

## H题题解报告：

题意：从某节点可拓展到其他节点上并获得其点值，每个节点点值为1或-1，求从任意一点能扩展到的最大点值。（所有点构成一颗生成树）

虽然题意是一颗无根生成树，但是我们可以假定某个点为根，比如节点1。

考虑下根能拓展到的最大点值是多少？

或者说根能拓展到哪些区域？

我们不能根据根的叶子节点直接判断是否能够拓展，因为我们不知道通过负的叶子节点的收益是否大于不扩展这个点的损失。

既然自上向下的方式不行，那我们就自下向上。

第一遍dfs，找到所有叶子节点，维护一个数组mmax\_val[]用来表示通过这个叶子节点的收益。回溯的时候根据返回值，大于0就更新，否则就不更新。

```
int dfs1(int fa,int u){
    int tot=0,tmp=0;
    while(tot<node[u].size()){
        if(node[u][tot]==fa){//不能访问父节点
            ++tot;
            continue;
        }
        int ttmp=dfs1(u,node[u][tot]);
        if(ttmp>0) tmp+=ttmp;//tmp用来存该节点所有子节点大于0的回溯值
        ++tot;
    }
    mmax_val[u]=tmp+val[u];//val[]存的是节点初始大小
    return tmp+val[u];//返回的时候当然要带上节点自己的值
}
```

经过一遍dfs，我们就可以得到根节点可以拓展到的最大点值了。

但是这个dfs，一次只能找到一个点可以拓展到的最大点值，总不能对所有点都跑一遍dfs。

我们令dp[i]表示第i个节点可以拓展到的最大点值，

由第一遍dfs我们得到了dp[i],

根据转移方程

```
dp[i] = max (mmax_val[i], dp[i] + mmax_val[i]) //mmax_val[i] < 0
dp[i] = max (mmax_val[i], dp[i])//mmax_val[i] >= 0
```

我们就可以自上而下地更新dp[i]了。

如何理解这两个转移方程？

假如mmax\_val[i] >= 0，那么dp[i]一定经过这个子节点（因为dp[i]存的是最大拓展点值，子节点总效益大于0的话肯定会被拓展掉的）

而同一个连通块里的可拓展到的最大点值其实都是一样的。

$max\_val[i] < 0$ ，那么他的父节点一定没有拓展他及他的子节点们，但是i点可以反过来踏入他父节点来获取最大点值。

理解这个之后，第二个dfs也就很好写了。

（也就一个自下而上，一个自上而下）

（那个转移方程貌似不是很简洁，可以稍微改改）

**时间复杂度:  $O(m)$**

每个边只要走两边就可以了