

****

**ACM/ICPC**

沈中海

计算机学院

Splay:

#include <cstdio>

#include <cstring>

#include <cstdlib>

#include <iostream>

#define N 6000000

#define INF 999999999

using namespace std;

long long root, n, m, flag2[N], list[N], fa[N], l[N], r[N], size[N], f[N][3], g[N], flag[N], sum[N], a[N];

char s[20];

long long read(){

long long p=0, q=1;

char ch=getchar();

while (ch<'0' || ch>'9'){

if (ch=='-') q=-1;

ch=getchar();

}

while (ch>='0' && ch<='9') p=p\*10+ch-'0', ch=getchar();

return p\*q;

}

void update(long long t){

if (l[t]) fa[l[t]]=t;

if (r[t]) fa[r[t]]=t;

sum[t]=sum[l[t]]+sum[r[t]]+a[t];

size[t]=size[l[t]]+size[r[t]]+1;

f[t][0]=max(f[l[t]][0],sum[l[t]]+f[r[t]][0]+a[t]);

f[t][1]=max(f[r[t]][1],sum[r[t]]+f[l[t]][1]+a[t]);

f[t][2]=f[l[t]][1]+f[r[t]][0]+a[t];

f[t][2]=max(f[t][2],max(f[l[t]][2],f[r[t]][2]));

}

void pushdown(long long t){

if (flag[t]){

if (l[t]) flag[l[t]]^=1;

if (r[t]) flag[r[t]]^=1;

swap(l[t],r[t]);

swap(f[l[t]][0],f[l[t]][1]);

swap(f[r[t]][0],f[r[t]][1]);

flag[t]=0;

}

if (flag2[t]){

if (l[t]){

sum[l[t]]=g[t]\*size[l[t]];

if (g[t]>0)

f[l[t]][0]=f[l[t]][1]=f[l[t]][2]=sum[l[t]];

else

f[l[t]][0]=f[l[t]][1]=0, f[l[t]][2]=g[t];

g[l[t]]=a[l[t]]=g[t];

flag2[l[t]]=1;

}

if (r[t]){

sum[r[t]]=g[t]\*size[r[t]];

if (g[t]>0)

f[r[t]][0]=f[r[t]][1]=f[r[t]][2]=sum[r[t]];

else

f[r[t]][0]=f[r[t]][1]=0, f[r[t]][2]=g[t];

g[r[t]]=a[r[t]]=g[t];

flag2[r[t]]=1;

}

flag2[t]=g[t]=0;

}

}

long long build(long long le, long long ri){

if (le>ri) return 0;

long long mid=le+ri>>1;

l[mid]=build(le,mid-1);

r[mid]=build(mid+1,ri);

update(mid);

return mid;

}

void insert(long long &t, long long k, long long p){

if (!t){

if (!size[t=p]){

size[t]=1;

f[t][0]=f[t][1]=a[t]>0?a[t]:0;

f[t][2]=sum[t]=a[t];

}

return;

}

pushdown(t);

if (size[l[t]]+1<=k) insert(r[t],k-size[l[t]]-1,p);

else insert(l[t],k,p);

update(t);

}

void zig(long long t){

long long f1=fa[t], f2=fa[f1];

if (f2)

if (l[f2]==f1) l[f2]=t;else r[f2]=t;

fa[t]=f2;

l[f1]=r[t];

r[t]=f1;

update(f1);

update(t);

}

void zag(long long t){

long long f1=fa[t], f2=fa[f1];

if (f2)

if (l[f2]==f1) l[f2]=t;else r[f2]=t;

fa[t]=f2;

r[f1]=l[t];

l[t]=f1;

update(f1);

update(t);

}

void splay(long long t){

long long ri=1;

list[1]=t;

for (long long i=1;fa[list[i]];i++) list[++ri]=fa[list[i]];

for (long long i=ri;i;i--) pushdown(list[i]);

long long f1=fa[t], f2=fa[f1];

while (f2){

if (l[f2]==f1)

if (l[f1]==t) zig(f1), zig(t);

else zag(t), zig(t);

else

if (r[f1]==t) zag(f1), zag(t);

else zig(t), zag(t);

f1=fa[t];f2=fa[f1];

}

if (f1)

if (l[f1]==t) zig(t);else zag(t);

root=t;

}

long long find(long long t, long long k){

pushdown(t);

while (size[l[t]]+1!=k){

if (size[l[t]]+1<k)

k-=size[l[t]]+1, t=r[t];

else

t=l[t];

pushdown(t);

}

return t;

}

void del(long long x, long long y){

splay(x);

fa[r[x]]=0;

splay(y);

l[r[root=x]=y]=0;

update(y);

update(x);

}

void modify(long long x, long long y, long long z){

splay(x);

fa[r[x]]=0;

splay(y);

r[root=x]=y;

flag2[l[y]]=1;

g[l[y]]=a[l[y]]=z;

sum[l[y]]=size[l[y]]\*z;

if (z>0)

f[l[y]][0]=f[l[y]][1]=f[l[y]][2]=sum[l[y]];

else

f[l[y]][0]=f[l[y]][1]=0, f[l[y]][2]=z;

update(y);

update(x);

