**[树形DP](http://www.cppblog.com/notonlysuccess/archive/2009/05/11/82614.html)**

<http://acm.hdu.edu.cn/showproblem.php?pid=2196>  
向下搜一遍，向上搜一遍  
<http://acm.hdu.edu.cn/showproblem.php?pid=1561>  
对每一个节点进行一次背包，好题啊，两个DP树形和背包结合的  
<http://acm.hdu.edu.cn/showproblem.php?pid=1011>  
这道是当年省赛的压轴题，但是感觉和上一道差不多，一样的难度，唯一不同的就是这个是无向图(我由于思维惯性拿来当单向图作，纠结了好久。。。)   
树形+背包+临街表  
  
下边是从天涯空间里找出来的练习  
<http://acm.pku.edu.cn/JudgeOnline/problem?id=3345>  
<http://acm.zju.edu.cn/onlinejudge/showProblem.do?problemCode=3201>  
<http://acm.pku.edu.cn/JudgeOnline/problem?id=3107>  
<http://acm.pku.edu.cn/JudgeOnline/problem?id=1655>  
<http://acm.pku.edu.cn/JudgeOnline/problem?id=2378>  
<http://acm.pku.edu.cn/JudgeOnline/problem?id=3140>  
<http://acm.hdu.edu.cn/showproblem.php?pid=2242>  
<http://acm.timus.ru/problem.aspx?space=1&num=1018>  
<http://acm.pku.edu.cn/JudgeOnline/problem?id=1947>  
<http://acm.pku.edu.cn/JudgeOnline/problem?id=2057>  
<http://acm.pku.edu.cn/JudgeOnline/problem?id=2486>  
<http://acm.pku.edu.cn/JudgeOnline/problem?id=1848>  
<http://acm.pku.edu.cn/JudgeOnline/problem?id=2152>  
  
  
  
<http://acm.hdu.edu.cn/showproblem.php?pid=1520>  
(第一个树形DP，附代码)  
最最简单的树形DP  
还学习了父子兄弟结构，爽

#include "stdio.h"  
  
struct Tree{  
    int father;  
    int child;  
    int brother;  
    int TakeParty;  
    int Not;  
    int Max() {  
        return TakeParty > Not ? TakeParty : Not;  
    }  
    void init() {  
        father = child = brother = Not = 0;  
    }  
}tree[6001];  
  
void dfs(int idx ) {  
    int child;  
    child = tree[idx].child;  
    while(child) {  
        dfs(child);  
        tree[idx].TakeParty += tree[child].Not;  
        tree[idx].Not += tree[child].Max();  
        child = tree[child].brother;  
    }  
}  
  
int main() {  
    int n,i,a,b;  
    while(scanf("%d",&n) == 1) {  
        for(i =1 ; i <= n ; i ++) {  
            scanf("%d",&tree[i].TakeParty);  
            tree[i].init();  
        }  
        while(scanf("%d%d",&a,&b),a+b) {  
            tree[a].father = b;  
            tree[a].brother = tree[b].child;  
            tree[b].child = a;  
        }  
        for(i = 1 ; i <= n ; i ++) {  
            if(!tree[i].father) {  
                dfs(i);  
                printf("%d\n",tree[i].Max());  
                break;  
            }  
        }  
    }  
    return 0;  
}