|  |
| --- |
|  |
| ACM模板 |
|  |

|  |
| --- |
|  |

目录

[数据结构 4](#_Toc24665168)

[长链剖分 4](#_Toc24665169)

[树链剖分 6](#_Toc24665170)

[整体二分 12](#_Toc24665171)

[整体二分(vector) 15](#_Toc24665172)

[Cdq分治 19](#_Toc24665173)

[带权并查集 22](#_Toc24665174)

[指针版字典树 24](#_Toc24665175)

[点分治 25](#_Toc24665176)

[树状数组 28](#_Toc24665177)

[Splay 30](#_Toc24665178)

[线段树套线性基 35](#_Toc24665179)

[老rmq（nlog-O(1)） 38](#_Toc24665180)

[新rmq ( O(n) ~ O(1) ) (能做500w及以上) 39](#_Toc24665181)

[可持久化并查集 40](#_Toc24665182)

[可持久化字典树 43](#_Toc24665183)

[可持久化数组 45](#_Toc24665184)

[可持久化线段树 46](#_Toc24665185)

[莫队算法 48](#_Toc24665186)

[线段树合并 49](#_Toc24665187)

[假LCT 52](#_Toc24665188)

[吉司机线段树 55](#_Toc24665189)

[大整数运算 59](#_Toc24665190)

[DP 61](#_Toc24665191)

[数位dp 61](#_Toc24665192)

[斜率优化 64](#_Toc24665193)

[数论 65](#_Toc24665194)

[ax=b ( mod c ) 65](#_Toc24665195)

[BSGS 66](#_Toc24665196)

[杜教筛 67](#_Toc24665197)

[扩展gcd 69](#_Toc24665198)

[扩展中国剩余定理 70](#_Toc24665199)

[高斯二项式系数 72](#_Toc24665200)

[格雷码 73](#_Toc24665201)

[预处理逆元 74](#_Toc24665202)

[拉格朗日插值 74](#_Toc24665203)

[整除分块 76](#_Toc24665204)

[矩阵乘法 77](#_Toc24665205)

[Min\_25筛 78](#_Toc24665206)

[线性筛 80](#_Toc24665207)

[Pohlig\_Hellman 82](#_Toc24665208)

[Pollard\_rho 87](#_Toc24665209)

[二次剩余 90](#_Toc24665210)

[博弈 93](#_Toc24665211)

[SG函数 93](#_Toc24665212)

[尼姆博弈 94](#_Toc24665213)

[威佐夫博弈 95](#_Toc24665214)

[图论 96](#_Toc24665215)

[二分图染色 96](#_Toc24665216)

[Dinic 96](#_Toc24665217)

[原始对偶(dijkstra费用流) 99](#_Toc24665218)

[浮点数版Dinic 104](#_Toc24665219)

[Erdos-gallai定理 106](#_Toc24665220)

[输出欧拉路/回路 109](#_Toc24665221)

[最小生成树 110](#_Toc24665222)

[假次小生成树 112](#_Toc24665223)

[Lca 114](#_Toc24665224)

[矩阵树定理 116](#_Toc24665225)

[Tarjan 118](#_Toc24665226)

[最小树形图 120](#_Toc24665227)

[差分约束 122](#_Toc24665228)

[Dfs版spfa 125](#_Toc24665229)

[割点 127](#_Toc24665230)

[边双连通 129](#_Toc24665231)

[多项式相关 132](#_Toc24665232)

[带了假MTT的分治NTT 132](#_Toc24665233)

[FFT 136](#_Toc24665234)

[FWT 139](#_Toc24665235)

[不带任意模的MTT 141](#_Toc24665236)

[多项式开根，求逆，除法，求模等各种操作 144](#_Toc24665237)

[杜教BM 150](#_Toc24665238)

[多项式快速插值(n\*log^2) 154](#_Toc24665239)

[多项式多点求值 160](#_Toc24665240)

[字符串 165](#_Toc24665241)