}

void reverse(long long x, long long y){

splay(x);

fa[r[x]]=0;

splay(y);

fa[r[root=x]=y]=x;

flag[l[y]]^=1;

swap(f[l[y]][0],f[l[y]][1]);

update(y);

update(x);

}

void calc(long long x, long long y){

splay(x);

fa[r[x]]=0;

splay(y);

fa[r[root=x]=y]=x;

printf("%d\n", sum[l[y]]);

update(y);

update(x);

}

void print(long long t){

if (!t) return;

pushdown(t);

print(l[t]);

printf("%d ", a[t]);

print(r[t]);

}

int main(){

freopen("sequence4.in","r",stdin);

freopen("1.ans","w",stdout);

n=read();m=read();

for (long long i=1;i<=n;i++) a[i]=read();

f[0][2]=-INF;

root=build(1,n);

a[N-3]=a[N-2]=-INF;

insert(root,0,N-3);

insert(root,n+1,N-2);

for (long long i=1;i<=m;i++){

scanf("%s", s);

if (s[0]=='I'){

long long pos=read(), tot=read(), n2=n+tot, root2;

if (!tot) continue;

for (long long i=n+1;i<=n2;i++) a[i]=read();

root2=build(n+1,n2);

insert(root,pos+1,root2);

splay(root2);

n=n2;

}

if (s[0]=='D'){

long long x=read(), y=read()+x+1;

if (x+1==y) continue;

x=find(root,x);

y=find(root,y);

del(x,y);

}

if (s[2]=='K'){

long long x=read(), y=read()+x+1, z=read();

if (x+1==y) continue;

x=find(root,x);

y=find(root,y);

modify(x,y,z);

}

if (s[0]=='R'){

long long x=read(), y=read()+x+1;

if (x+1==y) continue;

x=find(root,x);

y=find(root,y);

reverse(x,y);

}

if (s[0]=='G'){

long long x=read(), y=read()+x+1;

x=find(root,x);

y=find(root,y);

calc(x,y);

}

if (s[2]=='X'){

printf("%d\n", f[root][2]);

}

}

return 0;

}

LCT：

#include <cstdio>

#include <cstring>

#include <cstdlib>

#include <iostream>

#define mo 51061

#define N 200000

typedef unsigned int ll;

using namespace std;

int n, q, size[N], l[N], r[N], fa[N], rev[N], list[N];

ll sum[N], f[N], at[N], mt[N];

int read(){

int p=0;

char ch=getchar();

while (ch<'0' || ch>'9') ch=getchar();

while (ch>='0' && ch<='9') p=p\*10+ch-'0', ch=getchar();

return p;

}

bool isroot(int t){

return (l[fa[t]]!=t) && (r[fa[t]]!=t);

}

void calc(int u, int m, int a){

if (!u) return;

f[u]=(f[u]\*m+a)%mo;

sum[u]=(sum[u]\*m+a\*size[u])%mo;

at[u]=(at[u]\*m+a)%mo;

mt[u]=(mt[u]\*m)%mo;

}

void update(int t){

if (l[t]) fa[l[t]]=t;

if (r[t]) fa[r[t]]=t;

sum[t]=(f[t]+sum[l[t]]+sum[r[t]])%mo;

size[t]=1+size[l[t]]+size[r[t]];

}

void pushdown(int t){

if (rev[t]){

swap(l[t],r[t]);

if (l[t]) rev[l[t]]^=1;

if (r[t]) rev[r[t]]^=1;

rev[t]=0;

}

int ta=at[t], tm=mt[t];

if (ta || tm!=1){

calc(l[t],tm,ta);

calc(r[t],tm,ta);

}

at[t]=0;mt[t]=1;

}

void zig(int t){

int f1=fa[t], f2=fa[f1];

if (!isroot(f1))

if (l[f2]==f1) l[f2]=t;else r[f2]=t;

fa[t]=f2;

l[f1]=r[t];

r[t]=f1;

update(f1);

update(t);

}

void zag(int t){

int f1=fa[t], f2=fa[f1];

if (!isroot(f1))

if (l[f2]==f1) l[f2]=t;else r[f2]=t;

fa[t]=f2;

r[f1]=l[t];

l[t]=f1;

update(f1);

update(t);

}

void splay(int t){

int ri=1;

list[1]=t;

for (int i=1;!isroot(list[i]);i++) list[++ri]=fa[list[i]];

for (int i=ri;i;i--){

pushdown(list[i]);

}

int f1=fa[t], f2=fa[f1];

while (!isroot(t) && !isroot(f1)){

if (l[f2]==f1)

if (l[f1]==t) zig(f1), zig(t);

else zag(t), zig(t);

else

if (r[f1]==t) zag(f1), zag(t);

else zig(t), zag(t);

f1=fa[t];f2=fa[f1];

}

if (!isroot(t))

if (l[f1]==t) zig(t);else zag(t);

}

void access(int u){

for (int v=0;u;v=u,u=fa[u]){

splay(u);

r[u]=v;

update(u);

}

}

void makeroot(int u){

access(u);

splay(u);

rev[u]^=1;

