☰   🌸 Hello, **2024101067**.

Dominator Tree

Submit solution

All submissions
Best submissions

✔ **Points:** 100 (partial)
🕐 **Time limit:** 1.0s
🖥 **Memory limit:** 256M

❯ **Problem type**

❮ **Allowed languages**
   C, C++

You are given a tree with n vertices. Each vertex i has an integer value a_i ($1 \le a_i \le n$) written on it.

For each integer i from 1 to n, determine whether there exists a **non-trivial simple path** in the tree such that the integer i is the **majority** value among the values on the vertices in that path.

A value x is considered the **majority** in a sequence of k values if it appears **strictly more than $\lfloor k / 2 \rfloor$ times**. For eg, the sequence [2, 1, 1, 2, 1] has a majority 1 because it appears $3 > \lfloor 5 / 2 \rfloor = 2$ times, but [1, 2, 1, 2] and [2, 3, 4, 5] do not have a majority.

A sequence of vertices $v_1, v_2, ..., v_m$ ($m \ge 2$) forms a **non-trivial simple path** if:

- For every $1 \le i \le m - 1$, there is an edge between v_i and v_{i+1}, and
- All vertices in the sequence are **pairwise distinct**.

Note: The path must contain **at least 2 vertices**.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases.

Each test case consists of:

- An integer n ($2 \le n \le 5 * 10^5$) — the number of vertices in the tree.
- A line with n integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le n$) — the values on the vertices.
- $n - 1$ lines, each containing two integers u and v ($1 \le u, v \le n, u \ne v$), denoting an edge between vertex u and vertex v.

It is guaranteed that:

☰   🌸 Hello, **2024101067**.

Output

For each test case, output a binary string `s` of length `n`, where:

- `s[i] = '1'` if there exists a non-trivial simple path in which the value `i` is the majority.
- `s[i] = '0'` otherwise.

Each string should be printed on a new line.

Sample Input

```
2
3
1 2 3
1 3
2 3
4
3 1 1 3
1 2
2 3
4 2
```
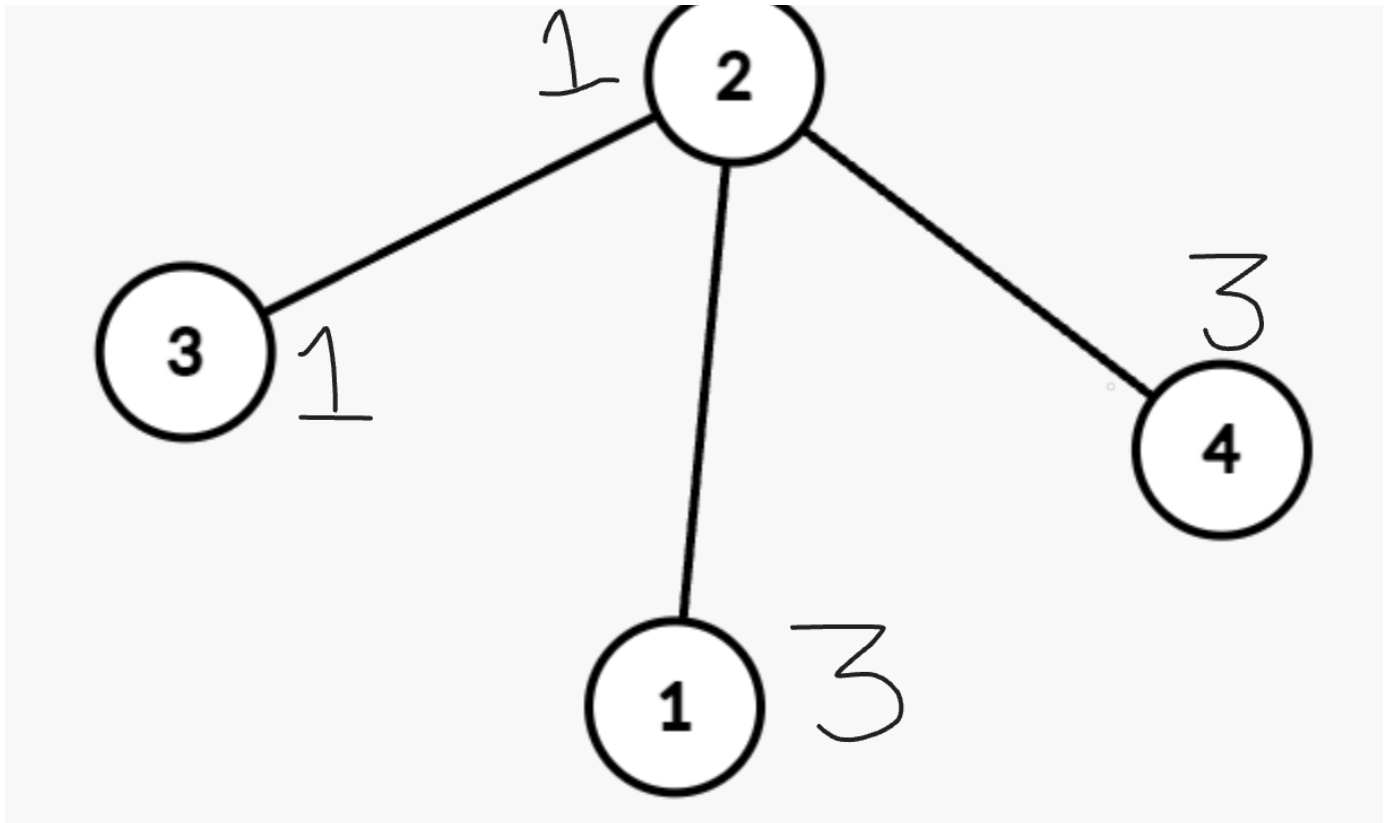
Copy

Sample Output

```
000
1010
```

Copy

Sample Explanation

For testcase 2 we have the following tree

The labels inside the node are node numbers, and the number drawn outside the node is the value of the node `a[i]`

The path 1->2->3 has a majority of 1, while the path 1->2->4 has a majority of 3. It can be seen that there exists no non-trivial simple path where the majority element is 2 or 4.

❓ Clarifications

Request clarification

No clarifications have been made at this time.