[AC自动机 165](#_Toc24665242)

[最小表示法 167](#_Toc24665243)

[KMP 169](#_Toc24665244)

[Manacher 170](#_Toc24665245)

[回文自动机 171](#_Toc24665246)

[后缀自动机(广义) 173](#_Toc24665247)

[后缀数组 176](#_Toc24665248)

[计算几何 178](#_Toc24665249)

[球体积交/并 178](#_Toc24665250)

[计算几何板子杂烩 180](#_Toc24665251)

[其他 189](#_Toc24665252)

[double数组的memset 189](#_Toc24665253)

[快速乘 189](#_Toc24665254)

[FAST IO 189](#_Toc24665255)

[O3优化 191](#_Toc24665256)

[FFT常用模数 191](#_Toc24665257)

[模拟退火 193](#_Toc24665258)

[Python 194](#_Toc24665259)

[Exgcd板子 194](#_Toc24665260)

[JAVA 197](#_Toc24665261)

[大整数 197](#_Toc24665262)

[公式及杂项 198](#_Toc24665263)

[二分图 198](#_Toc24665264)

# 数据结构

### 长链剖分

#include <bits/stdc++.h>

#define mp make\_pair

#define sqr(x) (x)\*(x)

using namespace std;

typedef pair<int,int> pii;

typedef long long ll;

const int maxn=1000007;

const int inf=1<<29;

int read(){

int x=0,f=1;

char ch=getchar();

while(ch<'0'||ch>'9') {if(ch=='-') f=-1;ch=getchar();}

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

return x\*f;

}

int md[maxn],d[maxn],son[maxn],e1;

int head[maxn],to[maxn<<1],nex[maxn<<1];

void dfs1(int u,int fa){

md[u]=d[u];son[u]=0;

for(int i=head[u];i;i=nex[i]){

int v=to[i];

if(v==fa) continue;

d[v]=d[u]+1;dfs1(v,u);

md[u]=max(md[u],md[v]);

if(md[v]>md[son[u]]) son[u]=v;

}

}

int n,\*f[maxn],dp[maxn],ans[maxn],\*cur=dp+1;

void dfs2(int u,int fa){

f[u][0]=1;ans[u]=0;

if(!son[u]) return;

f[son[u]]=f[u]+1;dfs2(son[u],u);ans[u]=ans[son[u]]+1;

for(int i=head[u];i;i=nex[i]){

int v=to[i];

if(v==son[u]||v==fa) continue;

f[v]=cur;cur+=md[v]-d[v]+1;dfs2(v,u);

for(int k=0;k<=md[v]-d[v];k++){

f[u][k+1]+=f[v][k];

if(f[u][k+1]>f[u][ans[u]]||f[u][k+1]==f[u][ans[u]]&&k+1<ans[u]){

ans[u]=k+1;

}

}

}

if(f[u][ans[u]]==1) ans[u]=0;

}

void addedge(int u,int v){

++e1;nex[e1]=head[u];head[u]=e1;to[e1]=v;

}

void init(){

for(int i=1;i<=n;i++) head[i]=0;

e1=0;cur=dp+1;

}

int main(){

n=read();init();

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

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

addedge(u,v);addedge(v,u);

}

dfs1(1,0);

f[1]=cur;cur+=md[1]+1;

dfs2(1,0);

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

return 0;

}

### 树链剖分

#include <stdio.h>

#include <algorithm>

#include <string.h>

#pragma comment(linker, "/STACK:302400000,302400000")

#define mp make\_pair

#define sqr(x) (x)\*(x)

using namespace std;

typedef pair<int,int> pii;

typedef long long ll;

const int maxn=100007;

const int inf=1<<29;

int read(){

int x=0,f=1;

char ch=getchar();

while(ch<'0'||ch>'9') {if(ch=='-') f=-1;ch=getchar();}

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

return x\*f;

