

# 没有兄弟的舞会

## 一. 考察内容:

动态规划 树形DP

## 二. 题目分析:

[题目大意]

给出一颗树，树上节点有点权，求出一个点集，使得在最多只有一对兄弟的情况下权值和最大。

[写题思路]

简单的树形DP，设 $f[i][0]$ 为 $i$ 的子树可以获得的最大点权，不能有兄弟，且不包括点 $i$ ， $f[i][1]$ 为 $i$ 的子树可以获得的最大点权，其中可以有一对兄弟，不包括点 $i$ 。

转移时 $f[i][0] = \sum f[j][0]$ ，其中 $j$ 为 $i$ 的子节点， $f[i][1]$ 可以选择该点的子节点作为这个有兄弟的点，也可以找该点的子节点中某点作为有兄弟的点，求出最大值和次大值、最小值和次小值维护一下即可。

## 三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATE
/*****
*创建时间: 2018 09 26
*文件类型: 源代码文件
*题目来源: HDU
*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 100001
int n, fa[MAXN], son[MAXN], bro[MAXN], v[MAXN], Max[MAXN][2], Min[MAXN][2];
/* Variable explain:

*/
void addt(int a, int b)
{
    bro[b] = son[a];
    son[a] = b;
    fa[b] = a;
}
void read()
{
    int ls1;
    scanf("%d", &n);
    for(int i=1; i<=n; ++i) fa[i] = bro[i] = son[i] = Max[i][0] = Max[i][1] = Min[i][0] = Min[i][1] = 0;
    for(int i=2; i<=n; ++i) scanf("%d", &ls1), addt(ls1, i);
    for(int i=1; i<=n; ++i) scanf("%d", &v[i]);
    return;
}
void dfs(int a)
```

```
{
    int max1=0,max2=0,min1=1e9,min2=1e9;
    for(int i=son[a];i;i=bro[i])
    {
        dfs(i);
        Max[a][0]+=Max[i][0];
        Min[a][0]+=Min[i][0];
        Max[a][1]=max(Max[a][1],Max[i][1]-Max[i][0]);
        Min[a][1]=min(Min[a][1],Min[i][1]-Min[i][0]);
        if(max2<v[i])max2=v[i];
        if(max1<max2)swap(max1,max2);
        if(min2>v[i])min2=v[i];
        if(min1>min2)swap(min1,min2);
    }
    Max[a][0]+=max(0,max1);
    Min[a][0]+=min(0,min1);
    Max[a][1]=max(Max[a][1],max2)+Max[a][0];
    Min[a][1]=min(Min[a][1],min2)+Min[a][0];
}
int main()
{
    // freopen(".out","r",stdin);
    // freopen(".out","w",stdout);
    int T;
    scanf("%d",&T);
    while(T--)
    {
        read();
        dfs(1);
        printf("%d %d\n",Max[1][1]+max(0,v[1]),Min[1][1]+min(0,v[1]));
    }
    return 0;
}
```

[<题目跳转>](#) [<查看代码>](#)