//https://ideone.com/XEMlTq

// DSU

void init()

{

lli i,j,k;

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

{

parent[i]=i;

siz[i]=1;

}

}

lli root(lli a)

{

while(parent[a]!=a)

{

parent[a]=parent[parent[a]];

a=parent[a];

}

return a;

}

void make\_union(lli root\_a,lli root\_b)

{

if(siz[root\_a]<siz[root\_b])

{

parent[root\_a]=root\_b;

siz[root\_b]+=siz[root\_a];

}

else

{

parent[root\_b]=root\_a;

siz[root\_a]+=siz[root\_b];

}

}

// seg tree with lazy

#include<bits/stdc++.h>

typedef long long int lli;

using namespace std;

typedef unsigned long long int ulli;

const lli mod=1e9+7;

const lli N = 2e5 + 9;

lli primes[6]={1125899906842597,1495921043,1005985879,1495921043,1005985879,1495921043};

#include<stdio.h>

#include <sstream>

#define pi 3.141592653589793238462

vector<lli>adj[N];

lli parent[N];

lli vis[N];

lli level[N];

lli dist[N];

lli dp[N];

lli arr[1000009];

lli brr[1000009];

lli crr[1000009];

lli hashing[N];

lli ar[509][509];

lli br[509][509];

lli cr[509][509];

lli multiply(lli a,lli b){return ((a%mod)\*(b%mod))%mod;}

lli add(lli a,lli b){return ((a%mod)+(b%mod))%mod;}

lli sub(lli a,lli b){return ((a%mod)-(b%mod)+mod)%mod;}

#define deb(x) cout<< #x << " " << x <<endl;

lli dx[] = {1, -1, 0, 0};

lli dy[] = {0, 0, 1, -1};

//deb(n);

lli siz[N];

lli pos[N];

lli cnter;

lli tree[4\*N][2];

lli lazy[4\*N][2];

void dfs(lli val)

{

lli i;

cnter++;

pos[val]=cnter;

vis[val]=1;

siz[val]=1;

for(i=0;i<adj[val].size();i++)

{

if(vis[adj[val][i]]==0)

{

dfs(adj[val][i]);

siz[val]+=siz[adj[val][i]];

}

}

}

void build\_tree(lli s,lli e,lli index)

{

if(s==e)

{

tree[index][1]=1;

tree[index][0]=0;

return ;

}

lli mid=(s+e)/2;

build\_tree(s,mid,2\*index);

build\_tree(mid+1,e,2\*index+1);

tree[index][1]=tree[2\*index][1]+tree[2\*index+1][1];

tree[index][0]=tree[2\*index][0]+tree[2\*index+1][0];

}

void update(lli s,lli e,lli index,lli qs,lli qe,lli type)

{

//cout<<"s="<<s<<" "<<"E="<<e<<" "<<"ind="<<index<<" "<<"qs="<<qs<<" "<<"qe="<<qe<<" "<<"typ="<<type<<"\n";

if(lazy[index][0]!=0 || lazy[index][1]!=0)

{

tree[index][0]=lazy[index][0];

tree[index][1]=lazy[index][1];

if(s!=e)

{

lli half=(s+e)/2;

if(lazy[index][0]!=0)

{

lazy[2\*index][0]=(half-s+1);

lazy[2\*index+1][0]=(e-(half+1)+1);

lazy[2\*index][1]=lazy[2\*index+1][1]=0;

}

else

{

lazy[2\*index][1]=(half-s+1);

lazy[2\*index+1][1]=(e-(half+1)+1);

lazy[2\*index][0]=lazy[2\*index+1][0]=0;

}

}

lazy[index][0]=lazy[index][1]=0;

}

if(e<qs || s>qe)

{

return ;

}

if(qs<=s && qe>=e)

{

lli half=(s+e)/2;

if(type==0)

{

// sblo sulla do;

tree[index][1]=0;

tree[index][0]=(e-s+1);

if(s!=e)

{

lazy[2\*index][0]=(half-s+1);

lazy[2\*index+1][0]=(e-(half+1)+1);

lazy[2\*index][1]=lazy[2\*index+1][1]=0;

}

}

else

{

//sbko jagga do;

tree[index][1]=(e-s+1);

tree[index][0]=0;

if(s!=e)

{

lazy[2\*index][1]=(half-s+1);

lazy[2\*index+1][1]=(e-(half+1)+1);

lazy[2\*index][0]=lazy[2\*index+1][0]=0;

}

}

return ;

}

lli mid=(s+e)/2;

update(s,mid,2\*index,qs,qe,type);

update(mid+1,e,2\*index+1,qs,qe,type);

tree[index][0]=tree[2\*index][0]+tree[2\*index+1][0];

tree[index][1]=tree[2\*index][1]+tree[2\*index+1][1];

}

lli query(lli s,lli e,lli index,lli qs,lli qe)

{

if(lazy[index][0]!=0 || lazy[index][1]!=0)

{

tree[index][0]=lazy[index][0];

tree[index][1]=lazy[index][1];

if(s!=e)

{

lli half=(s+e)/2;

if(lazy[index][0]!=0)

{

lazy[2\*index][0]=(half-s+1);

lazy[2\*index+1][0]=(e-(half+1)+1);

lazy[2\*index][1]=lazy[2\*index+1][1]=0;

}

else

{

lazy[2\*index][1]=(half-s+1);

lazy[2\*index+1][1]=(e-(half+1)+1);

lazy[2\*index][0]=lazy[2\*index+1][0]=0;

}

}

lazy[index][0]=lazy[index][1]=0;

}

if(s>qe || e<qs)

{

return 0;

}

if(qs<=s && qe>=e)

{

return tree[index][1];

}

lli mid=(s+e)/2;

lli lans=0,rans=0;

lans=query(s,mid,2\*index,qs,qe);

rans=query(mid+1,e,2\*index+1,qs,qe);

return lans+rans;

}

int main()

{

//freopen("input.txt","r",stdin);

//freopen("output.txt","w",stdout);

int start\_s = clock();

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL); cout.tie(NULL);

vector<lli>vect,vect1,vect2,vt,vct;

vector<lli>::iterator itt;

set<lli>st,stt,st1,st2;

set<lli>::iterator itr;

map<lli,lli>mp,mp1,mp2,mpp;

map<lli,lli>::iterator it;

lli i,j,k,n,m,q,t,a,d,b,c,l,r,e,idx,ind,index,u,v,x,y,z,h,sz,sz1,sz2,mid,len,tot,prev,temp,curr,p;

lli res=0,res1=0,res2=0,ans=0,ans1=0,ele,ans2=0,val=0,val1=0,val2=0,rem=0,diff=0,cnt=0,flag=0,fl=0,sum=0,maxi=INT\_MIN,mini=INT\_MAX,total=0;

string str,str1,str2;

char ch,ch1,ch2;

cin>>n;

