 \leq N, Q \leq 10^3$ | $T = 1$ | None. |
| 1 mark | $1 \leq N, Q \leq 10^5$ | $T = 0$ | All s_i are equal. |
| 4 marks | | $T = 1$ | |
| 4 marks | | $T = 0$ | $u_i = i$ and $v_i = i + 1$. |
| 5 marks | | $T = 1$ | |
| 3 marks | | $T = 0$ | |
| 7 marks | | $T = 1$ | None. |

Output Specification

For each query, output the answer to the query on a new line.

Sample Input 1

```
5 5 0
1 2 3 4 4
4 3
2 5
1 3
3 2
4 5 3
4 5 4
4 5 5
1 5 1
1 5 4
```

Output for Sample Input 1

```
2
1
4
1
1
```

Sample Input 2

```
5 5 1
1 2 3 4 4
4 3
2 5
1 3
3 2
4 5 3
2 3 2
3 4 4
2 1 999999997
5 4 3
```

Output for Sample Input 2

```
2
1
4
1
1
```