

{% note info %} **摘要** Title: 846. 树的重心 Tag: 树的DFS、树的重心、树 Memory Limit: 64 MB Time Limit: 1000 ms {% endnote %}

Powered by: NEFU AB-IN

[Link](#)

@TOC

846. 树的重心

• 题意

给定一颗树，树中包含 n 个结点（编号 $1 \sim n$ ）和 $n-1$ 条无向边。请你找到树的重心，并输出将重心删除后，剩余各个连通块中点数的最大值。

• 思路

树的重心：**重心是指树中的一个结点，如果将这个点删除后，剩余各个连通块中点数的最大值最小，那么这个节点被称为树的重心。** {% note info %} **唯一性** 树的重心不唯一，但树是唯一的 **树中无环!!!** {% endnote %}

树的DFS遍历

无向图的模板，有向图就不用带fa了

```
def dfs(u, fa):
    for v in g[u]:
        if v == fa:
            continue
        dfs(j) #一般都是先DFS，从底层返回过来结果
    ...
    ...
```

用DFS的好处就是，可以统计出**每个结点的子节点的数量** 思路就是从根节点开始遍历，每次DFS返回以u为根节点的子节点数量

• 代码

新版

```
...
Author: NEFU AB-IN
Date: 2022-03-04 22:26:35
FilePath: \ACM\Acwing\846.1.py
```

```

LastEditTime: 2022-03-04 22:27:42
'''

N = int(1e5 + 10)
g = [[] for _ in range(N)]
ans = N

def dfs(u, fa):
    global ans
    cnt = 1
    res = 0
    for v in g[u]:
        if v == fa:
            continue
        s = dfs(v, u)
        res = max(res, s)
        cnt += s
    res = max(res, n - cnt)
    ans = min(ans, res)
    return cnt

n = int(input())
for i in range(n - 1):
    a, b = map(int, input().split())
    g[a].append(b)
    g[b].append(a)

dfs(1, 0)
print(ans)

```

老版带注释

```

'''
Author: NEFU AB-IN
Date: 2022-03-02 19:43:45
FilePath: \ACM\Acwing\846.py
LastEditTime: 2022-03-02 20:02:42
'''

N = int(1e5 + 10)
st = [0] * N
g = [[] for _ in range(N)]
ans = N

def dfs(u):
    global ans
    st[u] = 1
    cnt = 1 # 以u为根的子节点的个数，一开始只有u一个

```

```
res = 0 # 连通块的点数的最大值
for j in g[u]:
    if st[j] == 0:
        s = dfs(j) #返回的是以u的子节点j为根节点的子节点数
        res = max(res, s)
        cnt += s
res = max(res, n - cnt) #剩余连通块的大小
ans = min(ans, res)
return cnt

n = int(input())
for i in range(n - 1):
    a, b = map(int, input().split())
    g[a].append(b)
    g[b].append(a)

dfs(1)
print(ans)
```