

Alice and Bob are playing a game, defined below:

- There is an undirected tree graph with n nodes that has the following properties:
 - Each node has c_i golden coins.
 - Node **1** is root of the tree.
 - The parent node of some node u is defined as $p(u)$.
- Moves
 - Players move in turns.
 - During a move, a player can select a node $u > 1$ and move one or more coins to $p(u)$.
 - If the current player can't make any move, they lose the game.

The game quickly becomes boring because the result is determined by the tree's configuration and the number of coins in each node (assuming that both players play optimally).

Alice decides to instead challenge Bob by asking him q questions. For each question i :

1. Alice picks a node $u_i > 1$ and *removes* the edge between u_i and $p(u_i)$.
2. She picks another node v and draws a new undirected edge between u_i and v_i . So now $p(u_i) = v_i$.

Bob must determine if the first player has a winning strategy for the new tree or not. It's possible that after Alice draws the new edge, the graph will no longer be a tree; if that happens, the question is *invalid*. Each question is independent, so the answer depends on the initial state of the graph (and not on previous questions).

Given the tree and the number of coins in each node, can you help Bob answer all q questions?

Input Format

The first line contains an integer, n (the number of nodes).

The second line contains n space-separated integers, c_1, c_2, \dots, c_n , describing the number of coins in each node.

Each of the $n - 1$ subsequent lines contains **2** space-separated integers denoting an undirected edge between nodes a and b , respectively.

The next line contains an integer, q (the number of questions Alice asks).

Each of the q subsequent lines contains **2** space-separated integers, u_i and v_i , respectively.

Constraints

- $1 \leq n, q \leq 5 \times 10^4$
- $1 \leq a, b \leq n$
- $0 \leq c_i \leq 20$

For each question:

- $2 \leq u_i \leq n$
- $1 \leq v_i \leq n$
- $u_i \neq v_i$

Output Format

On a new line for each question, print **YES** if the first player has a winning strategy, print **NO** if they do not, or print **INVALID** if the question is not valid.

Sample Input

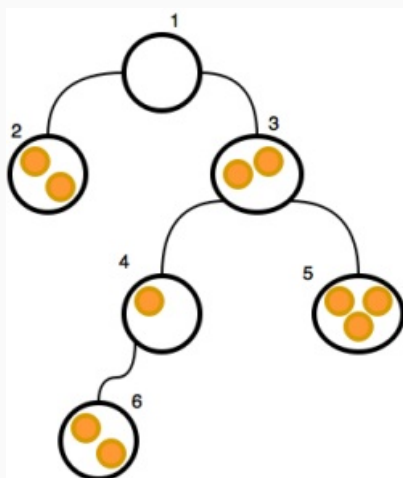
```
6
0 2 2 1 3 2
1 2
1 3
3 4
3 5
4 6
3
```

6 2
4 1
3 6

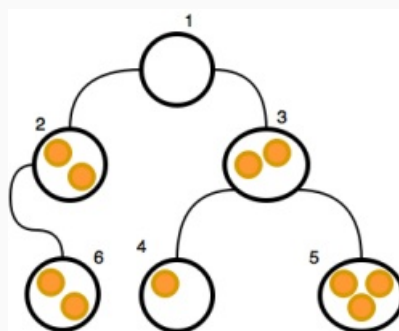
Sample Output

NO
YES
INVALID

Explanation

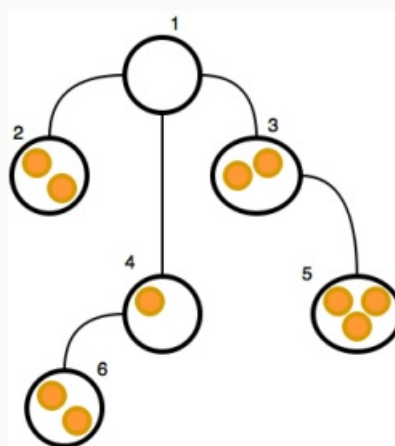


Initially the tree looks like this:



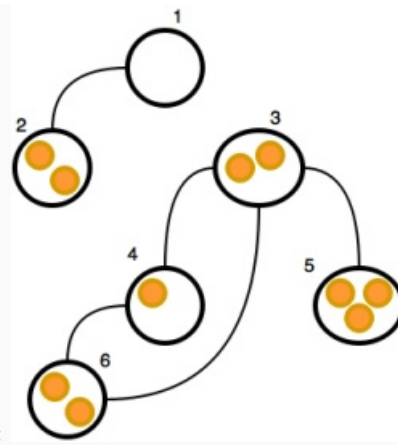
After the first question (6 2), the tree looks like this:
edge connecting node 6 to 4 and makes 2 the new parent node of 6. Because this configuration does not result in a winning strategy, we print **NO** on a new line.

Alice removes the



After the second question (4 1), the tree looks like this:
the edge connecting node 4 to 3 and makes 1 the new parent node of 4. Because this configuration results in a winning strategy, we print **YES** on a new line.

Alice removes



After the third question (**3 6**), the graph is no longer a tree: Alice removes the edge connecting node **3** to **1** and makes **6** the new parent node of **3**. The graph is now partitioned into two separate subgraphs (one of which is also not a tree); because the game must be played on a single undirected tree graph, we print **INVALID** on a new line.