}

void split(int u, int v){

makeroot(u);

access(v);

splay(v);

}

void link(int u, int v){

makeroot(u);

fa[u]=v;

}

void cut(int u, int v){

split(u,v);

fa[u]=l[v]=0;

update(v);

}

void modify(int u, int v, int m, int a){

split(u,v);

calc(v,m,a);

}

int main(){

n=read();q=read();

for (int i=1;i<=n;i++) size[i]=f[i]=sum[i]=mt[i]=1;

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

int u=read(), v=read();

link(u,v);

}

for (int i=1;i<=q;i++){

char s[2];

scanf("%s", s);

int u=read(), v=read();

if (s[0]=='+'){

int c=read();

modify(u,v,1,c);

}

if (s[0]=='-'){

cut(u,v);

u=read();v=read();

link(u,v);

}

if (s[0]=='\*'){

int c=read();

modify(u,v,c,0);

}

if (s[0]=='/'){

split(u,v);

printf("%d\n", sum[v]);

}

}

return 0;

}

Suffix array:

#include <cstdio>

#include <cstring>

#include <cstdlib>

#include <iostream>

using namespace std;

int ls, a[3000], wv[3000], sa[3000], rk[3000], y[3000], r[3000], h[3000];

char s[3000];

int main(){

while (scanf("%s", s)){

ls=strlen(s);

int m=max(ls,26);

for (int i=0;i<2\*ls;i++) rk[i]=-1;

for (int i=0;i<m;i++) wv[i]=0;

for (int i=0;i<ls;i++) a[i]=s[i]-'a';

for (int i=0;i<ls;i++) wv[a[i]]++;

for (int i=1;i<m;i++) wv[i]+=wv[i-1];

for (int i=0;i<ls;i++) sa[--wv[a[i]]]=i;

rk[sa[0]]=0;

for (int i=1;i<ls;i++) rk[sa[i]]=rk[sa[i-1]]+(a[sa[i]]!=a[sa[i-1]]);

for (int j=1;j<ls;j\*=2){

int p=0;

for (int i=ls-j;i<ls;i++) y[++p]=i;

for (int i=0;i<ls;i++)

if (sa[i]>=j) y[++p]=sa[i]-j;

for (int i=0;i<m;i++) wv[i]=0;

for (int i=0;i<ls;i++) wv[rk[i]]++;

for (int i=1;i<m;i++) wv[i]+=wv[i-1];

for (int i=ls;i;i--) sa[--wv[rk[y[i]]]]=y[i];

r[sa[0]]=0;

for (int i=1;i<ls;i++)

r[sa[i]]=r[sa[i-1]]+(rk[sa[i-1]]!=rk[sa[i]] || rk[j+sa[i-1]]!=rk[j+sa[i]]);

for (int i=0;i<ls;i++)

rk[i]=r[i];

}

int j=0;

for (int i=0;i<ls;i++)

if (rk[i]<ls-1){

for (;j+sa[rk[i]+1]<ls && j+i<ls && a[j+sa[rk[i]+1]]==a[i+j];++j);

h[rk[i]]=j?j--:0;

}

for (int i=0;i<ls-1;i++) cout<<h[i]<<endl;

}

return 0;

}

Miller-Rabin:

#include <iostream>

#include <cstdio>

#include <algorithm>

#include <cmath>

#include <cstring>

#include <map>

using namespace std;

const int times = 20;

int number = 0;

map<long long, int>m;

long long Random( long long n ) //生成[ 0 , n ]的随机数

{

return ((double)rand( ) / RAND\_MAX\*n + 0.5);

}

long long q\_mul( long long a, long long b, long long mod ) //快速计算 (a\*b) % mod

{

long long ans = 0;

while(b)

{

if(b & 1)

{

b--;

ans =(ans+ a)%mod;

}

b /= 2;

a = (a + a) % mod;

}

return ans;

}

long long q\_pow( long long a, long long b, long long mod ) //快速计算 (a^b) % mod

{

long long ans = 1;

while(b)

{

if(b & 1)

{

ans = q\_mul( ans, a, mod );

}

b /= 2;

a = q\_mul( a, a, mod );

}

return ans;

}

bool witness( long long a, long long n )//miller\_rabin算法的精华

{//用检验算子a来检验n是不是素数

long long tem = n - 1;

int j = 0;

while(tem % 2 == 0)

{

tem /= 2;

j++;

}

//将n-1拆分为a^r \* s

long long x = q\_pow( a, tem, n ); //得到a^r mod n

if(x == 1 || x == n - 1) return true; //余数为1则为素数

while(j--) //否则试验条件2看是否有满足的 j

{

x = q\_mul( x, x, n );

if(x == n - 1) return true;

}

return false;

}

bool miller\_rabin( long long n ) //检验n是否是素数

{

if(n == 2)

return true;

if(n < 2 || n % 2 == 0)

return false; //如果是2则是素数，如果<2或者是>2的偶数则不是素数

for(int i = 1; i <= times; i++) //做times次随机检验

{

long long a = Random( n - 2 ) + 1; //得到随机检验算子 a

if(!witness( a, n )) //用a检验n是否是素数

return false;