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

{

cin>>ele;

if(ele!=0){

adj[i].push\_back(ele);

adj[ele].push\_back(i);}

else

{

idx=i;

}

}

dfs(idx);

/\*

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

{

cout<<"i="<<i<<" "<<"pos="<<pos[i]<<" "<<"siz="<<siz[i]<<"\n";

}

\*/

build\_tree(1,n,1);

cin>>q;

lli id,type;

while(q--)

{

cin>>type>>id;

if(type==1)

{

//wake\_up;

l=pos[id]+1;

r=pos[id]+siz[id]-1;

if(r<l)

continue;

update(1,n,1,l,r,1);

}

else if(type==2)

{

//sleep;

l=pos[id]+1;

r=pos[id]+siz[id]-1;

if(r<l)

continue;

update(1,n,1,l,r,0);

}

else

{

//count of awake

l=pos[id]+1;

r=pos[id]+siz[id]-1;

if(r<l)

cout<<0<<"\n";

else

cout<<query(1,n,1,l,r)<<"\n";

}

}

}

// BIT

#include "bits/stdc++.h"

#include<stdio.h>

#include <sstream>

#define pi 3.141592653589793238462

#define limit 100005

#define mo 1000000007;

typedef long long int lli;

typedef unsigned long long int ulli;

const lli mod=9999999999999983;

lli primes[6]={1125899906842597,1495921043,1005985879,1495921043,1005985879,1495921043};

using namespace std;

vector<lli>adj[2000009];

lli parent[2000009];

lli vis[2000009];

lli level[2000009];

lli dist[2000009];

lli dp[100009];

lli arr[100009];

lli brr[2000009];

lli crr[2000009];

lli hashing[2000009];

lli ar[509][509];

lli br[509][509];

lli cr[509][509];

lli multiply(lli a,lli b){return ((a%mod)\*(b%mod))%mod;}

lli add(lli a,lli b){return ((a%mod)+(b%mod))%mod;}

lli sub(lli a,lli b){return ((a%mod)-(b%mod)+mod)%mod;}

#define deb(x) cout<< #x << " " << x <<endl;

//deb(n);

lli tree[2000009];

lli finalans;

lli cnt;

lli power(lli x, lli y,lli p)

{

lli res=1;

x=x%p;

while(y>0)

{

if(y&1)

res=((res%p)\*(x%p))%p;

y=y>>1;

x=((x%p)\*(x%p))%p;

}

return res;

}

void update(lli ind)

{

while(ind<=1e6)

{

tree[ind]++;

ind+=ind&(-ind);

}

}

void upd(lli ind)

{

while(ind<=1e6)

{

tree[ind]--;

ind+=ind&(-ind);

}

}

lli query(lli ind)

{

lli sum=0;

while(ind>0)

{

sum+=tree[ind];

ind-=ind&(-ind);

}

return sum;

}

void dfs(lli val)

{

lli i,ans;

vis[val]=1;

cnt++;

ans=query(arr[val]);

finalans+=arr[val]\*((cnt-1)-ans);

update(arr[val]);

//cout<<"val bef="<<val<<endl;

for(i=0;i<adj[val].size();i++)

{

if(vis[adj[val][i]]==0)

{

parent[adj[val][i]]=val;

dfs(adj[val][i]);

}

}

cnt--;

upd(arr[val]);

//cout<<"val after"<<val<<endl;

}

int main()

{

//freopen("input.txt","r",stdin);

//freopen("output.txt","w",stdout);

int start\_s = clock();

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL); cout.tie(NULL);

vector<lli>vect,vt,vct;

vector<lli>::iterator itt;

set<lli>st;

set<lli>::iterator itr;

map<lli,lli>mp,mpp,mp1,mp2;

map<lli,lli>::iterator it;

lli i,j,k,n,m,p,res1,res2,q,t,a,b,c,maxi=INT\_MIN,mini=INT\_MAX,sum=0,ans=0,res=0,val=0,ans1=0,ans2=0,rem=0,diff=0,cnt=0,l,r,flag=0,e,index,val1=0,val2=0,z,h=0,u,v,mid,len,tot,fl=0;

string str,str1,str2;

cin>>n;

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

{

cin>>u>>v;

adj[u].push\_back(v);

adj[v].push\_back(u);

}

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

{

cin>>arr[i];

}

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

{

if(vis[i]==0)

{

dfs(i);

}

}

cout<<finalans;

}

// lowest common ancestor

#include "bits/stdc++.h"

#include<stdio.h>

#include <sstream>

#define pi 3.141592653589793238462

#define limit 100005

#define mo 1000000007;

typedef long long int lli;

typedef unsigned long long int ulli;

const lli mod=9999999999999983;

lli primes[6]={1125899906842597,1495921043,1005985879,1495921043,1005985879,1495921043};

using namespace std;

vector<lli>adj[2000009];

lli parent[200009][30];

lli vis[2000009];

lli level[2000009];

lli dist[2000009];

lli dp[100009];

lli arr[100009];

lli brr[2000009];

lli crr[2000009];

lli hashing[2000009];

lli ar[509][509];

lli br[509][509];

lli cr[509][509];

lli multiply(lli a,lli b){return ((a%mod)\*(b%mod))%mod;}

lli add(lli a,lli b){return ((a%mod)+(b%mod))%mod;}

lli sub(lli a,lli b){return ((a%mod)-(b%mod)+mod)%mod;}

#define deb(x) cout<< #x << " " << x <<endl;

//deb(n);

lli power(lli x, lli y,lli p)

{

lli res=1;

x=x%p;

while(y>0)

{

if(y&1)

res=((res%p)\*(x%p))%p;

y=y>>1;

x=((x%p)\*(x%p))%p;

}

return res;

}

void dfs(lli val)

{

lli i;

vis[val]=1;

// cout<<"val="<<val<<endl;

for(i=0;i<adj[val].size();i++)

{

if(vis[adj[val][i]]==0)

{

level[adj[val][i]]=level[val]+1;

parent[adj[val][i]][0]=val;

dfs(adj[val][i]);

}

}

}

lli lca(lli u,lli v)

{

lli i;

if(u==v)

{

return u;

}

if(level[u]<level[v])

{

swap(u,v);

}

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

{

if(level[u]-(1<<i)>=level[v])

{

u=parent[u][i];

}

}

if(u==v)

return u;

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

{

if(parent[u][i]!=-1&&parent[v][i]!=-1&&parent[u][i]!=parent[v][i])

{

u=parent[u][i];

v=parent[v][i];

}

}

return parent[u][0];

}

int main()

{

//freopen("input.txt","r",stdin);

//freopen("output.txt","w",stdout);

int start\_s = clock();