}

int cl,n,dfn[maxn],f[maxn],d[maxn],sz[maxn],id[maxn];

int e1,head[maxn],to[maxn<<1],nex[maxn<<1],tp[maxn],w[maxn<<1];

int tag[maxn<<2],a[maxn],son[maxn],mx[maxn<<2],mn[maxn<<2];

void pushup(int o){

mx[o]=max(mx[o<<1],mx[o<<1|1]);

mn[o]=min(mn[o<<1],mn[o<<1|1]);

}

void rev(int o){

tag[o]^=1;

mn[o]=-mn[o];mx[o]=-mx[o];

swap(mx[o],mn[o]);

}

void pushdown(int o){

if(tag[o]){

rev(o<<1);

rev(o<<1|1);

tag[o]=0;

}

}

void build(int o,int l,int r){

if(l==r){

mx[o]=mn[o]=a[id[l]];return;

}

tag[o]=0;

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

build(o<<1,l,m);

build(o<<1|1,m+1,r);

pushup(o);

}

void modify(int o,int l,int r,int x,int y){

if(l==r){

mx[o]=mn[o]=y;

return;

}

pushdown(o);

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

if(x<=m) modify(o<<1,l,m,x,y);

else modify(o<<1|1,m+1,r,x,y);

pushup(o);

}

void update(int o,int l,int r,int ql,int qr){

if(l==ql&&r==qr){

rev(o);return;

}

pushdown(o);

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

if(ql<=m&&qr>m) update(o<<1,l,m,ql,m),update(o<<1|1,m+1,r,m+1,qr);

else if(ql<=m) update(o<<1,l,m,ql,qr);

else update(o<<1|1,m+1,r,ql,qr);

pushup(o);

}

int query(int o,int l,int r,int ql,int qr){

if(l==ql&&r==qr){

return mx[o];

}

pushdown(o);

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

if(ql<=m&&qr>m) return max(query(o<<1,l,m,ql,m),query(o<<1|1,m+1,r,m+1,qr));

else if(ql<=m) return query(o<<1,l,m,ql,qr);

else return query(o<<1|1,m+1,r,ql,qr);

}

void addedge(int u,int v,int x){

++e1;nex[e1]=head[u];head[u]=e1;to[e1]=v;w[e1]=x;

}

void dfs1(int u,int fa){

f[u]=fa;

sz[u]=1;son[u]=0;

for(int i=head[u];i;i=nex[i]){

int v=to[i];

if(v==fa) continue;

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

a[v]=w[i];

dfs1(v,u);

sz[u]+=sz[v];

if(sz[v]>sz[son[u]]){

son[u]=v;

}

}

}

void dfs2(int u,int t){

tp[u]=t;dfn[u]=++cl;id[cl]=u;

if(!son[u]) return;

dfs2(son[u],t);

for(int i=head[u];i;i=nex[i]){

int v=to[i];

if(v!=son[u]&&v!=f[u]) dfs2(v,v);

}

}

int Query(int x,int y){

int fx=tp[x],fy=tp[y],ans=-0x3f3f3f3f;

while(fx!=fy){

if(d[fx]>=d[fy]){

ans=max(ans,query(1,1,n,dfn[fx],dfn[x]));

x=f[fx],fx=tp[x];

}

else{

ans=max(ans,query(1,1,n,dfn[fy],dfn[y]));

y=f[fy],fy=tp[y];

}

}

if(dfn[x]<dfn[y]) ans=max(ans,query(1,1,n,dfn[x]+1,dfn[y]));

else if(dfn[x]>dfn[y]) ans=max(ans,query(1,1,n,dfn[y]+1,dfn[x]));

return ans;

}

void Update(int x,int y){

int fx=tp[x],fy=tp[y];

while(fx!=fy){

if(d[fx]>=d[fy]){

update(1,1,n,dfn[fx],dfn[x]);

x=f[fx],fx=tp[x];

}

else{

update(1,1,n,dfn[fy],dfn[y]);

y=f[fy],fy=tp[y];

