

树上启发式合并

```

#include <cstdio>
#include <cstring>
#define maxn 100010

int n,m,col[maxn];
struct edge{int y,next;};
edge e[maxn*2];
int first[maxn];
void buildroad(int x,int y)
{
    static int len=0;
    e[++len]=(edge){y,first[x]};
    first[x]=len;
}
int size[maxn],mson[maxn];
void dfs1(int x,int fa)//求重儿子
{
    size[x]=1;
    for(int i=first[x];i;i=e[i].next)
    {
        int y=e[i].y;
        if(y==fa)continue;
        dfs1(y,x);
        if(size[y]>size[mson[x]])mson[x]=y;
        size[x]+=size[y];
    }
}
int tong[maxn],ans[maxn],now_ans=0;
void go(int x,int fa,int type)
{
    tong[col[x]]+=type;
    if(type==1&&tong[col[x]]==1)now_ans++;
    if(type==-1&&tong[col[x]]==0)now_ans--;
    for(int i=first[x];i;i=e[i].next)
        if(e[i].y!=fa)go(e[i].y,x,type);
}
void dfs2(int x,int fa,bool del)
//求解，del表示求完x的子树的答案后需不需要清空x的子树的信息
{
    for(int i=first[x];i;i=e[i].next)//先统计轻儿子的答案
        if(e[i].y!=fa&&e[i].y!=mson[x])dfs2(e[i].y,x,true);
    if(mson[x]!=0)dfs2(mson[x],x,false);//最后统计重儿子的答案

    tong[col[x]]++;if(tong[col[x]]==1)now_ans++;//统计自己以及轻子树的信息
    for(int i=first[x];i;i=e[i].next)
        if(e[i].y!=fa&&e[i].y!=mson[x])go(e[i].y,x,1);
    ans[x]=now_ans;//得到自己的答案

    if(del)go(x,fa,-1);//假如要删掉自己的信息，就暴力地删掉
}

int main()
{
    scanf("%d",&n);
    for(int i=1,x,y;i<n;i++)
        scanf("%d %d",&x,&y),buildroad(x,y),buildroad(y,x);

```

```
for(int i=1;i<=n;i++)
scanf("%d",&col[i]);
dfs1(1,0);
dfs2(1,0,false);
scanf("%d",&m);
for(int i=1,x;i<=m;i++)
scanf("%d",&x),printf("%d\n",ans[x]);
}
```