}

return true;

}

int main( )

{

long long tar;

cout<<rand()<<endl;

cout<<RAND\_MAX<<endl;

cout<<Random( 100 - 2 )<<endl;

cout<<Random( 100 - 2 )<<endl;

while(cin >> tar)

{

if(miller\_rabin( tar )) //检验tar是不是素数

cout << "Yes, Prime!" << endl;

else

cout << "No, not prime.." << endl;

}

return 0;

}

树链剖分:

#include <cstdio>

#include <cstring>

#include <cstdlib>

#include <iostream>

#include <algorithm>

#define N 31000

#define M 100000

#define INF 999999

typedef long long ll;

using namespace std;

int n, cnt, son[N], sum[N\*4], dep[N], fa[N], f[N\*4], nex[M], nu[M], dfn[N], pre[N], top[N];

char s[10];

int read(){

int p=0, q=1;

char ch=getchar();

while (ch<'0' || ch>'9'){

if (ch=='-') q=-1;

ch=getchar();

}

while (ch>='0' && ch<='9') p=p\*10+ch-'0', ch=getchar();

return p\*q;

}

void add(int u, int v){

nex[++cnt]=nex[u];nex[u]=cnt;nu[cnt]=v;

}

void dfs1(int u, int father){

son[u]=1;

int p=0;

for (int j=nex[u];j;j=nex[j]){

int v=nu[j];

if (v==father) continue;

fa[v]=u;

dep[v]=dep[u]+1;

dfs1(v,u);

son[u]+=son[v];

if (son[v]>son[p]) p=v;

}

pre[u]=p;

}

void dfs2(int u, int father){

if (!u) return;

if (pre[father]==u) top[u]=top[father];else top[u]=u;

dfn[u]=++cnt;

dfs2(pre[u],u);

for (int j=nex[u];j;j=nex[j]){

int v=nu[j];

if (v==father || v==pre[u]) continue;

dfs2(v,u);

}

}

void update(int t, int l, int r, int x, int y){

if (l==r){

f[t]=sum[t]=y;

return;

}

int mid=l+r>>1;

if (x<=mid) update(t<<1,l,mid,x,y);else update((t<<1)+1,mid+1,r,x,y);

sum[t]=sum[t<<1]+sum[(t<<1)+1];

f[t]=max(f[t<<1],f[(t<<1)+1]);

}

int get\_max(int t, int l, int r, int le, int ri){

if (le<=l && r<=ri) return f[t];

int mid=l+r>>1, p=-INF;

if (le<=mid) p=max(p,get\_max(t<<1,l,mid,le,ri));

if (ri>mid) p=max(p,get\_max((t<<1)+1,mid+1,r,le,ri));

return p;

}

void query\_max(int u, int v){

int f1=top[u], f2=top[v], ans=-INF;

while (f1!=f2)

if (dep[f1]<dep[f2])

ans=max(ans,get\_max(1,1,n,dfn[f2],dfn[v])),

v=fa[f2],

f2=top[v];

else

ans=max(ans,get\_max(1,1,n,dfn[f1],dfn[u])),

u=fa[f1],

f1=top[u];

ans=max(ans,get\_max(1,1,n,min(dfn[u],dfn[v]),max(dfn[u],dfn[v])));

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

}

int get\_sum(int t, int l, int r, int le ,int ri){

if (le<=l && r<=ri) return sum[t];

int mid=l+r>>1, p=0;

if (le<=mid) p+=get\_sum(t<<1,l,mid,le,ri);

if (ri>mid) p+=get\_sum((t<<1)+1,mid+1,r,le,ri);

return p;

}

void query\_sum(int u, int v){

int f1=top[u], f2=top[v], ans=0;

while (f1!=f2)

if (dep[f1]<dep[f2])

ans+=get\_sum(1,1,n,dfn[f2],dfn[v]),

v=fa[f2],

f2=top[v];

else

ans+=get\_sum(1,1,n,dfn[f1],dfn[u]),

u=fa[f1],

f1=top[u];

ans+=get\_sum(1,1,n,min(dfn[u],dfn[v]),max(dfn[u],dfn[v]));

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

}

int main(){

cnt=n=read();

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

int u=read(), v=read();

add(u,v);

add(v,u);

}

dfs1(1,0);

dfs2(1,cnt=0);

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

update(1,1,n,dfn[i],read());

for (int q=read();q;q--){

scanf("%s", s);

int u=read(), v=read();

if (s[0]=='C') update(1,1,n,dfn[u],v);

if (s[1]=='M') query\_max(u,v);

if (s[1]=='S') query\_sum(u,v);

}

return 0;

}

Dinic:

#include <iostream>

#include <cstdlib>

#include <cstring>

#include <cstdio>

#include <algorithm>

#define N 10000

#define M 200000

#define INF 1e9

using namespace std;

int head[N],next[M],to[M],len[M],pr[M];

int n,m,cnt,S,T,mlen;

int tim[1000][1000];

int dis[N],pre[N],q[M];

bool vis[N];

inline void add(int u,int v,int r,int w)