}

}

if(dfn[x]<dfn[y]) update(1,1,n,dfn[x]+1,dfn[y]);

else if(dfn[x]>dfn[y]) update(1,1,n,dfn[y]+1,dfn[x]);

}

int U[maxn],V[maxn];

void init(){

for(int i=1;i<=n;i++) head[i]=0;

e1=cl=0;

}

int main(){

int t=read();

while(t--){

int q;n=read();init();

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

U[i]=read(),V[i]=read();int w=read();

addedge(U[i],V[i],w);addedge(V[i],U[i],w);

}

dfs1(1,0);dfs2(1,1);build(1,1,n);

while(1){

char s[11];scanf("%s",s+1);

if(s[1]=='Q'){

int x=read(),y=read();

printf("%d\n",Query(x,y));

}

else if(s[1]=='C'){

int x=read(),y=read();

modify(1,1,n,dfn[d[U[x]]<d[V[x]]?V[x]:U[x]],y);

}

else if(s[1]=='N'){

int x=read(),y=read();

Update(x,y);

}

else break;

}

}

return 0;

}

### 整体二分

#include <stdio.h>

#include <algorithm>

#include <string.h>

#include <math.h>

#include <iostream>

#include <vector>

#define mp make\_pair

#define pii pair<int,int>

#define lowbit(x) (x)&(-x)

using namespace std;

typedef long long ll;

#define rint register int

const int maxn=100007;

const int inf=(1LL<<29);

int read(){

int x=0;int f=1;

char c=getchar();

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

if(c=='-') f=-1;c=getchar();

}

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

x\*=f;return x;

}

int ans[maxn];

struct Query{

int l,r,x,pos;

Query(int l=0,int r=0,int x=0,int pos=0):l(l),r(r),x(x),pos(pos){}

}q[maxn],qL[maxn],qR[maxn];

struct node{

int x,pos;

node(int x=0,int pos=0):x(x),pos(pos){}

}num[maxn],numL[maxn],numR[maxn];

int n,v[maxn];

void add(int o,int x){

for(int i=o;i<=n;i+=lowbit(i)) v[i]+=x;

}

int query(int o){

int x=0;

for(int i=o;i;i-=lowbit(i)) x+=v[i];

return x;

}

inline void solve(int l,int r,int ql,int qr,int nl,int nr){

// cout<<q.size()<<endl;

if(l==r){

for(int i=ql;i<=qr;i++){

ans[q[i].pos]=l;

}

return;

}

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

int qst=ql,qed=qr,nst=nl,ned=nr;

for(int i=nl;i<=nr;i++){

if(num[i].x<=m) numL[nst++]=num[i],add(num[i].pos,1);

else numR[ned--]=num[i];

}

for(int i=ql;i<=qr;i++){

int x=query(q[i].r)-query(q[i].l-1);

if(q[i].x<=x){

qL[qst++]=q[i];

}

else q[i].x-=x,qR[qed--]=q[i];

}

for(int i=nl;i<nst;i++){

add(numL[i].pos,-1);

}

for(int i=ql;i<qst;i++) q[i]=qL[i];

for(int i=qst;i<=qr;i++) q[i]=qR[i];

for(int i=nl;i<nst;i++) num[i]=numL[i];

for(int i=nst;i<=nr;i++) num[i]=numR[i];

if(qst!=ql) solve(l,m,ql,qst-1,nl,nst-1);

if(qed!=qr) solve(m+1,r,qst,qr,nst,nr);

}

int main(){

// freopen("test.in","r",stdin);

// freopen("my.out","w",stdout);

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

for(int i=1;i<=n;i++) num[i]=node(read(),i);

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

int l=read(),r=read(),x=read();

q[i]=Query(l,r,x,i);

}

solve(-1e9,1e9,1,m,1,n);

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

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

}

return 0;