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL); cout.tie(NULL);

vector<lli>vect,vt,vct;

vector<lli>::iterator itt;

set<lli>st;

set<lli>::iterator itr;

map<lli,lli>mp,mpp,mp1,mp2;

map<lli,lli>::iterator it;

lli i,j,k,n,m,p,res1,res2,q,t,a,b,c,maxi=INT\_MIN,mini=INT\_MAX,sum=0,ans=0,res=0,val=0,ans1=0,ans2=0,rem=0,diff=0,cnt=0,l,r,flag=0,e,index,val1=0,val2=0,z,h=0,u,v,mid,len,tot,fl=0;

string str,str1,str2;

cin>>n;

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

{

cin>>e;

adj[i].push\_back(e);

adj[e].push\_back(i);

}

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

dfs(0);

for(i=1;i<=20;i++)

{

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

{

if(parent[j][i-1]!=-1)

parent[j][i]=parent[parent[j][i-1]][i-1];

}

}

cin>>m;

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

{

cin>>brr[i];

}

if(m==1)

{

cout<<m;

}

else

{

val=lca(brr[1],brr[2]);

for(i=3;i<=m;i++)

{

val=lca(val,brr[i]);

}

cout<<val;

}

}

// dijkstra

#include "bits/stdc++.h"

#include<stdio.h>

#include <sstream>

#define pi 3.141592653589793238462

#define limit 100005

#define mo 1000000007;

typedef long long int lli;

typedef unsigned long long int ulli;

const lli mod=9999999999999983;

lli primes[6]={1125899906842597,1495921043,1005985879,1495921043,1005985879,1495921043};

using namespace std;

vector< pair<lli,lli> >adj[2000009];

lli parent[2000009];

lli vis[2000009];

lli level[2000009];

lli dist[2000009];

lli dp[100009];

lli arr[100009];

lli brr[2000009];

lli crr[2000009];

lli hashing[2000009];

lli ar[509][509];

lli br[509][509];

lli cr[509][509];

lli multiply(lli a,lli b){return ((a%mod)\*(b%mod))%mod;}

lli add(lli a,lli b){return ((a%mod)+(b%mod))%mod;}

lli sub(lli a,lli b){return ((a%mod)-(b%mod)+mod)%mod;}

#define deb(x) cout<< #x << " " << x <<endl;

//deb(n);

lli n;

void djk(lli s,lli e)

{

lli i,index,ind,w,wei;

memset(vis,0,sizeof(vis));

multiset< pair<lli,lli> >ms;

pair<lli,lli>p;

ms.insert({0,s});

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

dist[i]=INT\_MAX;

dist[s]=0;

while(ms.size()>0)

{

p=\*ms.begin();

ms.erase(ms.begin());

index=p.second;

wei=p.first;

if(vis[index]==1)

continue;

vis[index]=1;

for(i=0;i<adj[index].size();i++)

{

ind=adj[index][i].second;

w=adj[index][i].first;

if(dist[index]+w<dist[ind])

{

dist[ind]=dist[index]+w;

ms.insert({dist[ind],ind});

}

}

}

if(dist[e]<INT\_MAX)

cout<<dist[e]<<"\n";

else

cout<<"NONE\n";

}

int main()

{

//freopen("input.txt","r",stdin);

//freopen("output.txt","w",stdout);

int start\_s = clock();

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL); cout.tie(NULL);

vector<lli>vect,vt,vct;

vector<lli>::iterator itt;

set<lli>st;

set<lli>::iterator itr;

map<lli,lli>mp,mpp,mp1,mp2;

map<lli,lli>::iterator it;

lli i,j,k,m,p,res1,res2,q,t,a,b,c,maxi=INT\_MIN,mini=INT\_MAX,sum=0,ans=0,res=0,val=0,ans1=0,ans2=0,rem=0,diff=0,cnt=0,l,r,flag=0,e,index,val1=0,val2=0,z,h=0,u,v,mid,len,tot,fl=0;

string str,str1,str2;

lli s,wei;

cin>>t;

while(t--)

{

cin>>n>>m>>s>>e;

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

{

cin>>u>>v>>wei;

adj[u].push\_back({wei,v});

adj[v].push\_back({wei,u});

}

djk(s,e);

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

adj[i].clear();

}

}

//BELLMAN - FORD

vector <int> v [2000 + 10];

int dis [1000 + 10];

for(int i = 0; i < m + 2; i++){

v[i].clear();

dis[i] = 2e9;

}

for(int i = 0; i < m; i++){

scanf("%d%d%d", &from , &next , &weight);

v[i].push\_back(from);

v[i].push\_back(next);

v[i].push\_back(weight);

}

dis[0] = 0;

for(int i = 0; i < n - 1; i++){

int j = 0;

while(v[j].size() != 0){

if(dis[ v[j][0] ] + v[j][2] < dis[ v[j][1] ] ){

dis[ v[j][1] ] = dis[ v[j][0] ] + v[j][2];

}

j++;

}

}

// prim

#include "bits/stdc++.h"

#include<stdio.h>

#include <sstream>

#define pi 3.141592653589793238462

#define limit 100005

#define mo 1000000007;

typedef long long int lli;

typedef unsigned long long int ulli;

const lli mod=9999999999999983;

lli primes[6]={1125899906842597,1495921043,1005985879,1495921043,1005985879,1495921043};

using namespace std;

vector< pair<lli,lli> >adj[2000009];

lli parent[2000009];

lli vis[2000009];

lli level[2000009];

lli dist[2000009];

lli dp[100009];

lli arr[100009];

lli brr[2000009];

lli crr[2000009];

lli hashing[2000009];

lli ar[509][509];

lli br[509][509];

lli cr[509][509];

lli multiply(lli a,lli b){return ((a%mod)\*(b%mod))%mod;}

lli add(lli a,lli b){return ((a%mod)+(b%mod))%mod;}

lli sub(lli a,lli b){return ((a%mod)-(b%mod)+mod)%mod;}

#define deb(x) cout<< #x << " " << x <<endl;

//deb(n);

lli n,m,pp;

void prim()

{

lli ans=0,index,wei,i;

pair<lli,lli>p;

multiset< pair<lli,lli> >ms;

ms.insert({0,1});

while(ms.size()>0)

{

p=\*ms.begin();

ms.erase(ms.begin());

index=p.second;

wei=p.first;

if(vis[index]==1)

continue;

vis[index]=1;

ans+=wei;

for(i=0;i<adj[index].size();i++)

{

if(vis[adj[index][i].second]==0)

{

ms.insert({adj[index][i].first,adj[index][i].second});

}

}

}

cout<<ans\*pp<<"\n";

}

int main()

{

//freopen("input.txt","r",stdin);

//freopen("output.txt","w",stdout);

int start\_s = clock();