{

to[cnt]=v; len[cnt]=r; pr[cnt]=w; next[cnt]=head[u]; head[u]=cnt++;

to[cnt]=u; len[cnt]=0; pr[cnt]=-w; next[cnt]=head[v]; head[v]=cnt++;

}

inline void read()

{

memset(head,-1,sizeof head); cnt=0;

scanf("%d%d",&m,&n);

S=0; T=n+n\*m+1;

for(int i=1,a;i<=n;i++)

for(int j=1;j<=m;j++)

scanf("%d",&tim[i][j]);

for(int i=1;i<=n;i++) add(S,i,1,0);

for(int i=n+n\*m;i>=n+1;i--) add(i,T,1,0);

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

for(int j=1;j<=m;j++)

for(int k=1;k<=n;k++)

add(i,j\*n+k,1,(n-k+1)\*tim[i][j]);

}

inline bool spfa(){

memset(pre,-1,sizeof(pre));

memset(dis,0x3f,sizeof(dis));

int h=1, t=2, sta;

q[1]=S;dis[S]=0;vis[S]=true;

while (h<t){

sta=q[h++];vis[sta]=false;

for (int i=head[sta];~i;i=next[i])

if (len[i] & dis[to[i]] >dis[sta]+ pr[i]){

dis[to[i]]=dis[sta]+pr[i];

pre[to[i]]=i;

if (!vis[to[i]]) q[t++]=to[i], vis[to[i]]=true;

}

}

return pre[T]!=-1;

}

inline void updata(){

mlen=INF;

for (int i=pre[T];i;i=pre[to[i^1]])

mlen=min(mlen,len[i]);

for (int i=pre[T];i;i=pre[to[i^1]])

len[i]-=mlen, len[i^1]+=mlen;

}

inline void go(){

int ans=0;

while (spfa()) updata(), ans+=dis[T]\*mlen;

printf("%.2lf\n", double(ans)/n);

}

int main(){

read();

go();

return 0;

}

Qsort:

#include <ctime>

#include <cstdio>

#include <cstring>

#include <cstdlib>

#include <iostream>

using namespace std;

int n, a[11000];

void qsort(int l, int r){

int i=l, j=r, x=a[l+r>>1];

while (i<=j){

while (a[i]<x && i<r) i++;

while (a[j]>x && j>l) j--;

if (i<=j) swap(a[i++],a[j--]);

}

if (i<r) qsort(i,r);

if (j>l) qsort(l,j);

}

int main(){

srand(unsigned(time(NULL)));

n=300;

for (int i=1;i<=n;i++) a[i]=rand()%100;

qsort(1,n);

for (int i=1;i<=n;i++) cout<<a[i]<<' ';

return 0;

}

整体二分:

#include <map>

#include <cstdio>

#include <cstring>

#include <cstdlib>

#include <iostream>

#include <algorithm>

#define N 80010

#define S 2000000

using namespace std;

int n, m, T, x, gt, cnt, DFN, LSH;

int a[N], k[N], u[N], v[N], c[N], ans[N], q[N];

int fa[N][21], dep[N], trans[N\*2], lsh[N\*2], nex[N\*3], nu[N\*3], dfn[N][2];

map<int,int> mp;

char s[S+100];

struct qlz\_ques{

int k, u, v, n;

}l[N\*6], b1[N\*6], b2[N\*6];

int read(){

int p=0;

while (s[x]<'0' || s[x]>'9') x++;

while (s[x]>='0' && s[x]<='9') p=p\*10+s[x++]-'0';

return p;

}

void add\_edge(int u, int v){

nex[++cnt]=nex[u];nex[u]=cnt;nu[cnt]=v;

}

void dfs(int u, int father){

dfn[u][0]=++DFN;

fa[u][0]=father;

for (int i=1;fa[fa[u][i-1]][i-1];i++)

fa[u][i]=fa[fa[u][i-1]][i-1];

//cout<<DFN<<' '<<u<<endl;

for (int j=nex[u];j;j=nex[j]){

int v=nu[j];

if (v==father) continue;

dep[v]=dep[u]+1;

dfs(v,u);

}

dfn[u][1]=DFN+1;

}

int LCA(int u, int v){

if (dep[u]<dep[v]) swap(u,v);

//cout<<u<<' '<<v<<endl;

for (int i=20;i>=0;i--)

if (dep[fa[u][i]]>=dep[v]) u=fa[u][i];

if (u==v) return u;

for (int i=20;i>=0;i--)

if (fa[u][i]!=fa[v][i]) u=fa[u][i], v=fa[v][i];

return fa[u][0];

}

void add(int k, int u, int v){

l[++gt].k=k, l[gt].u=u, l[gt].v=v;

}

void update(int u, int v){

for (int i=u;i<=n;i+=i&(-i)) c[i]+=v;

}

int sum(int u){

int p=0;

for (int i=dfn[u][0];i;i-=i&(-i)) p+=c[i];

return p;

}

void solve(int le, int ri, int L, int R){

//cout<<le<<' '<<ri<<' '<<L<<' '<<R<<endl;

if (le>ri) return;

if (L==R){

for (int i=le;i<=ri;i++)

if (l[i].n) ans[l[i].n]=L;

return;

