

Problem H. Orgrimmar

Input file: **standard input**
Output file: **standard output**
Time limit: 10 seconds
Memory limit: 1024 megabytes

"In my memory, the last time such a tragic farewell to a respected Horde leader was at the top of Thunder Bluff. That day, Mother Earth was crying for him too. "

"This time, it is the Shadow of the Horde who has left us. At this moment, the entire Horde is whispering affectionately for him. "

"Son of Sen'jin, leader of the Darkspear tribe, Warchief of the Horde - Vol'jin."

Born in the cunning and vicious troll race, he spent his life explaining to the world what loyalty and faith are.

A dissociation set of an undirected graph is a set of vertices such that if we keep only the edges between these vertices, each vertex in the set is connected to at most one edge.

The size of a dissociation set is defined by the size of the set of vertices.

The maximal dissociation set of the graph is defined by the dissociation set of the graph with the maximum size.

Sylvanas has a connected undirected graph that has n vertex and $n - 1$ edges, and she wants to find the size of the maximal dissociation set of the graph.

But since she just became the warchief of the Horde, she is too busy to solve the problem.

Please help her to do so.

Input

The input consists of multiple test cases.

The first line contains one integer T ($1 \leq T \leq 10$) denoting the number of test cases.

The following are T test cases.

For each test case, the first line contains one integer n ($n \leq 500000$), which is the number of vertexes.

The following $n - 1$ lines each contains two integers x and y denoting an edge between x and y .

It is guaranteed that the graph is connected.

Output

For each test case, output one line containing one integer indicating the answer.

Notes:

–In this problem, you may need more stack space to pass this problem.

–We suggest you to add the following code into your main function if you use C++.

```
int main() {
    int size(512<<20); // 512M
    __asm__ ( "movq %0, %%rsp\n"::"r"((char*)malloc(size)+size));
    // YOUR CODE
    ...
    exit(0);
}
```

And if you use the code above please DON'T forget to add `exit(0)`; in the end of your main function, otherwise your code may get RE.

Example

standard input	standard output
2	4
5	7
1 2	
2 3	
3 4	
4 5	
10	
1 2	
2 4	
3 2	
5 3	
6 4	
7 5	
8 3	
9 8	
10 7	