F. Skibidus and Slay

time limit per test: 4 seconds
memory limit per test: 512 megabytes

Let's define the *majority* of a sequence of k elements as the unique value that appears strictly more than $\left\lfloor \frac{k}{2} \right\rfloor$ times. If such a value does not exist, then the sequence does **not** have a majority. For example, the sequence $[1, 3, 2, 3, 3]$ has a majority 3 because it appears $3 > \left\lfloor \frac{5}{2} \right\rfloor = 2$ times, but $[1, 2, 3, 4, 5]$ and $[1, 3, 2, 3, 4]$ do not have a majority.

Skibidus found a tree* of n vertices and an array a of length n. Vertex i has the value a_i written on it, where a_i is an integer in the range $[1, n]$.

For each i from 1 to n, please determine if there exists a non-trivial simple path† such that i is the *majority* of the **sequence of integers written on the vertices** that form the path.

*A tree is a connected graph without cycles.

†A sequence of vertices $v_1, v_2, \ldots, v_m$ ($m \geq 2$) forms a non-trivial simple path if v_i and v_{i+1} are connected by an edge for all $1 \leq i \leq m-1$ and all v_i are pairwise distinct. **Note that the path must consist of at least 2 vertices.**

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n ($2 \leq n \leq 5 \cdot 10^5$) — the number of vertices.

The second line of each test case contains $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq n$) — the integers written on the vertices.

Each of the next $n-1$ lines contains two integers u_i and v_i, denoting the two vertices connected by an edge ($1 \leq u_i, v_i \leq n, u_i \neq v_i$).

It is guaranteed that the given edges form a tree.

It is guaranteed that the sum of n over all test cases does not exceed $5 \cdot 10^5$.

Output

For each test case, output a binary string s of length n on a separate line. s_i should be computed as follows:

- If there is a non-trivial path containing i as the majority, s_i is '1';
- Otherwise, s_i is '0'.

Example

input Copy

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

```
2 3
3 4
13
1 4 4 7 4 7 1 1 7 11 11 11 11
1 2
2 3
3 4
4 5
4 6
2 7
7 8
2 9
6 10
5 11
11 12
10 13
```

output Copy

```
000
1010
0001
1001001000100
```

Note

In the first test case, there is no non-trivial path with 1, 2, or 3 as a majority, so the binary string outputted is "000".

In the second test case, $1 \to 2 \to 4$ is a non-trivial path with 3 as a majority.