}

int mid=L+R>>1, ct1=0, ct2=0;

for (int i=le;i<=ri;i++){

if (l[i].n){

int u=l[i].u, v=l[i].v, lca=LCA(u,v),k=sum(u)+sum(v)-sum(lca)-sum(fa[lca][0]);

if (k>=l[i].k)

b2[++ct2]=l[i];

else

l[i].k-=k,

b1[++ct1]=l[i];

}

else

if (l[i].v>mid || l[i].v<-mid)

b2[++ct2]=l[i],

update(l[i].u,l[i].v>0?1:-1);

else

b1[++ct1]=l[i];

}

for (int i=1;i<=ct1;i++) l[le+i-1]=b1[i];

for (int i=1;i<=ct2;i++) l[le+ct1+i-1]=b2[i];

for (int i=le;i<=ri;i++)

if (!l[i].n && (l[i].v>mid || l[i].v<-mid))

update(l[i].u,l[i].v>0?-1:1);

solve(le,le+ct1-1,L,mid);

solve(le+ct1,ri,mid+1,R);

}

int main(){

freopen("network10.in","r",stdin);

freopen("整体二分.out","w",stdout);

//read

fread(s,1,S,stdin);

cnt=n=read();m=read();

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

lsh[++LSH]=a[i]=read();

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

int u=read(), v=read();

add\_edge(u,v);

add\_edge(v,u);

}

for (int i=1;i<=m;i++)

k[i]=read(),

u[i]=read(),

v[i]=read(),

(!k[i]?lsh[++LSH]=v[i]:0);

//lsh

dfs(dep[1]=1,0);

sort(lsh+1,lsh+1+LSH);

trans[mp[0]=++T]=0;

for (int i=1;i<=LSH;i++)

if (lsh[i]!=lsh[i-1]) trans[mp[lsh[i]]=++T]=lsh[i];

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

add(0,dfn[i][0],mp[a[i]]),

add(0,dfn[i][1],-mp[a[i]]);

for (int i=1;i<=m;i++)

if (k[i])

add(k[i],u[i],v[i]),

l[gt].n=i,

q[i]=1;

else

add(0,dfn[u[i]][0],-mp[a[u[i]]]),

add(0,dfn[u[i]][1],mp[a[u[i]]]),

add(0,dfn[u[i]][0],mp[a[u[i]]=v[i]]),

add(0,dfn[u[i]][1],-mp[v[i]]);

//work

solve(1,gt,0,T);

//int tot=0;

for (int i=1;i<=m;i++)

if(q[i]){

//tot++;

if (ans[i]) printf("%d\n", trans[ans[i]]);

else printf("invalid request!\n");

}

//cout<<n<<' '<<m<<' '<<tot<<' '<<m-tot<<endl;

return 0;

}

主席树:

#include <map>

#include <cstdio>

#include <cstring>

#include <cstdlib>

#include <iostream>

#include <algorithm>

#define N 80010

#define M 8001000

#define S 2000000

using namespace std;

int n, m, T, x, cnt, DFN, LSH, ct\_in, ct\_out, cnt\_tree;

int f[M], ls[M], rs[M];

int fa[N][21], dep[N], trans[N\*2], lsh[N\*2], a[N], nex[N\*3], nu[N\*3], root[N], bit[N], b1[N\*2], b2[N\*2], dfn[N][2];

char s[S+100];

map<int,int> mp;

struct qlz\_in{

int n, dfn;

}in[N];

struct qlz\_out{

int n, dfn;

}out[N];

struct qlz\_ques{

int k, u, v;

}l[N];

int read(){

int p=0;

while (s[x]<'0' || s[x]>'9') x++;

while (s[x]>='0' && s[x]<='9') p=p\*10+s[x++]-'0';

return p;

}

void add\_edge(int u, int v){

nex[++cnt]=nex[u];nex[u]=cnt;nu[cnt]=v;

}

bool cmp\_in(qlz\_in a, qlz\_in b){return a.dfn<b.dfn;}

bool cmp\_out(qlz\_out a, qlz\_out b){return a.dfn<b.dfn;}

void dfs(int u, int father){

fa[u][0]=father;

for (int i=1;fa[fa[u][i-1]][i-1];i++)

fa[u][i]=fa[fa[u][i-1]][i-1];

in[++ct\_in].dfn=dfn[u][0]=++DFN;

//cout<<DFN<<' '<<u<<endl;

in[ct\_in].n=u;

for (int j=nex[u];j;j=nex[j]){

int v=nu[j];

if (v==father) continue;

dep[v]=dep[u]+1;

dfs(v,u);

}

out[++ct\_out].dfn=dfn[u][1]=DFN+1;

out[ct\_out].n=u;

}

void add\_b1(int u, int &ct1){

if (root[dfn[u][0]]) b1[++ct1]=root[dfn[u][0]];

for (int i=dfn[u][0];i;i-=i&(-i))

if (bit[i]) b1[++ct1]=bit[i];

