

Computer

一. 考察内容:

树形DP 动态规划

二. 题目分析:

[题目大意]

求一棵树上每个节点能到达的最远节点之间的距离。

[写题思路]

首先，先找到这棵树一条直径上的两个端点，不难发现，任意一个点可以到达的最远点都是这两个点中的一个，所以，分别以这两个点为根，求出每个点距离根的深度，最后求一个最大值即可。

三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATE
/*****
*创建时间: 2018 08 11
*文件类型: 源代码文件
*题目来源: HDU
*当前状态: 已通过
*备忘录: 树形DP
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
using namespace std;
#define MAXN 100001
int n, head[MAXN], _edge, max1, max2, maxv, d[2][MAXN];
struct EDGE
{
    int a, b, v, next;
    EDGE(int a=0, int b=0, int v=0, int next=0):a(a), b(b), v(v), next(next){}
}edge[2*MAXN];
/* Variable explain:

*/
void adde(int a, int b, int v)
{
    edge[++_edge]=EDGE(a, b, v, head[a]);
    head[a]=_edge;
}
void read()
{
    int ls1, ls2;
    if(scanf("%d", &n)==-1)exit(0);
    for(int i=1; i<=n; ++i) head[i]=d[0][i]=d[1][i]=0;
    max1=max2=maxv=0;
    _edge=1;
    for(int i=2; i<=n; ++i) scanf("%d%d", &ls1, &ls2), adde(ls1, i, ls2), adde(i, ls1, ls2);
    return;
}
void dfs1(int a, int l, int v)
```

```
{
    if(v>maxv)maxv=v,max1=a;
    for(int i=head[a];i;i=edge[i].next)
    {
        if((i^1)==l)continue;
        int b=edge[i].b;
        dfs1(b,i,v+edge[i].v);
    }
}
void dfs2(int a,int l,int d[])
{
    d[a]=d[edge[l].a]+edge[l].v;
    for(int i=head[a];i;i=edge[i].next)
    {
        if((i^1)==l)continue;
        int b=edge[i].b;
        dfs2(b,i,d);
    }
}
int main()
{
    while(1)
    {
        read();
        dfs1(1,0,0);
        max2=max1;
        max1=maxv=0;
        dfs1(max2,0,0);
        dfs2(max1,0,d[0]);
        dfs2(max2,0,d[1]);
        int ls1=0,ls2=0;
        for(int i=1;i<=n;++i)
        {
            printf("%d\n",max(d[0][i],d[1][i]));
        }
    }
    return 0;
}
```

[<题目跳转>](#) [<查看代码>](#)