☰   ✿ Hello, **2024101067**.

Hide and Seek

All submissions
Best submissions

✔ **Points:** 100 (partial)
⏱ **Time limit:** 2.0s
☰ **Memory limit:** 256M

✐ **Author:**
  2022101116

❯ **Problem type**

⌄ **Allowed languages**
  C, C++

Alice and Bob are playing a game of hide and seek in a forest. The forest paths form a **tree** - a connected undirected graph with n nodes and $n - 1$ edges. Each node represents a possible hiding spot, and each edge is a bidirectional trail between two spots.

Some spots are **leaf nodes**, meaning they have exactly one neighboring spot. These leaf nodes are closest to the forest's edge, and Bob always starts his search from one of them.

Alice wants to hide in the **safest** place - the one that is **as far as possible** from any leaf node, so Bob takes the longest time to reach her.

Your task is to help Alice choose the **best hiding spot**:

- the node that is **farthest from any leaf**, and
- if there are multiple such nodes, the one with the **smallest index** among them.

Input Format

- The first line contains an integer t - the number of test cases.
- For each test case:
  - The first line contains an integer n $(1 \leq n \leq 10^6)$- the number of hiding spots.
  - Each of the next $n - 1$ lines contains two integers u and v $(1 \leq u, v \leq n, u \neq v)$ - representing a trail between spot u and spot v.

It is guaranteed that the given structure forms a valid tree.

---

proudly powered by **DMOJ** | English (en) ⌄

≡   ✳ Hello, **2024101067**.

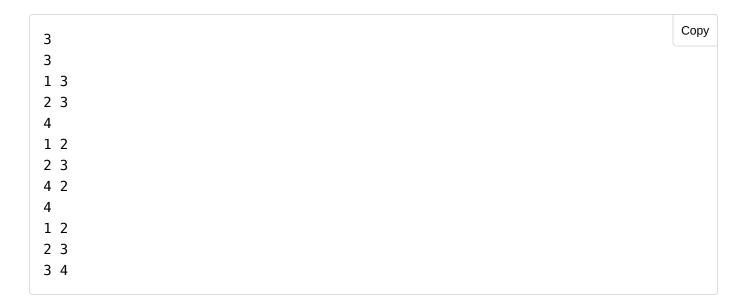For each test case, print a single integer - the index of the best hiding spot for Alice.
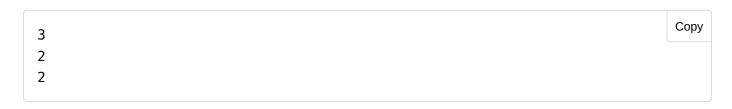

Constraints

$\sum n \le 10^3$ [40 Pts]

$\sum n \le 10^6$ [60 Pts]

Samples

Input

```
3
3
1 3
2 3
4
1 2
2 3
4 2
4
1 2
2 3
3 4
```

Copy

Output

```
3
2
2
```

Copy

Notes

- A **leaf node** is a node with exactly one neighbor.
- The **distance** between two nodes is the number of edges in the shortest path between them.

❓ Clarifications   Report an issue