7534 The Parenthesis Tree

Samosa Bhai and Jalebi Bai are taking cooking classes from CodeChef. CodeChef, as the name suggests, is also a coding enthusiast. So he decides to get his pupils interested in coding. For this, he takes them both to a strange tree made up of N nodes. Each of these nodes is of one of the following two types — nodes containing open parenthesis '(' and nodes containing closed parenthesis ')'. CodeChef asks the students Q queries, where in each query they have to find out if the path between two given nodes is a balanced parentheses string or not. If they solve all the queries, they will get to eat the special Christmas cake made by CodeChef.

Samosa Bhai and Jalebi Bai are lazy kids, but they also want to eat the cake. So they ask you for help.

Note: A balanced parentheses string means that each opening parenthesis has a corresponding closing parenthesis and the pairs of parentheses are properly nested.

Input

The first line contains T, the number of test cases to follow.

Each test case begins with N and Q, the number of nodes in the tree and the number of queries to follow.

$N-1$ lines follow. Each line contains 2 space-separated integers, x and y, which denotes that there is an edge between them.

The next line contains N space-separated parentheses. The i-th parenthesis denotes the value on the i-th node.

The Q queries follow. Each query contains two space-separated integers u and v.

Output

For each query, output '`Yes`' [without quotes] if the path from u to v is a balanced parentheses string and '`No`' [without quotes] if the path from u to v is not a balanced parentheses string.

Constraints:

- $1 \leq N \leq 10^5$
- $1 \leq Q \leq 3 * 10^5$
- $1 \leq$ Sum of N over all cases $\leq 10^5$
- $1 \leq$ Sum of Q over all cases $\leq 3 * 10^5$

Explanation: In the sample case below

- **Query 1**: The path contains `(())`.
- **Query 2**: The path contains `))((`.
- **Query 3**: The path contains `()))`.
- **Query 4**: The path contains `())(`.

Sample Input

```
1
6 4
1 2
2 5
1 3
3 6
3 4
) ( ) ) ( (
5 3
3 5
2 4
6 2
```

Sample Output

```
Yes
No
No
No
```