树的中心

```cpp
#include <bits/stdc++.h>
using namespace std;

const int N = 1e5 + 5, M = 2 * N, INF = 0x3f3f3f3f;
int h[N], tot;
struct edge{
    int to, w, nxt;
}e[M];

void add(int u, int v, int w){
    e[++tot].to = v;
    e[tot].w = w;
    e[tot].nxt = h[u];
    h[u] = tot;
}
int d1[N], d2[N], p[N], up[N];
int dfs_d(int u, int fa){
    d1[u] = d2[u] = -INF;
    for(int i = h[u]; i; i = e[i].nxt){
        int v = e[i].to;
        if(v == fa) continue;
        int d = dfs_d(v, u) + e[i].w;
        if(d >= d1[u]){
            d2[u] = d1[u]; d1[u] = d;
            p[u] = v;
        }
        else if(d > d2[u]){
            d2[u] = d;
        }
    }
    if(d1[u] == -INF) d1[u] = d2[u] = 0;
    return d1[u];
}

void dfs_u(int u, int fa){
    for(int i = h[u]; i; i = e[i].nxt){
        int v = e[i].to;
        if(v == fa) continue;
        if(p[u] == v) up[v] = max(up[u], d2[u]) + e[i].w;
        else up[v] = max(up[u], d1[u]) + e[i].w;
        dfs_u(v, u);
    }
}

int main(){
    int n;
    cin >> n;
    for(int i = 0; i < n - 1; i++){
        int u, v;
        cin >> u >> v;
```

```
        add(u, v, 1);
        add(v, u, 1);
    }
    dfs_d(1, -1);
    dfs_u(1, -1);
    int d = INF;
    for(int i = 1; i <= n; i++){
        d = min(d, max(up[i], d1[i]));
    }
    cout << d << '\n';
    return 0;
}
```