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL); cout.tie(NULL);

vector<lli>vect,vt,vct;

vector<lli>::iterator itt;

set<lli>st;

set<lli>::iterator itr;

map<lli,lli>mp,mpp,mp1,mp2;

map<lli,lli>::iterator it;

lli i,j,k,res1,res2,q,t,a,b,c,maxi=INT\_MIN,mini=INT\_MAX,sum=0,ans=0,res=0,val=0,ans1=0,ans2=0,rem=0,diff=0,cnt=0,l,r,flag=0,e,index,val1=0,val2=0,z,h=0,u,v,mid,len,tot,fl=0;

string str,str1,str2;

lli wei;

cin>>t;

while(t--)

{

cin>>pp;

cin>>n;

cin>>m;

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

{

cin>>u>>v>>wei;

adj[u].push\_back({wei,v});

adj[v].push\_back({wei,u});

}

prim();

}

}

// kruskal

#include "bits/stdc++.h"

#include<stdio.h>

#include <sstream>

#define pi 3.141592653589793238462

#define limit 100005

#define mo 1000000007;

typedef long long int lli;

typedef unsigned long long int ulli;

const lli mod=9999999999999983;

lli primes[6]={1125899906842597,1495921043,1005985879,1495921043,1005985879,1495921043};

using namespace std;

vector<lli>adj[2000009];

lli parent[2000009];

lli vis[2000009];

lli level[2000009];

lli dist[2000009];

lli dp[100009];

lli arr[100009];

lli brr[2000009];

lli crr[2000009];

lli hashing[2000009];

lli ar[509][509];

lli br[509][509];

lli cr[509][509];

lli multiply(lli a,lli b){return ((a%mod)\*(b%mod))%mod;}

lli add(lli a,lli b){return ((a%mod)+(b%mod))%mod;}

lli sub(lli a,lli b){return ((a%mod)-(b%mod)+mod)%mod;}

#define deb(x) cout<< #x << " " << x <<endl;

lli siz[2000009];

vector< pair< lli,pair<lli,lli> > >vect,vt,vct;

void initialize(lli n)

{

lli i;

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

{

parent[i]=i;

siz[i]=1;

}

}

lli root(lli a)

{

while(parent[a]!=a)

{

parent[a]=parent[parent[a]];

a=parent[a];

}

return a;

}

void make\_union(lli root\_a,lli root\_b)

{

if(siz[root\_a]<siz[root\_b])

{

siz[root\_b]+=siz[root\_a];

parent[root\_a]=root\_b;

}

else

{

siz[root\_a]+=siz[root\_b];

parent[root\_b]=root\_a;

}

}

void mst()

{

lli i,j,k,u,v,ans=0,wei,root\_a,root\_b;

for(i=0;i<vect.size();i++)

{

wei=vect[i].first;

u=vect[i].second.first;

v=vect[i].second.second;

root\_a=root(u);

root\_b=root(v);

if(root\_a!=root\_b)

{

make\_union(root\_a,root\_b);

ans+=wei;

}

}

cout<<ans;

}

int main()

{

//freopen("input.txt","r",stdin);

//freopen("output.txt","w",stdout);

int start\_s = clock();

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL); cout.tie(NULL);

//vector<lli>vect,vt,vct;

//vector<lli>::iterator itt;

set<lli>st;

set<lli>::iterator itr;

map<lli,lli>mp,mpp,mp1,mp2;

map<lli,lli>::iterator it;

lli i,j,k,n,m,p,wei,res1,res2,q,t,a,b,c,maxi=INT\_MIN,mini=INT\_MAX,sum=0,ans=0,res=0,val=0,ans1=0,ans2=0,rem=0,diff=0,cnt=0,l,r,flag=0,e,index,val1=0,val2=0,z,h=0,u,v,mid,len,tot,fl=0;

string str,str1,str2;

cin>>n>>m;

initialize(n);

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

{

cin>>u>>v>>wei;

vect.push\_back({wei,{u,v}});

}

sort(vect.begin(),vect.end());

mst();

}

// Articulation point

#include "bits/stdc++.h"

#include<stdio.h>

#include <sstream>

#define pi 3.141592653589793238462

#define limit 100005

#define mo 1000000007;

typedef long long int lli;

typedef unsigned long long int ulli;

const lli mod=9999999999999983;

lli primes[6]={1125899906842597,1495921043,1005985879,1495921043,1005985879,1495921043};

using namespace std;

vector<lli>adj[2000009];

lli parent[2000009];

lli vis[2000009];

lli level[2000009];

lli dist[2000009];

lli dp[100009];

lli arr[100009];

lli brr[2000009];

lli crr[2000009];

lli hashing[2000009];

lli ar[509][509];

lli br[509][509];

lli cr[509][509];

lli multiply(lli a,lli b){return ((a%mod)\*(b%mod))%mod;}

lli add(lli a,lli b){return ((a%mod)+(b%mod))%mod;}

lli sub(lli a,lli b){return ((a%mod)-(b%mod)+mod)%mod;}

#define deb(x) cout<< #x << " " << x <<endl;

//deb(n);

lli cnt;

lli low[10009];

lli disc[10009];

lli ap[10009];

lli power(lli x, lli y,lli p)

{

lli res=1;

x=x%p;

while(y>0)

{

if(y&1)

res=((res%p)\*(x%p))%p;

y=y>>1;

x=((x%p)\*(x%p))%p;

}

return res;

}

void dfs(lli val)

{

lli i;

lli children=0;

vis[val]=1;

disc[val]=low[val]=++cnt;

for(i=0;i<adj[val].size();i++)

{

if(vis[adj[val][i]]==0)

{

children++;

parent[adj[val][i]]=val;

dfs(adj[val][i]);

low[val]=min(low[val],low[adj[val][i]]);

if(parent[val]!=0 && low[adj[val][i]]>=disc[val])

{

ap[val]=1;

}

if(parent[val]==0 && children>=2)

{

ap[val]=1;

}

}

else if(adj[val][i]!=parent[val])

{

low[val]=min(low[val],disc[adj[val][i]]);

}

}

}

int main()

{

//freopen("input.txt","r",stdin);

//freopen("output.txt","w",stdout);

int start\_s = clock();

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL); cout.tie(NULL);

vector<lli>vect,vt,vct;

vector<lli>::iterator itt;

set<lli>st;

set<lli>::iterator itr;

map<lli,lli>mp,mpp,mp1,mp2;

map<lli,lli>::iterator it;

lli i,j,k,n,m,p,res1,res2,q,t,a,b,c,maxi=INT\_MIN,mini=INT\_MAX,sum=0,ans=0,res=0,val=0,ans1=0,ans2=0,rem=0,diff=0,l,r,flag=0,e,index,val1=0,val2=0,z,h=0,u,v,mid,len,tot,fl=0;

string str,str1,str2;

while(1)

{

cin>>n>>m;

if(n==0&&m==0)

break;

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

{

cin>>u>>v;

adj[u].push\_back(v);

adj[v].push\_back(u);

}

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

{

if(vis[i]==0)

{

dfs(i);

