

C Game: Adversarial Distance Oracle

TIME LIMIT: 2.0s
 MEMORY LIMIT: 1024MB

Alice and Bob are playing a game on a tree T with n vertices. Alice wants to identify a secret vertex x . However, Bob is adversarial and does not have to choose x in advance.

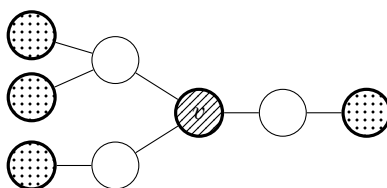
Initially, every vertex of T is considered a “possible” candidate for x . In each turn, Alice chooses a vertex v and asks for the distance from v to x . Bob responds with a non-negative integer d . Alice then removes every candidate vertex u such that $\text{dist}(u, v) \neq d$.

Bob’s answer must be consistent with at least one remaining candidate. In other words, after Alice removes all vertices u with $\text{dist}(u, v) \neq d$, the candidate set must still be non-empty. Subject to this rule, Bob chooses his answers to force Alice to use as many queries as possible.

Alice wins as soon as exactly one candidate vertex remains. Alice chooses her queries optimally to minimize the number of queries.

Given the structure of the tree, find the minimum number of queries Alice needs to guarantee a win, regardless of Bob’s strategy.

For example, if Alice queries a vertex v and Bob answers $d = 2$, then all vertices at distance exactly 2 from v remain possible, and all other vertices are removed.



Querying v and receiving answer 2: the hatched vertex is queried, dotted vertices remain possible, and plain vertices are removed.

INPUT

The first line contains one integer t ($1 \leq t \leq 10^5$), the number of test cases.

Each test case begins with one integer n ($2 \leq n \leq 2000$), the number of vertices.

Each of the next $n - 1$ lines contains two integers a_i and b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$), denoting an edge of the tree.

It is guaranteed that the edges of each test case form a tree and that $\sum n^2 \leq 2000^2$ over all test cases.

OUTPUT

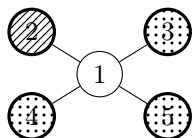
For each test case, print one integer: the minimum number of queries Alice needs in the worst case.

SAMPLES

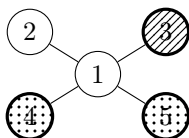
Sample input 1	Sample output 1
2 4 1 2 2 3 3 4 5 1 2 1 3 1 4 1 5	1 3

Explanation of sample 1.

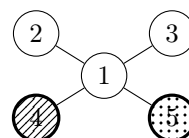
- In the first test case, the tree is a path on four vertices. Querying vertex 1 gives possible distances 0, 1, 2, 3, all distinct, so one query is enough.
- In the second test case, the tree is a star centered at vertex 1:
- In the following diagrams, the hatched vertex is the queried vertex, dotted vertices remain possible after Bob’s answer, and plain vertices have been removed.



Query 2, answer 2: vertices 3, 4, 5 remain possible.



Query 3, answer 2: vertices 4, 5 remain possible.



Query 4, answer 2: only vertex 5 remains.

- This shows that three queries are sufficient. Alice first queries leaf 2. If Bob answers 0 or 1, the secret is immediately determined; the worst answer is 2, leaving leaves 3, 4, 5. Then Alice queries leaf 3, and in the worst case leaves 4, 5 remain. One final query distinguishes them.
- Two queries are not enough. After the first query, Bob can keep at least three leaves possible. Among three leaves of a star, one more distance query cannot distinguish all of them, because at least two of the leaves have the same distance to the queried vertex.