}

void add\_b2(int u, int &ct2){

if (root[dfn[u][0]]) b2[++ct2]=root[dfn[u][0]];

for (int i=dfn[u][0];i;i-=i&(-i))

if (bit[i]) b2[++ct2]=bit[i];

}

int LCA(int u, int v){

if (dep[u]<dep[v]) swap(u,v);

//cout<<u<<' '<<v<<endl;

for (int i=20;i>=0;i--)

if (dep[fa[u][i]]>=dep[v]) u=fa[u][i];

if (u==v) return u;

for (int i=20;i>=0;i--)

if (fa[u][i]!=fa[v][i]) u=fa[u][i], v=fa[v][i];

return fa[u][0];

}

void solve(int u, int v, int k){

int ct1=0, ct2=0, l=0, r=T, lca=LCA(u,v);

add\_b1(u,ct1);

add\_b1(v,ct1);

add\_b2(lca,ct2);

add\_b2(fa[lca][0],ct2);

//cout<<u<<' '<<v<<' '<<k<<' '<<lca<<endl;

//for (int i=1;i<=ct1;i++) cout<<b1[i]<<' ';cout<<endl;

//for (int i=1;i<=ct2;i++) cout<<b2[i]<<' ';cout<<endl;

while (l<r){

int mid=l+r>>1, p=0;

for (int i=1;i<=ct1;i++) p+=f[rs[b1[i]]];

for (int i=1;i<=ct2;i++) p-=f[rs[b2[i]]];

//cout<<l<<' '<<r<<' '<<mid<<' '<<p<<' '<<k<<endl;

if (p<k){

for (int i=1;i<=ct1;i++)

b1[i]=ls[b1[i]],

(!b1[i]?b1[i--]=b1[ct1--]:0);

for (int i=1;i<=ct2;i++)

b2[i]=ls[b2[i]],

(!b2[i]?b2[i--]=b2[ct2--]:0);

k-=p;

r=mid;

}

else{

for (int i=1;i<=ct1;i++)

b1[i]=rs[b1[i]],

(!b1[i]?b1[i--]=b1[ct1--]:0);

for (int i=1;i<=ct2;i++)

b2[i]=rs[b2[i]],

(!b2[i]?b2[i--]=b2[ct2--]:0);

l=mid+1;

}

}

if (l) printf("%d\n", trans[l]);

else printf("invalid request!\n");

}

void update(int x, int y, int z){

int ct=0, l=0, r=T;

for (int i=x;i<=DFN;i+=i&(-i)){

if (!bit[i]) bit[i]=++cnt\_tree;

f[b1[++ct]=bit[i]]+=z;

}

while (l<r){

int mid=l+r>>1;

if (y<=mid){

r=mid;

for (int i=1;i<=ct;i++){

if (!ls[b1[i]]) ls[b1[i]]=++cnt\_tree;

f[b1[i]=ls[b1[i]]]+=z;

}

}

else{

l=mid+1;

for (int i=1;i<=ct;i++){

if (!rs[b1[i]]) rs[b1[i]]=++cnt\_tree;

f[b1[i]=rs[b1[i]]]+=z;

}

}

}

}

int main(){

freopen("network10.in","r",stdin);

freopen("p1146\_主席树静态建树查询优化.out","w",stdout);

//read

fread(s,1,S,stdin);

cnt=n=read();m=read();

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

lsh[++LSH]=a[i]=read();

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

int u=read(), v=read();

add\_edge(u,v);

add\_edge(v,u);

}

for (int i=1;i<=m;i++)

l[i].k=read(),

l[i].u=read(),

l[i].v=read(),

(!l[i].k?lsh[++LSH]=l[i].v:0);

//lsh

sort(lsh+1,lsh+1+LSH);

trans[mp[0]=++T]=0;

for (int i=1;i<=LSH;i++)

if (lsh[i]!=lsh[i-1]) trans[mp[lsh[i]]=++T]=lsh[i];

for (int i=1;i<=n;i++) a[i]=mp[a[i]];

//build

dfs(dep[1]=1,0);

sort(in+1,in+1+n,cmp\_in);

sort(out+1,out+1+n,cmp\_out);

int j=1;

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

int k=root[in[i].dfn]=++cnt\_tree, kk=root[in[i].dfn-1], l=0, r=T, v=a[in[i].n];

while (l<r){

int mid=l+r>>1;

if (v<=mid)

rs[k]=rs[kk],

f[k=ls[k]=++cnt\_tree]=f[kk=ls[kk]]+1,

r=mid;

else

ls[k]=ls[kk],

f[k=rs[k]=++cnt\_tree]=f[kk=rs[kk]]+1,

l=mid+1;

}

while (out[j].dfn==in[i].dfn){

kk=root[in[i].dfn], k=root[in[i].dfn]=++cnt\_tree, l=0, r=T, v=a[out[j++].n];

while (l<r){

int mid=l+r>>1;

if (v<=mid)

rs[k]=rs[kk],

f[k=ls[k]=++cnt\_tree]=f[kk=ls[kk]]-1,

r=mid;

else

ls[k]=ls[kk],

f[k=rs[k]=++cnt\_tree]=f[kk=rs[kk]]-1,

l=mid+1;

