

电话网络

一. 考察内容:

动态规划 树形DP

二. 题目分析:

[题目大意]

给出一棵树，选择一些节点建立通讯站，使得任意一个节点或者该节点相邻的节点都有通讯站，求最小需要建多少通讯站。

[写题思路]

考虑用树形DP实现，设状态 $f[i][0]$ 为子树 i 内节点可以通讯， i 节点不一定能通讯的最小花费， $f[i][1]$ 为子树 i 内节点可以通讯，且 i 节点能通讯的最小花费， $f[i][2]$ 为子树 i 可以通讯，且 i 节点有通讯站的最小花费，

转移： $f[i][0] = \sum f[\text{son}[i]][1]$ ， $f[i][1] = (\sum f[\text{son}[i]][1]) + \min(f[\text{son}[i]][2] - f[\text{son}[i]][1])$ ， $f[i][2] = \sum f[\text{son}[i]][0] + 1$ 。

对树进行遍历并求出每个节点状态即可。

三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATE
/*****
*创建时间: 2018 09 10
*文件类型: 源代码文件
*题目来源: BZOJ
*当前状态: 已通过
*备忘录: 动态规划 树形DP
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
#include <bitset>
// #include <sys/wait.h>
// #include <sys/types.h>
// #include <unistd.h>
using namespace std;
#define MAXN 10001
int n, head[MAXN], f[MAXN][3], _edge, vis[MAXN];
struct EDGE
{
    int a, b, next;
    EDGE(int a=0, int b=0, int next=0):a(a), b(b), next(next){}
}edge[2*MAXN];
/* Variable explain:
f[i][0]: 节点i不设为转接点时子树最小花费 (可以不含i节点)
f[i][1]: 含i节点...
f[i][2]: 节点i设为转接点时子树最小花费
*/
void adde(int a, int b)
{
    edge[++_edge] = EDGE(a, b, head[a]);
    head[a] = _edge;
}
void read()
```

```
{
    int ls1,ls2;
    scanf("%d\n",&n);
    for(int i=1;i<n;++i)scanf("%d%d",&ls1,&ls2),adde(ls1,ls2),adde(ls2,ls1);
    return;
}
void dfs(int a)
{
    vis[a]=1;
    int min_more_cost=1e9;
    for(int i=head[a];i;i=edge[i].next)
    {
        int b=edge[i].b;
        if(vis[b])continue;
        dfs(b);
        min_more_cost=min(min_more_cost,f[b][2]-f[b][1]);
        f[a][0]+=f[b][1];
        f[a][1]+=f[b][1];
        f[a][2]+=f[b][0];
    }
    f[a][1]+=min_more_cost;
    ++f[a][2];
    f[a][1]=min(f[a][1],f[a][2]);
    f[a][0]=min(f[a][0],f[a][1]);
}
int main()
{
    // freopen(".in","r",stdin);
    // freopen(".out","w",stdout);
    read();
    dfs(1);
    printf("%d\n",f[1][1]);
    return 0;
}
```

[<题目跳转>](#) [<查看代码>](#)