

Leaf Meeting

Input file: *standard input*
Output file: *standard output*
Time limit: 3 seconds
Memory limit: 1024 mebibytes

You are given an undirected tree with n vertices. A *leaf* is a vertex of degree 1. Let k be the number of leaves; initially, there is one token on each leaf, so there are k tokens in total.

For every token, you must choose an infinite sequence of leaves $a_1, a_2, a_3, \dots$ with $a_i \neq a_{i+1}$ for all $i \geq 1$, and with $a_1 \neq v$, where v is the leaf the token starts on. The sequences for different tokens can be chosen independently. The chosen sequence determines the route of the token: it first walks along the shortest path from v to a_1 , then along the shortest path from a_1 to a_2 , then from a_2 to a_3 , and so on indefinitely.

In one move, every token simultaneously traverses exactly one edge of its own route. No token may stay in place.

Determine whether the sequences can be chosen so that, after some finite number of moves, all k tokens are on the same leaf at the same time.

Input

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

The first line of each test case contains an integer n ($2 \leq n \leq 2 \cdot 10^5$): the number of vertices in the tree.

Each of the next $n - 1$ lines contains two integers u and v ($1 \leq u, v \leq n$): a bidirectional edge between vertices u and v . It is guaranteed that these edges form a tree.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print the answer on a separate line. If it is possible to choose the required sequences for all tokens so that they all gather in one leaf at the same time, print **YES**, otherwise print **NO**.

Example

<i>standard input</i>	<i>standard output</i>
5	NO
2	NO
1 2	NO
3	NO
2 1	YES
2 3	
4	
1 2	
1 3	
3 4	
6	
1 2	
2 3	
2 4	
4 5	
4 6	
8	
1 2	
1 3	
1 4	
1 5	
1 6	
1 7	
1 8	