}

}

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

{

if(ap[i]){

//cout<<"i="<<i<<endl;

ans++;}

}

cout<<ans<<"\n";

cnt=0;

ans=0;

memset(parent,0,sizeof(parent));

memset(disc,0,sizeof(disc));

memset(low,0,sizeof(low));

memset(ap,0,sizeof(ap));

memset(vis,0,sizeof(vis));

for(i=1;i<=100009;i++)

adj[i].clear();

}

}

// AP + bridges

#include "bits/stdc++.h"

#include<stdio.h>in

#include <sstream>

#define pi 3.141592653589793238462

#define limit 100005

#define mo 1000000007;

typedef long long int lli;

typedef unsigned long long int ulli;

const lli mod=9999999999999983;

lli primes[6]={1125899906842597,1495921043,1005985879,1495921043,1005985879,1495921043};

using namespace std;

vector<lli>adj[2000009];

lli parent[2000009];

lli vis[2000009];

lli level[2000009];

lli dist[2000009];

lli dp[100009];

lli arr[100009];

lli brr[2000009];

lli crr[2000009];

lli hashing[2000009];

lli ar[509][509];

lli br[509][509];

lli cr[509][509];

lli multiply(lli a,lli b){return ((a%mod)\*(b%mod))%mod;}

lli add(lli a,lli b){return ((a%mod)+(b%mod))%mod;}

lli sub(lli a,lli b){return ((a%mod)-(b%mod)+mod)%mod;}

#define deb(x) cout<< #x << " " << x <<endl;

//deb(n);

lli disc[2000009];

lli low[2000009];

vector< pair<lli,lli> >vect;

vector< lli>vt;

lli n;

void init()

{

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

{

disc[i] = 0;

low[i] = INT\_MAX;

vis[i] = false;

parent[i] = -1;

}

}

void dfs(lli val)

{

static lli tim=0;

lli i; io

vis[val]=1;

lli child=0;

disc[val]=low[val]=tim++;

for(i=0;i<adj[val].size();i++)

{

//child++;

if(vis[adj[val][i]]==0)

{

child++;

parent[adj[val][i]]=val;

dfs(adj[val][i]);

low[val]=min(low[val],low[adj[val][i]]);

// AP condition 1 & 2

// bridge condition 3

if(parent[val]!=-1 && disc[val]<=low[adj[val][i]]) //1

{

vt.push\_back(val);

}

if(parent[val]==-1 && child>=2) //2

{

vt.push\_back(val);

}

if(disc[val]<low[adj[val][i]]) //3

{

if(val<adj[val][i])

vect.push\_b66ack({val,adj[val][i]});

else

vect.push\_back({adj[val][i],val});

}

}

else if(adj[val][i]!=parent[val])

{

low[val]=min(low[val],disc[adj[val][i]]);

}

}

}

lli power(lli x, lli y,lli p)

{

lli res=1;

x=x%p;

while(y>0)

{

if(y&1)

res=((res%p)\*(x%p))%p;

y=y>>1;

x=((x%p)\*(x%p))%p;

}

return res;

}

int main()

{

//freopen("input.txt","r",stdin);

//freopen("output.txt","w",stdout);

int start\_s = clock();

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL); cout.tie(NULL);

vector<lli>::iterator itt;

set<lli>st;

set<lli>::iterator itr;

map<lli,lli>mp,mpp,mp1,mp2;

map<lli,lli>::iterator it;

lli i,j,k,m,res1,res2,q,t,a,b,c,maxi=INT\_MIN,mini=INT\_MAX,sum=0,ans=0,res=0,val=0,ans1=0,ans2=0,rem=0,diff=0,cnt=0,l,r,flag=0,e,index,val1=0,val2=0,z,h=0,u,v,mid,len,tot,fl=0;

string str,str1,str2;

cin>>n>>m;

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

{

cin>>u>>v;

adj[u].push\_back(v);

adj[v].push\_back(u);

}

init();

// for(i=0;i<n;i++)

// {

// if(vis[i]==0)

// {

dfs(1);

// }

// }

sort(vt.begin(),vt.end());

sort(vect.begin(),vect.end());

cout<<vt.size()<<"\n";

for(i=0;i<vt.size();i++)

cout<<vt[i]<<" ";

cout<<"\n";

cout<<vect.size()<<"\n";

for(i=0;i<vect.size();i++)

{

cout<<vect[i].first<<" "<<vect[i].second<<"\n";

}

}

// Biconnected Components

// Two codes one after another , check both in emergency

#include "bits/stdc++.h"

#include<stdio.h>

#include <sstream>

#define pi 3.141592653589793238462

#define limit 100005

#define mo 1000000007;

typedef long long int lli;

typedef unsigned long long int ulli;

const lli mod=9999999999999983;

lli primes[6]={1125899906842597,1495921043,1005985879,1495921043,1005985879,1495921043};

using namespace std;

vector<lli>adj[2000009];

lli parent[2000009];

lli vis[2000009];

lli level[2000009];

lli dist[2000009];

lli dp[100009];

lli arr[100009];

lli brr[2000009];

lli crr[2000009];

lli hashing[2000009];

lli ar[509][509];

lli br[509][509];

lli cr[509][509];

lli multiply(lli a,lli b){return ((a%mod)\*(b%mod))%mod;}

lli add(lli a,lli b){return ((a%mod)+(b%mod))%mod;}

lli sub(lli a,lli b){return ((a%mod)-(b%mod)+mod)%mod;}

#define deb(x) cout<< #x << " " << x <<endl;

lli low[2000009];

lli disc[2000009];

lli tim=0;

stack< pair<lli,lli> >st;

lli odd,even;

lli cnt;

lli power(lli x, lli y,lli p)

{

lli res=1;

x=x%p;

while(y>0)

{

if(y&1)

res=((res%p)\*(x%p))%p;

y=y>>1;

x=((x%p)\*(x%p))%p;

}

return res;

}

void dfs(lli val)

{

lli i;

lli child=0;

vis[val]=1;

low[val]=disc[val]=++tim;

for(i=0;i<adj[val].size();i++)

{

if(vis[adj[val][i]]==0)

{

child++;

st.push({val,adj[val][i]});

parent[adj[val][i]]=val;

dfs(adj[val][i]);

low[val]=min(low[val],low[adj[val][i]]);

set<lli>stt;

if((parent[val]==-1 && child>=2)||(disc[val]<=low[adj[val][i]] && parent[val]!=-1))

{

while(!(st.top().first==val && st.top().second==adj[val][i]))

{

stt.insert(st.top().first);

stt.insert(st.top().second);

st.pop();

}

stt.insert(st.top().first);

stt.insert(st.top().second);

st.pop();

if(stt.size()&1)

odd++;

else

even++;

}

}

else if(disc[adj[val][i]] < low[val] && parent[val]!=adj[val][i])

{

low[val]=min(low[val],disc[adj[val][i]]);

st.push({val,adj[val][i]});

}

}

}

int main()

