Description | Hints | Submissions | Discussions | Notes

Colour Tree

Difficulty	Time Limit	Memory	Score
[illegible]	2 sec	256000KB	100

80/80 XP | 30/30

Description

You have been given a tree with N nodes and N - 1 edges. You want to colour each node, such that no two adjacent nodes (directly connected by an edge) and no two nearly-adjacent nodes (both directly connected to a common node with edges) has the same colour.

Your task is to find the minimum number of colours required to accomplished this.

Input Format

The first line of input contains N. Each of the remaining N-1 lines describes an edge in terms of the two nodes it connects.

Output Format

Print the minimum number of colours require.

Constraints

$1 \leq N \leq 10^5$

Sample Input 1 Copy

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

Sample Output 1 Copy

```
3
```

C++14 | 00:00:00 | 12 px

```cpp
#include<bits/stdc++.h>
using namespace std;
#define endl "\n"
using lli=long long int;

vector<vector<int>>g;
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    int n;
    cin>>n;
    g.resize(n+1);
    for(int i=1;i<=n-1;i++){
        int u,v;
        cin>>u>>v;
        g[u].emplace_back(v);
        g[v].emplace_back(u);
    }
    int temp=0;
    for(auto i:g){
        temp=max(temp,(int)i.size());
    }
    cout<<temp+1;
    return 0;
}
```

Sample Tests | Manual Tests

Test Case 1

Input

```
4
1 2
```

Console | Run on Sample