}

### 整体二分(vector)

#include <stdio.h>

#include <algorithm>

#include <string.h>

#include <math.h>

#include <iostream>

#include <vector>

#define mp make\_pair

#define pii pair<int,int>

#define lowbit(x) (x)&(-x)

using namespace std;

typedef long long ll;

#define rint register int

const int maxn=100007;

const int inf=(1LL<<29);

ll read(){

ll x=0;int f=1;

char c=getchar();

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

if(c=='-') f=-1;c=getchar();

}

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

x\*=f;return x;

}

int ans[maxn];

struct qu{

int type,l,r,pos;ll x;

qu(int type=0,int l=0,int r=0,ll x=0,int pos=0):type(type),l(l),r(r),x(x),pos(pos){}

};

ll tag[maxn<<2],sum[maxn<<2];

void pushup(int o){

sum[o]=sum[o<<1]+sum[o<<1|1];

}

void add(int o,int l,int r,int x){

tag[o]+=x;sum[o]+=(r-l+1)\*x;

}

void pushdown(int o,int l,int r){

if(tag[o]){

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

add(o<<1,l,m,tag[o]);

add(o<<1|1,m+1,r,tag[o]);

tag[o]=0;

}

}

void update(int o,int l,int r,int ql,int qr,int x){

if(l==ql&&r==qr){

add(o,l,r,x);return;

}

pushdown(o,l,r);

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

if(ql<=m&&qr>m) update(o<<1,l,m,ql,m,x),update(o<<1|1,m+1,r,m+1,qr,x);

else if(ql<=m) update(o<<1,l,m,ql,qr,x);

else update(o<<1|1,m+1,r,ql,qr,x);

pushup(o);

}

ll qu(int o,int l,int r,int ql,int qr){

if(l==ql&&r==qr){

return sum[o];

}

pushdown(o,l,r);

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

if(ql<=m&&qr>m) return qu(o<<1,l,m,ql,m)+qu(o<<1|1,m+1,r,m+1,qr);

else if(ql<=m) return qu(o<<1,l,m,ql,qr);

else return qu(o<<1|1,m+1,r,ql,qr);

}

int n;

inline void solve(int l,int r,vector<qu> q){

if(l==r){

for(int i=0;i<q.size();i++){

if(q[i].type==2)ans[q[i].pos]=l;

}

return;

}

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

vector<qu> ql,qr;

for(int i=0;i<q.size();i++){

if(q[i].type==1){

if(q[i].x>m){

qr.push\_back(q[i]);update(1,1,n,q[i].l,q[i].r,1);

}

else ql.push\_back(q[i]);

}

else{

ll x=qu(1,1,n,q[i].l,q[i].r);

// cout<<x<<endl;

if(q[i].x>x) q[i].x-=x,ql.push\_back(q[i]);

else qr.push\_back(q[i]);

}

}

for(int i=0;i<qr.size();i++){

if(qr[i].type==1) update(1,1,n,qr[i].l,qr[i].r,-1);

}

if(!ql.empty()) solve(l,m,ql);

if(!qr.empty()) solve(m+1,r,qr);

}

int main(){

n=read();int m=read();vector<qu> q;int top=0;

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

int opt=read(),l=read(),r=read();ll x=read();

if(opt-1) top++;

q.push\_back(qu(opt,l,r,x,top));

}

solve(-50000,50000,q);

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

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

}

return 0;

}

### Cdq分治

#include <stdio.h>

#include <algorithm>

#include <string.h>

#include <math.h>

#include <iostream>

#include <vector>

#define mp make\_pair

#define pii pair<int,int>

#define lowbit(x) (x)&(-x)

using namespace std;

typedef long long ll;

#define rint register int

const int maxn=200007;

const int inf=(1LL<<29);

ll read(){

ll x=0;int f=1;

char c=getchar();

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

if(c=='-') f=-1;c=getchar();

}

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

x\*=f;return x;

