题解

建10棵线段树，表示10种颜色。

Set操作只需把表示原来颜色的树上的这个点删去，再在表示新颜色的树上加上就行了。

Change操作，当某个区间的左 /右儿子被修改范围覆盖时，交换这两种颜色的左 /右儿子。

Ask就简单了。

代码

#include <cstdio>

#include <iostream>

#include <cstring>

#include <algorithm>

using namespace std;

const int Q=50005;

int ls[Q<<4],rs[Q<<4],cnt[Q<<4],n,tot=10;

int dep[Q],id[Q],be[Q],f[Q],bs[Q],si[Q],out[Q];

int last[Q],nn[Q<<1],e[Q<<1],inc=0,c[Q],v[Q];

void add(int x,int y)

{

e[++inc]=y;

nn[inc]=last[x];

last[x]=inc;

}

inline int R()

{

char o=getchar();

int t=0;bool f=false;

while(o>'9'||o<'0')

{

if(o=='-')f=true;

o=getchar();

}

while(o<='9'&&o>='0')

{

t=10\*t+int(o-'0');

o=getchar();

}

return f?-t:t;

}

void fbc(int x)

{

bs[x]=0,si[x]=1;

int y,t,now=0;

for(t=last[x];t;t=nn[t])

{

y=e[t];

if(y==f[x])continue;

dep[y]=dep[x]+1,f[y]=x;

fbc(y);

si[x]+=si[y];

if(now<si[y])

now=si[y],bs[x]=y;

}

}

void cbc(int x,int now)

{

id[x]=++inc,be[x]=now;

if(bs[x]!=0)cbc(bs[x],now);

int t,y;

for(t=last[x];t;t=nn[t])

{

y=e[t];

if(y==f[x]||y==bs[x])continue;

cbc(y,y);

}

out[x]=inc;

}

void xiu(int now,int l,int r,int x,int v)

{

if(l==r){

cnt[now]=v;

return;

}

int mid=(l+r)>>1;

if(x<=mid){

if(!ls[now])ls[now]=++tot;

xiu(ls[now],l,mid,x,v);

}

else{

if(!rs[now])rs[now]=++tot;

xiu(rs[now],mid+1,r,x,v);

}

cnt[now]=cnt[ls[now]]+cnt[rs[now]];

}

int get(int now,int l,int r,int x,int y)

{

if(x<=l&&y>=r)return cnt[now];

int mid=(l+r)>>1,temp=0;

if(x<=mid&&ls[now])temp+=get(ls[now],l,mid,x,y);

if(y>mid&&rs[now])temp+=get(rs[now],mid+1,r,x,y);

return temp;

}

void swip(int &nowl,int &nowr,int l,int r,int x,int y)

{

if(x<=l&&y>=r){

swap(nowl,nowr);

return;

}

int mid=(l+r)>>1;

if(x<=mid){

if(!ls[nowl])ls[nowl]=++tot;

if(!ls[nowr])ls[nowr]=++tot;

swip(ls[nowl],ls[nowr],l,mid,x,y);

}

if(y>mid){

if(!rs[nowl])rs[nowl]=++tot;

if(!rs[nowr])rs[nowr]=++tot;

swip(rs[nowl],rs[nowr],mid+1,r,x,y);

}

cnt[nowl]=cnt[ls[nowl]]+cnt[rs[nowl]];

cnt[nowr]=cnt[ls[nowr]]+cnt[rs[nowr]];

}

int main()

{

char o;

int i,x,y;

n=R();

for(i=1;i<=n;i++)

c[i]=R()+1;

for(i=1;i<=n;i++)

v[i]=R();

for(i=1;i<n;i++)

{

x=R()+1,y=R()+1;

add(x,y),add(y,x);

}

inc=0;

dep[1]=1,f[1]=0;

fbc(1);

cbc(1,1);

for(i=1;i<=n;i++)

xiu(c[i],1,n,id[i],v[i]);

for(i=1;i<=10;i++)

c[i]=i;

for(int m=R();m;--m)

while(true)

{

o=getchar();

if(o=='A')

{

int temp=0;

x=R()+1,y=R()+1,i=R()+1;

while(be[x]!=be[y])

{

if(dep[be[x]]<dep[be[y]])swap(x,y);

temp+=get(c[i],1,n,id[be[x]],id[x]);

x=f[be[x]];

}

if(dep[x]>dep[y])swap(x,y);

temp+=get(c[i],1,n,id[x],id[y]);

printf("%d\n",temp);

break;

}

if(o=='C')

{

i=R()+1,x=R()+1,y=R()+1;

swip(c[x],c[y],1,n,id[i],out[i]);

break;

}

if(o=='S')

{

x=R()+1,i=R()+1,y=R();

for(int t=1;t<=10;t++)

xiu(c[t],1,n,id[x],0);

xiu(c[i],1,n,id[x],y);

break;

}

}

return 0;

}