{

int start\_s = clock();

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL); cout.tie(NULL);

vector< lli>vect,vt,vct;

vector<lli>::iterator itt;

set<lli>::iterator itr;

map<lli,lli>mp,mpp,mp1,mp2;

map<lli,lli>::iterator it;

lli i,j,k,n,m,res1,res2,q,t,a,b,c,maxi=INT\_MIN,mini=INT\_MAX,sum=0,ans=0,res=0,val=0,ans1=0,ans2=0,rem=0,diff=0,cnt=0,l,r,flag=0,e,index,val1=0,val2=0,z,h=0,u,v,mid,len,tot,fl=0;

string str,str1,str2;

cin>>n>>m;

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

{

cin>>u>>v;

adj[u].push\_back(v);

adj[v].push\_back(u);

}

for(i=0;i<100000;i++)

parent[i]=-1;

for(i=0;i<n;i++)

{

if(vis[i]==0)

dfs(i);

if(st.size()>0)

{

set<lli>stt;

while(st.size()>0)

{

stt.insert(st.top().first);

stt.insert(st.top().second);

st.pop();

}

if(stt.size()&1)

odd++;

else

even++;

}

}

cout<<odd<<" "<<even;

}

// OR TRY THIS CODE for biconnected code

#include "bits/stdc++.h"

#include<stdio.h>

#include <sstream>

#define pi 3.141592653589793238462

#define limit 100005

#define mo 1000000007;

typedef long long int lli;

typedef unsigned long long int ulli;

const lli mod=9999999999999983;

lli primes[6]={1125899906842597,1495921043,1005985879,1495921043,1005985879,1495921043};

using namespace std;

vector<lli>adj[2000009];

lli parent[2000009];

lli vis[2000009];

lli level[2000009];

lli dist[2000009];

lli dp[100009];

lli arr[100009];

lli brr[2000009];

lli crr[2000009];

lli hashing[2000009];

lli ar[509][509];

lli br[509][509];

lli cr[509][509];

lli multiply(lli a,lli b){return ((a%mod)\*(b%mod))%mod;}

lli add(lli a,lli b){return ((a%mod)+(b%mod))%mod;}

lli sub(lli a,lli b){return ((a%mod)-(b%mod)+mod)%mod;}

#define deb(x) cout<< #x << " " << x <<endl;

lli low[2000009];

lli disc[2000009];

lli tim=0;

stack< pair<lli,lli> >st;

lli odd,even;

lli cnt;

lli power(lli x, lli y,lli p)

{

lli res=1;

x=x%p;

while(y>0)

{

if(y&1)

res=((res%p)\*(x%p))%p;

y=y>>1;

x=((x%p)\*(x%p))%p;

}

return res;

}

void dfs(lli val)

{

lli i;

lli child=0;

vis[val]=1;

low[val]=disc[val]=++tim;

for(i=0;i<adj[val].size();i++)

{

if(vis[adj[val][i]]==0)

{

child++;

st.push({val,adj[val][i]});

parent[adj[val][i]]=val;

dfs(adj[val][i]);

low[val]=min(low[val],low[adj[val][i]]);

set<lli>stt;

if((disc[val]<=low[adj[val][i]]))

{

while(!(st.top().first==val && st.top().second==adj[val][i]))

{

stt.insert(st.top().first);

stt.insert(st.top().second);

st.pop();

}

stt.insert(st.top().first);

stt.insert(st.top().second);

st.pop();

if(stt.size()&1)

odd++;

else

even++;

}

}

else if(disc[adj[val][i]] < low[val] && parent[val]!=adj[val][i])

{

low[val]=min(low[val],disc[adj[val][i]]);

st.push({val,adj[val][i]});

}

}

}

int main()

{

int start\_s = clock();

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL); cout.tie(NULL);

vector< lli>vect,vt,vct;

vector<lli>::iterator itt;

set<lli>::iterator itr;

map<lli,lli>mp,mpp,mp1,mp2;

map<lli,lli>::iterator it;

lli i,j,k,n,m,res1,res2,q,t,a,b,c,maxi=INT\_MIN,mini=INT\_MAX,sum=0,ans=0,res=0,val=0,ans1=0,ans2=0,rem=0,diff=0,cnt=0,l,r,flag=0,e,index,val1=0,val2=0,z,h=0,u,v,mid,len,tot,fl=0;

string str,str1,str2;

cin>>n>>m;

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

{

cin>>u>>v;

adj[u].push\_back(v);

adj[v].push\_back(u);

}

for(i=0;i<100000;i++)

parent[i]=-1;

for(i=0;i<n;i++)

{

if(vis[i]==0)

dfs(i);

if(st.size()>0)

{

set<lli>stt;

while(st.size()>0)

{

stt.insert(st.top().first);

stt.insert(st.top().second);

st.pop();

}

if(stt.size()&1)

odd++;

else

even++;

}

}

cout<<odd<<" "<<even;

}

// SCC

#include<bits/stdc++.h>

typedef long long int lli;

using namespace std;

typedef unsigned long long int ulli;

const lli mod=1e9+7;

const lli N = 2e5 + 9;

lli primes[6]={1125899906842597,1495921043,1005985879,1495921043,1005985879,1495921043};

#include<stdio.h>

#include <sstream>

#define pi 3.141592653589793238462

lli parent[N];

lli vis[N];

lli level[N];

lli dist[N];

lli dp[N];

lli arr[1000009];

lli brr[1000009];

lli crr[1000009];

lli hashing[N];

lli ar[509][509];

lli br[509][509];

lli cr[509][509];

lli multiply(lli a,lli b){return ((a%mod)\*(b%mod))%mod;}

lli add(lli a,lli b){return ((a%mod)+(b%mod))%mod;}

lli sub(lli a,lli b){return ((a%mod)-(b%mod)+mod)%mod;}

#define deb(x) cout<< #x << " " << x <<endl;

lli dx[] = {1, -1, 0, 0};

lli dy[] = {0, 0, 1, -1};

//deb(n);

vector<lli>adj[N];

vector<lli>GRAPH[N];

stack<lli>STA;

lli finalcnt;

vector<lli>SCC[N];

map< pair<lli,lli> ,lli>mp;

lli cost[N];

lli weight[N];

vector<lli>REV\_SCC[N];

lli max\_weight[N];

void dfs(lli val)

{

lli i;

vis[val]=1;

for(i=0;i<adj[val].size();i++)

{

if(vis[adj[val][i]]==0)

{

dfs(adj[val][i]);

}

}

STA.push(val);

}

void DFS(lli val)

{

lli i;

vis[val]=1;

// cout<<"val="<<val<<"\n";

parent[val]=finalcnt;

weight[finalcnt]+=cost[val];

for(i=0;i<GRAPH[val].size();i++)

{

if(vis[GRAPH[val][i]]==0)

{

DFS(GRAPH[val][i]);

}

}

}

int main()