}

struct nd{

int x,y,z,v;

nd(int x=0,int y=0,int z=0,int v=0):x(x),y(y),z(z),v(v){}

bool operator < (nd a) const{

if(x!=a.x) return x<a.x;

else if(y!=a.y) return y>a.y;

else return z<a.z;

}

}p[maxn],pr[maxn];

int a[maxn],v[maxn];

ll ans[maxn];

int n,vis[maxn];

void add(int o,int x){

for(int i=o;i<=n;i+=lowbit(i)) v[i]+=x;

}

int query(int o){

int x=0;

for(int i=o;i;i-=lowbit(i)) x+=v[i];

return x;

}

//外部排序时如表达式(第一位第二位第三位)出现相等的情况，需要调整cmp，如果全部相等，则需缩点

//内部排序时第二维第三维可随意处理，第一维如出现不等号（但是数据中有第一维相等的情况），需要加上判断条件(只加入不等于的点)

//注意内存

void cdq(int l,int r){

if(l==r) return;

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

cdq(l,m);cdq(m+1,r);

int top=m;

for(int i=r;i>m;--i){

while(top>=l&&p[top].y>p[i].y) add(p[top].z,p[top].v),top--;

ans[p[i].x]+=query(p[i].z-1);

}

for(int i=top+1;i<=m;i++) add(p[i].z,-p[i].v);

top=l;

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

while(top<=m&&p[top].y<p[i].y) add(p[top].z,p[top].v),top++;

if(p[i].x) ans[p[i].x]+=query(n)-query(p[i].z);

}

for(int i=l;i<top;i++) add(p[i].z,-p[i].v);

top=l;int now=l;

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

while(top<=m&&p[top].y<=p[i].y) pr[now++]=p[top++];

pr[now++]=p[i];

}

while(top<=m) pr[now++]=p[top++];

for(int i=l;i<=r;i++) p[i]=pr[i];

}

int pos[maxn];

int main(){

//freopen("test.in","r",stdin);

n=read();int m=read();int top=0;

for(int i=1;i<=n;i++) a[i]=read(),p[++top]=nd(0,a[i],i,1),pos[a[i]]=i;

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

int x=read();x=pos[x];

p[++top]=nd(i,a[x],x,-1);

}

sort(p+1,p+top+1);

cdq(1,top);

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

return 0;

}

### 带权并查集

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

using namespace std;

const int maxn=10007;

struct query{

int l,r,id;

}p[maxn];

int len,a[maxn],fa[maxn],v[maxn];

int find(int x){

if(x==fa[x]) return x;

else{

int t=fa[x];

fa[x]=find(fa[x]);

if(t!=fa[t]) v[x]^=v[t];

}

return fa[x];

}

int main(){

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

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

char s[39];scanf("%d%d%s",&p[i].l,&p[i].r,s+1);

a[++len]=p[i].l-1;a[++len]=p[i].r;

if(s[1]=='o') p[i].id=1;

else p[i].id=0;

}

sort(a+1,a+len+1);

len=unique(a+1,a+len+1)-a-1;

for(int i=1;i<=len;i++) fa[i]=i;

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

p[i].l=lower\_bound(a+1,a+len+1,p[i].l-1)-a;

p[i].r=lower\_bound(a+1,a+len+1,p[i].r)-a;

if(find(p[i].l)==find(p[i].r)){

if((v[p[i].l]^v[p[i].r])!=p[i].id){

printf("%d",i-1);return 0;

}

}

else if(find(p[i].l)<find(p[i].r)){

v[find(p[i].r)]=(v[p[i].l]^p[i].id^v[p[i].r]);

fa[find(p[i].r)]=find(p[i].l);

}

else{

v[find(p[i].l)]=(v[p[i].l]^v[p[i].r]^p[i].id);

fa[find(p[i].l)]=find(p[i].r);

}

}

printf("%d",m);

return 0;

}

### 指针版字典树

#include <bits/stdc++.h>

using namespace std;

struct node{

int cnt;

node\* tr[26];

