

Food Fight

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Busy Beaver is causing chaos at the farmers' market again! This time, he is starting a food fight among the stalls.

The stalls are numbered from 1 to N and are connected by $N - 1$ roads, forming a tree. In other words, it is possible to travel from any stall to any other stall by following the roads, and there is exactly one simple path between any two stalls.

Busy Beaver assigns each stall to either Team Potato or Team Tomato as follows:

- He starts at stall 1, travels along the roads, visits every stall, and finally returns to stall 1. Among all such routes, he chooses one with the minimum possible total number of roads traversed.
- Stall 1 is assigned to Team Potato.
- Whenever Busy Beaver visits a stall for the first time (other than stall 1), he immediately assigns it to one of the two teams. To keep the fight fair, he alternates the team assignment each time he assigns a new stall. That is, if the most recently assigned stall was assigned to Team Potato, then the next newly visited stall must be assigned to Team Tomato, and vice versa.

Your task is to count the number of possible team assignments. Note that different minimum-length routes may produce the same final assignment; such assignments should be counted only once. Since the answer may be enormous, output it modulo $10^9 + 7$.

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 a single integer N ($2 \leq N \leq 10^5$).

Each of the next $N - 1$ lines contains two integers u and v ($1 \leq u, v \leq N$, $u \neq v$), indicating that there is a road between stalls u and v .

The sum of N across all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output a single integer: the number of possible final team assignments modulo $10^9 + 7$.

Scoring

There are two subtasks for this problem.

- (10 points): The stalls form a star graph rooted at stall 1. More specifically, stall 1 has $N - 1$ roads connected to it.
- (20 points): The stalls form a binary tree rooted at stall 1. More specifically, stall 1 has at most 2 roads connected to it, and every other stall has at most 3 roads connected to it.
- (70 points): No additional constraints.

Example

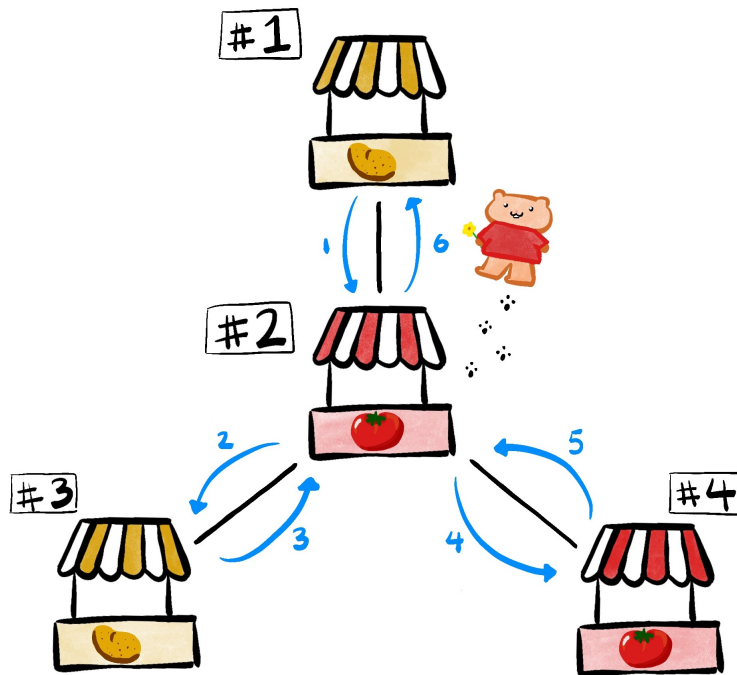
| standard input | standard output |
|----------------|-----------------|
| 4 | 2 |
| 4 | 20 |
| 1 2 | 8 |
| 2 3 | 12 |
| 2 4 | |
| 7 | |
| 1 2 | |
| 1 3 | |
| 1 4 | |
| 1 5 | |
| 1 6 | |
| 1 7 | |
| 7 | |
| 1 2 | |
| 1 3 | |
| 2 4 | |
| 2 5 | |
| 3 6 | |
| 3 7 | |
| 7 | |
| 1 2 | |
| 1 3 | |
| 1 4 | |
| 2 5 | |
| 2 6 | |
| 2 7 | |

Note

The sample contains 4 test cases:

- Test case 1 satisfies the second subtask (binary tree).
- Test case 2 satisfies the first subtask (star graph).
- Test case 3 satisfies the second subtask (binary tree).
- Test case 4 satisfies the third subtask (no additional constraints).

In the first test case, one possible team assignment is shown below.



One minimum-length route is:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 2 \rightarrow 1.$$

Its total length is 6, which is the smallest possible for a route that starts at stall 1, visits every stall, and returns to stall 1.

The stalls are then assigned in the order in which they are first visited:

- Stall 1 is assigned to Team Potato.
- Stall 2 is assigned to Team Tomato.
- Stall 3 is assigned to Team Potato.
- Stall 4 is assigned to Team Tomato.

The other possible team assignment is obtained by visiting stall 4 before stall 3, which swaps their teams. Therefore, the total number of possible team assignments is 2.