}

}

}

//work

for (int i=1;i<=m;i++)

if (l[i].k)

solve(l[i].u,l[i].v,l[i].k);

else{

int u=l[i].u, v=mp[l[i].v];

update(dfn[u][0],a[u],-1);

update(dfn[u][1],a[u],1);

update(dfn[u][0],a[u]=v,1);

update(dfn[u][1],a[u],-1);

}

return 0;

}

Cdq(三维偏序):

#include <cstdio>

#include <cstring>

#include <cstdlib>

#include <iostream>

#include <algorithm>

#define N 600

#define M 500000

using namespace std;

int n, m, x, cnt, ans[M], q[M], c[N][N];

char s[6000010];

struct qlz{

    int n, v, x, y, c, x1, x2, y1, y2;

}l[M], b1[M], b2[M];

inline int read(){

    int p=0;

    while (s[x]<'0' || s[x]>'9') x++;

    while (s[x]>='0' && s[x]<='9') p=p\*10+s[x++]-'0';

    return p;

}

inline bool cmp(qlz a, qlz b){return a.c<b.c;}

inline void update(int x, int y, int z){

    for (int i=x;i<=n;i+=i&(-i))

        for (int j=y;j<=n;j+=j&(-j))

            c[i][j]+=z;

}

inline int sum(int x, int y){

    int p=0;

    for (int i=x;i;i-=i&(-i))

        for (int j=y;j;j-=j&(-j))

            p+=c[i][j];

    return p;

}

inline void solve(int le, int ri, int L, int R){

    if (le>ri) return;

    if (L==R){

        for (int i=le;i<=ri;i++)

            if (!l[i].v) ans[l[i].n]=L;

        return;

    }

    int mid=L+R>>1;

    int ct1=0, ct2=0;

    for (int i=le;i<=ri;i++)

        if (l[i].v){

            if (l[i].v<=mid)

                b1[++ct1]=l[i],

                update(l[i].x,l[i].y,1);

            else

                b2[++ct2]=l[i];

        }

        else{

            int k=sum(l[i].x2,l[i].y2)+sum(l[i].x1-1,l[i].y1-1)-sum(l[i].x1-1,l[i].y2)-sum(l[i].x2,l[i].y1-1);

            if (k>=l[i].c)

                b1[++ct1]=l[i];

            else

                l[i].c-=k,

                b2[++ct2]=l[i];

        }

    for (int i=1;i<=ct1;i++) l[le+i-1]=b1[i];

    for (int i=1;i<=ct2;i++) l[le+ct1+i-1]=b2[i];

    //memcpy(l+le,b1+1,sizeof(l[0])\*ct1);

    //memcpy(l+le+ct1,b2+1,sizeof(l[0])\*ct2);

    for (int i=le;i<=ri;i++)

        if (l[i].v && l[i].v<=mid) update(l[i].x,l[i].y,-1);

    solve(le,le+ct1-1,L,mid);

    solve(le+ct1,ri,mid+1,R);

}

int main(){

    fread(s,1,6000000,stdin);

    n=read();m=read();

    for (int i=1;i<=n;i++)

        for (int j=1;j<=n;j++)

            l[++cnt].c=read(),

            l[cnt].x=i,

            l[cnt].y=j;

    sort(l+1,l+1+cnt,cmp);

    for (int i=1;i<=cnt;i++) q[l[i].v=i]=l[i].c;

    for (int i=1;i<=m;i++)

        l[++cnt].x1=read(),

        l[cnt].y1=read(),

        l[cnt].x2=read(),

        l[cnt].y2=read(),

        l[cnt].c=read(),

        l[cnt].n=i;

    solve(1,cnt,1,n\*n);

    for (int i=1;i<=m;i++) printf("%d\n", q[ans[i]]);

    return 0;

}

Kmp:

#include <cstdio>

#include <cstring>

#include <cstdlib>

#include <iostream>

#define N 1010000

#define mo 1000000007

typedef long long ll;

using namespace std;

int ls, n, f[N], p[N];

char s[N];

int read(){

    int p=0;

    char ch=getchar();

    while (ch<'0' || ch>'9') ch=getchar();

    while (ch>='0' && ch<='9') p=p\*10+ch-'0', ch=getchar();

    return p;

}

void pre(){

    ls=strlen(s+1);

    int j=0;

    f[1]=1;

    for (int i=2;i<=ls;i++){

        while (j && s[j+1]!=s[i]) j=p[j];

        f[i]=f[p[i]=j+=s[j+1]==s[i]]+1;

    }

}

void solve(){

    ll ans=1;

    int j=0;

    for (int i=2;i<=ls;i++){

        while (j && s[j+1]!=s[i]) j=p[j];

        if (s[j+1]==s[i]) j++;

        while ((j<<1)>i && j) j=p[j];

        ans=ans\*(f[j]+1)%mo;

    }

    cout<<ans<<endl;

}

void \_\_init(){

    for (int i=read();i;i--){

        scanf("%s", s+1);

        pre();

        solve();

    }

}

int main(){

    \_\_init();

    return 0;

}