};

node\* rt;

void init(){

rt=new node();

}

void insert(){

node\* u=rt;

for(int i=0;s[i];i++){

int x=s[i]-'a';

if(u->tr[x]==NULL) u->tr[x]=new node();

u=u->tr[x];

}

}

int query(){

node\* u=rt;

for(int i=0;s[i];i++){

int go=s[i]-'a';

if(u->tr[x]==NULL) return -1;

u=u->tr[x];

}

return u->cnt;

}

### 点分治

#include <bits/stdc++.h>

#define make\_pair mp

#define pii pair<int,int>

using namespace std;

typedef long long ll;

const int maxn=100007;

const int inf=(1LL<<29);

int ans,e1,rt,totn,val[4],dep[maxn],w[maxn<<1],f[maxn],head[maxn],to[maxn<<1],nex[maxn<<1],sz[maxn];

bool vis[maxn];

int gcd(int a,int b){

return b==0?a:gcd(b,a%b);

}

void getrt(int u,int fa){

sz[u]=1;f[u]=0;

for(int i=head[u];i;i=nex[i]){

int v=to[i];

if(v!=fa&&!vis[v]){

getrt(v,u);sz[u]+=sz[v];f[u]=max(f[u],sz[v]);

}

}

f[u]=max(f[u],totn-sz[u]);

if(f[u]<f[rt]) rt=u;

}

void addedge(int u,int v,int x){

++e1;nex[e1]=head[u];head[u]=e1;to[e1]=v;w[e1]=x;

}

void getdep(int u,int fa){

val[dep[u]]++;

for(int i=head[u];i;i=nex[i]){

int v=to[i];

if(!vis[v]&&v!=fa){

dep[v]=(dep[u]+w[i])%3;

getdep(v,u);

}

}

}

int calc(int u,int v){

for(int i=0;i<3;i++) val[i]=0;

dep[u]=v;getdep(u,0);

return val[1]\*val[2]\*2+val[0]\*val[0];

}

void solve(int u){

ans+=calc(u,0);

vis[u]=1;

for(int i=head[u];i;i=nex[i]){

int v=to[i];

if(!vis[v]){

ans-=calc(v,w[i]);

rt=0;totn=sz[v];

getrt(v,0);

solve(rt);

}

}

}

int main(){

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

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

int u,v,x;scanf("%d%d%d",&u,&v,&x);

x%=3;

addedge(u,v,x);addedge(v,u,x);

}

f[0]=totn=n;

getrt(1,0);

solve(rt);

int t=gcd(ans,n\*n);

printf("%d/%d",ans/t,n\*n/t);

return 0;

}

### 树状数组

#include <bits/stdc++.h>

#define make\_pair mp

#define pii pair<int,int>

using namespace std;

typedef long long ll;

const int maxn=100007;

const int inf=(1LL<<29);

int ans,e1,rt,totn,val[4],dep[maxn],w[maxn<<1],f[maxn],head[maxn],to[maxn<<1],nex[maxn<<1],sz[maxn];

bool vis[maxn];

int gcd(int a,int b){

return b==0?a:gcd(b,a%b);

}

void getrt(int u,int fa){

sz[u]=1;f[u]=0;

for(int i=head[u];i;i=nex[i]){

int v=to[i];

if(v!=fa&&!vis[v]){

getrt(v,u);sz[u]+=sz[v];f[u]=max(f[u],sz[v]);

}

}

f[u]=max(f[u],totn-sz[u]);

if(f[u]<f[rt]) rt=u;

}

void addedge(int u,int v,int x){

++e1;nex[e1]=head[u];head[u]=e1;to[e1]=v;w[e1]=x;

}

void getdep(int u,int fa){

val[dep[u]]++;

for(int i=head[u];i;i=nex[i]){

int v=to[i];

if(!vis[v]&&v!=fa){

dep[v]=(dep[u]+w[i])%3;

getdep(v,u);

