B. Christmas Spruce

difficulty: 1200
time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Consider a rooted tree. A rooted tree has one special vertex called the root. All edges are directed from the root. Vertex u is called a *child* of vertex v and vertex v is called a *parent* of vertex u if there exists a directed edge from v to u. A vertex is called a *leaf* if it doesn't have children and has a parent.

Let's call a rooted tree a *spruce* if its every non-leaf vertex has at least 3 leaf children. You are given a rooted tree, check whether it's a spruce.

The definition of a rooted tree can be found here (https://en.wikipedia.org/wiki/Tree_(graph_theory)).

Input
The first line contains one integer n — the number of vertices in the tree $(3 \le n \le 1\ 000)$. Each of the next $n - 1$ lines contains one integer p_i $(1 \le i \le n - 1)$ — the index of the parent of the $i + 1 - th$ vertex $(1 \le p_i \le i)$.

Vertex 1 is the root. It's guaranteed that the root has at least 2 children.

Output
Print "Yes" if the tree is a spruce and "No" otherwise.

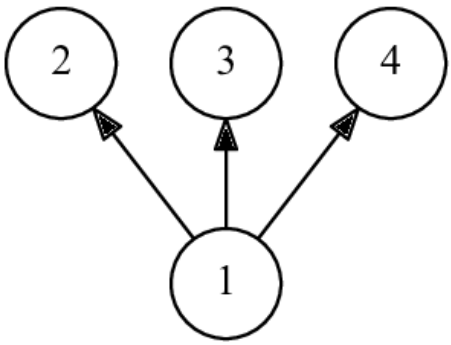
Examples
input
4
1
1
1
output
Yes

input
7
1
1
1
2
2
2
output
No

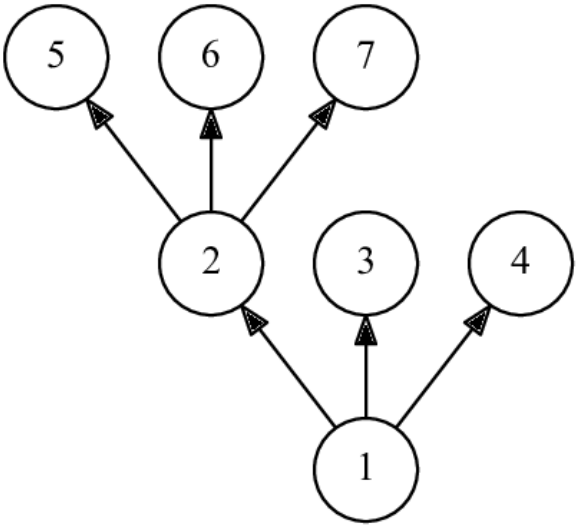
input
8
1
1
1
1
3
3

3
output
Yes

Note
The first example:

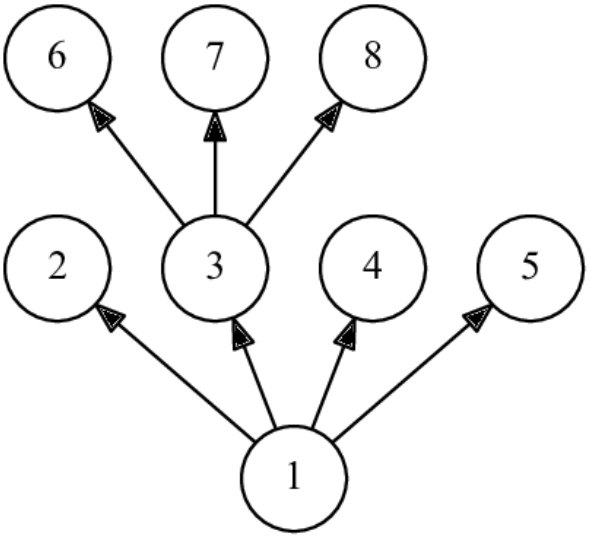


The second example:



It is not a spruce, because the non-leaf vertex 1 has only 2 leaf children.

The third example:



913B Christmas Spruce
implementation, trees
https://codeforces.com/problemset/problem/913/B
github.com/andy489