{

//freopen("input.txt","r",stdin);

//freopen("output.txt","w",stdout);

int start\_s = clock();

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL); cout.tie(NULL);

vector<lli>vect,vect1,vect2,vt,vct;

vector<lli>::iterator itt;

set<lli>st,stt,st1,st2;

set<lli>::iterator itr;

//map<lli,lli>mp,mp1,mp2,mpp;

// map<lli,lli>::iterator it;

lli i,j,k,n,m,q,t,a,d,b,c,l,r,e,idx,ind,index,u,v,x,y,z,h,sz,sz1,sz2,mid,len,tot,prev,temp,curr,p;

lli res=0,res1=0,res2=0,ans=0,ans1=0,ans2=0,val=0,val1=0,val2=0,rem=0,diff=0,cnt=0,flag=0,fl=0,sum=0,maxi=INT\_MIN,mini=INT\_MAX,total=0;

string str,str1,str2;

char ch,ch1,ch2;

cin>>n>>m;

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

{

cin>>cost[i];

}

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

{

cin>>u>>v;

arr[i]=u;

brr[i]=v;

adj[u].push\_back(v);

GRAPH[v].push\_back(u);

}

memset(vis,0,sizeof(vis));

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

{

if(vis[i]==0)

{

dfs(i);

}

}

memset(vis,0,sizeof(vis));

while(STA.size()>0)

{

p=STA.top();

STA.pop();

if(vis[p]==1)

{

continue;

}

//cout<<"P is="<<p<<"\n";

finalcnt++;

DFS(p);

}

/\*

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

{

cout<<"i="<<i<<" "<<"pa="<<parent[i]<<"\n";

}

\*/

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

{

if(parent[arr[i]]==parent[brr[i]])

{

continue;

}

if(mp[{parent[arr[i]],parent[brr[i]]}]==1)

{

continue;

}

SCC[parent[arr[i]]].push\_back(parent[brr[i]]);

mp[{parent[arr[i]],parent[brr[i]]}]=1;

}

/\*

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

{

cout<<"i="<<i<<"\n";

for(j=0;j<SCC[i].size();j++)

{

cout<<SCC[i][j]<<" ";

}

cout<<"\n";

}

\*/

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

{

for(j=0;j<SCC[i].size();j++)

{

REV\_SCC[SCC[i][j]].push\_back(i);

}

}

/\*

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

{

cout<<"i="<<i<<"\n";

for(j=0;j<REV\_SCC[i].size();j++)

{

cout<<REV\_SCC[i][j]<<" ";

}

cout<<"\n";

}

\*/

lli indeg[finalcnt+9]={0};

queue<lli>que;

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

{

for(j=0;j<REV\_SCC[i].size();j++)

{

indeg[REV\_SCC[i][j]]++;

}

}

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

{

if(indeg[i]==0)

{

que.push(i);

// dp[i]=weight[i];

}

}

while(que.size()>0)

{

p=que.front();

que.pop();

max\_weight[p]=weight[p]+dp[p];

for(i=0;i<REV\_SCC[p].size();i++)

{

indeg[REV\_SCC[p][i]]--;

if(indeg[REV\_SCC[p][i]]==0)

{

que.push(REV\_SCC[p][i]);

}

dp[REV\_SCC[p][i]]=max(dp[REV\_SCC[p][i]],max\_weight[p]);

}

}

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

{

cout<<max\_weight[parent[i]]<<" ";

}

//solve();

return 0;

}

//sieve of erastoneses

// minimum prime factor

void pre()

{

lli i,j;

for (i = 2; i \* i <= 1e7; ++i)

{

if (minprime[i] == 0)

{ //If i is prime

for (j = i \* i; j <= 1e7; j += i)

{

if (minprime[j] == 0)

{

minprime[j] = i;

}

}

}

}

for (i = 2; i <= 1e7; ++i)

{

if (minprime[i] == 0)

{

minprime[i] = i;

}

}

}

vector <lli> factorize(lli k)

{

vector <lli> ans;

while(k>1) {

ans.push\_back(minprime[k]);

k/=minprime[k];

}

return ans;

}

//SieveOfEratosthenes

void pre()

{

lli i,j;

prime[1]=prime[0]=1;

for(i=2;i\*i<=1e6;i++)

{

if(prime[i]==0)

{

for(j=i\*i;j<=1e6;j+=i)

{

prime[j]=1;

}

}

}

for(i=1;i<=100;i++)

{

if(prime[i]==0)

cout<<i<<" ";

}

}

//segmented sieve

#include "bits/stdc++.h"

#include<stdio.h>

#include <sstream>

#define pi 3.141592653589793238462

#define limit 100005

#define mo 1000000007;

typedef long long int lli;

typedef unsigned long long int ulli;

const lli mod=9999999999999983;

lli primes[6]={1125899906842597,1495921043,1005985879,1495921043,1005985879,1495921043};

using namespace std;

vector<lli>adj[2000009];

lli parent[2000009];

lli vis[2000009];

lli level[2000009];

lli dist[2000009];

lli dp[100009];

lli arr[100009];

lli brr[2000009];

lli crr[2000009];

lli hashing[2000009];

lli ar[509][509];

lli br[509][509];

lli cr[509][509];

lli multiply(lli a,lli b){return ((a%mod)\*(b%mod))%mod;}

lli add(lli a,lli b){return ((a%mod)+(b%mod))%mod;}

lli sub(lli a,lli b){return ((a%mod)-(b%mod)+mod)%mod;}

#define deb(x) cout<< #x << " " << x <<endl;

lli prime[1000009];

//lli l,r;

vector<lli>vect;

vector<lli>fac[1000009];

void sieve()

{

lli i,j;

for(i=2;i\*i<=1000003;i++)

{

if(prime[i]==0)

{

for(j=2\*i;j<=1000003;j+=i)

{

prime[j]=1;

}

}

}

for(i=2;i<=1000003;i++)

{

if(prime[i]==0)

vect.push\_back(i);

}

}

void segsieve(lli l,lli r)

{

lli i,j,k,base;

for(i=0;vect[i]\*vect[i]<=r;i++)

{

base=(l/vect[i])\*vect[i];

if(base<l)

base+=vect[i];

for(j=base;j<=r;j+=vect[i])

{

fac[j-l].push\_back(vect[i]);

}

}

}

int main()

{

//freopen("input.txt","r",stdin);

//freopen("output.txt","w",stdout);

int start\_s = clock();

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL); cout.tie(NULL);

//vector<lli>vect,vt,vct;

vector<lli>::iterator itt;

set<lli>st;

set<lli>::iterator itr;

map<lli,lli>mp,mpp,mp1,mp2;

map<lli,lli>::iterator it;

lli i,j,k,n,m,p,res1,res2,q,t,a,b,c,maxi=INT\_MIN,mini=INT\_MAX,sum=0,ans=0,res=0,val=0,ans1=0,ans2=0,rem=0,diff=0,cnt=0,flag=0,e,index,val1=0,val2=0,z,h=0,u,v,mid,len,tot,fl=0;

string str,str1,str2;

lli finalans,num,prod1,prod2,expo,div,l,r;

sieve();

cin>>t;

//deb(n);

while(t--)

{

cin>>l>>r;

for(i=0;i<=1000003;i++)

fac[i].clear();

segsieve(l,r);

finalans=0;

for(i=l;i<=r;i++)

{

map<lli,lli>mp;

// cout<<"number is="<<i<<endl;

num=i;

for(j=0;j<fac[i-l].size();j++)

{

//num=i;

div=fac[i-l][j];

// cout<<"factor="<<div<<"num="<<num<<endl;

while(num>0)

{

//cout<<"num="<<num<<"rem="<<num%div<<endl;

if((num%div)==0)

{

num/=div;

mp[div]++;

}

else

{

break;

}

}

}

//cout<<"num="<<num<<endl;

if(num>1)

{

mp[num]++;

}

prod1=1;

prod2=1;

mini=INT\_MAX;

for(it=mp.begin();it!=mp.end();it++)

{

val=it->first;

expo=it->second;

//cout<<"val="<<val<<" "<<"expo="<<expo<<endl;

prod1\*=(expo+1);

mini=min(mini,expo);

}

flag=0;

for(it=mp.begin();it!=mp.end();it++)

{

val=it->first;

expo=it->second;

if(expo==mini&&flag==0){

prod2\*=(expo);

flag=1;}

else

prod2\*=(expo+1);

}

// cout<<"prod1="<<prod1<<" "<<"prod2="<<prod2<<endl;

res=prod1-prod2;

finalans+=res;

}

cout<<finalans<<"\n";

}

return 0;

}

/\*

1

100000000000 100001000000

\*/

//prime factors

void primeFactors(int n)

{

lli i;

while(n%2==0)

{

cout<<2<<" ";

n/=2;

}

for(i=3;i<=sqrt(n);i+=2)

{

while(n%i==0)

{

cout<<i<<" ";

n/=i;

}

}

if(n>2)

cout<<n<<" ";

}

//sum of divisor

void seive()

{

lli i,j;

lli p = 1e7 + 5;

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

{

for(j=i;j<=p;j+=i)

{

divisor[j]+=i;

}

}

}

// modular exponentiation

lli power(lli x, lli y,lli p)

{

lli res=1;

x=x%p;

while(y>0)

{

if(y&1)

res=((res%p)\*(x%p))%p;

y=y>>1;

x=((x%p)\*(x%p))%p;

}

return res;

}

lli POWER(lli x, lli y)

{

lli res=1;

while(y>0)

{

if(y&1)

res\*=x;

y=y>>1;

x\*=x;

}

return res;

}

#define maxn 100001

int arr[maxn],prime[maxn];

ll store\_cnt[maxn],store\_sum[maxn];

vector<int>pf[maxn];

inline ll check\_overflow(ll x,ll y)

{

if(x>inf-y)

{

cout<<"-1";

exit(0);

}

x+=y;

return x;

}

//distinct prime factors of a number

void seive()

{

prime[0]=prime[1]=1;

for(int i=2;i<maxn;i++)

{

if(prime[i])

continue;

pf[i].pb(i);

for(int j=2\*i;j<maxn;j+=i)

{

prime[j]=1;

pf[j].pb(i);

}

}

}

// Mo's

#include "bits/stdc++.h"

#include<stdio.h>

#include <sstream>

#define pi 3.141592653589793238462

#define limit 100005

//#define mo 1000000007;

typedef long long int lli;

typedef unsigned long long int ulli;

const lli mod=9999999999999983;

lli primes[6]={1125899906842597,1495921043,1005985879,1495921043,1005985879,1495921043};

using namespace std;

vector<lli>adj[2000009];

lli parent[2000009];

lli vis[2000009];

lli level[2000009];

lli dist[2000009];

lli dp[100009];

lli arr[30009];

lli brr[2000009];

lli crr[2000009];

lli hashing[2000009];

lli ar[509][509];

lli br[509][509];

lli cr[509][509];

lli multiply(lli a,lli b){return ((a%mod)\*(b%mod))%mod;}

lli add(lli a,lli b){return ((a%mod)+(b%mod))%mod;}

lli sub(lli a,lli b){return ((a%mod)-(b%mod)+mod)%mod;}

#define deb(x) cout<< #x << " " << x <<endl;

//deb(n);

lli blk\_sz;

lli mp[1000009];

lli power(lli x, lli y,lli p)

{

lli res=1;

x=x%p;

while(y>0)

{

if(y&1)

res=((res%p)\*(x%p))%p;

y=y>>1;

x=((x%p)\*(x%p))%p;

}

return res;

}

struct mo

{

lli l;

lli r;

lli index;

};

bool comp(mo a,mo b)

{

if(a.l/blk\_sz!=b.l/blk\_sz)

return a.l/blk\_sz<b.l/blk\_sz;

return a.r<b.r;

}

int main()

{

//freopen("input.txt","r",stdin);

//freopen("output.txt","w",stdout);

int start\_s = clock();

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL); cout.tie(NULL);

vector<lli>vect,vt,vct;

vector<lli>::iterator itt;

set<lli>st;

set<lli>::iterator itr;

//unordered\_map<lli,lli>mp;

map<lli,lli>::iterator it;

lli i,j,k,n,m,p,res1,res2,q,t,a,b,c,maxi=INT\_MIN,mini=INT\_MAX,sum=0,ans=0,res=0,val=0,ans1=0,ans2=0,rem=0,diff=0,cnt=0,l,r,flag=0,e,index,val1=0,val2=0,z,h=0,u,v,mid,len,tot,fl=0;

string str,str1,str2;

lli currL,currR,lef,rig,ind;

cin>>n;

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

cin>>arr[i];

cin>>m;

mo query[m+1];

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

{

cin>>query[i].l>>query[i].r;

query[i].index=i;

}

blk\_sz=sqrt(n);

sort(query+1,query+m+1,comp);

currL=1;

currR=1;

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

{

lef=query[i].l;

rig=query[i].r;

ind=query[i].index;

while(currL<lef)

{

mp[arr[currL]]--;

if(mp[arr[currL]]==0)

ans--;

currL++;

}

while(currL>lef)

{

mp[arr[currL-1]]++;

if(mp[arr[currL-1]]==1)

ans++;

currL--;

}

while(currR<=rig)

{

mp[arr[currR]]++;

if(mp[arr[currR]]==1)

ans++;

currR++;

}

while(currR>rig+1)

{

mp[arr[currR-1]]--;

if(mp[arr[currR-1]]==0)

ans--;

currR--;

}

dist[ind]=ans;

}

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

{

cout<<dist[i]<<"\n";

}

}