}

}

}

int calc(int u,int v){

for(int i=0;i<3;i++) val[i]=0;

dep[u]=v;getdep(u,0);

return val[1]\*val[2]\*2+val[0]\*val[0];

}

void solve(int u){

ans+=calc(u,0);

vis[u]=1;

for(int i=head[u];i;i=nex[i]){

int v=to[i];

if(!vis[v]){

ans-=calc(v,w[i]);

rt=0;totn=sz[v];

getrt(v,0);

solve(rt);

}

}

}

int main(){

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

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

int u,v,x;scanf("%d%d%d",&u,&v,&x);

x%=3;

addedge(u,v,x);addedge(v,u,x);

}

f[0]=totn=n;

getrt(1,0);

solve(rt);

int t=gcd(ans,n\*n);

printf("%d/%d",ans/t,n\*n/t);

return 0;

}

### Splay

#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

const int maxn=100007;

int val[maxn],sz[maxn],ans,e1,num[maxn],rt,fa[maxn],tr[maxn][2],a[maxn],rev[maxn];

void pushup(int x){

if(!x) return;

sz[x]=sz[tr[x][0]]+sz[tr[x][1]]+1;

}

void re(int o){

if(!o) return;

rev[o]^=1;swap(tr[o][0],tr[o][1]);

}

void pushdown(int x){

if(!x) return;

if(rev[x]){

re(tr[x][0]);re(tr[x][1]);

rev[x]=0;

}

}

void rotate(int x,int &o){

int y=fa[x],z=fa[y],l=(tr[y][1]==x),r=l^1;

pushdown(y);pushdown(x);

if(y==o) o=x;

else tr[z][tr[z][1]==y]=x;

fa[x]=z;fa[y]=x;fa[tr[x][r]]=y;

tr[y][l]=tr[x][r];tr[x][r]=y;

pushup(y);pushup(x);

}

void splay(int x,int &o){

int y,z;

while(x!=o){

y=fa[x],z=fa[y];

if(y!=o){

if(tr[y][0]==x^tr[z][0]==y) rotate(x,o);

else rotate(y,o);

}

rotate(x,o);

}

}

int find(int o,int x){

pushdown(o);while(o&&num[o]!=x) o=tr[o][num[o]<x],pushdown(o);

return o;

}

int query\_min(int o){

pushdown(o);while(tr[o][0]) o=tr[o][0],pushdown(o);

return o;

}

int query\_max(int o){

pushdown(o);while(tr[o][1]) o=tr[o][1],pushdown(o);

return o;

}

void del(int x){

if(!x) return;

splay(x,rt);pushdown(x);

if(1LL\*tr[x][0]\*tr[x][1]==0) rt=tr[x][0]+tr[x][1];

else{

int y=query\_min(tr[x][1]);

splay(y,rt);

tr[y][0]=tr[x][0];

fa[tr[x][0]]=y;

}

fa[rt]=0;pushup(rt);

}

void ins(int &o,int p,int x){

if(!o){

o=++e1;num[o]=x;fa[o]=p;splay(o,rt);return;

}

if(x<num[o]) ins(tr[o][0],o,x);else ins(tr[o][1],o,x);

pushup(o);

}

int query\_pre(int x){

x=tr[x][0];pushdown(x);

while(tr[x][1]) x=tr[x][1],pushdown(x);

return x;

}

int query\_sub(int x){

x=tr[x][1];pushdown(x);

while(tr[x][0]) x=tr[x][0],pushdown(x);

return x;

}

void build(int &o,int l,int r,int p){

//printf("%d %d\n",o,p);

if(l>r) return;

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

o=m;fa[o]=p;sz[o]=1;tr[o][0]=tr[o][1]=0;

build(tr[o][0],l,m-1,o);build(tr[o][1],m+1,r,o);

pushup(o);

}

int query\_kth(int o,int x){

if(!o) return -1;

pushdown(o);

if(sz[tr[o][0]]+1==x) return o;

else if(sz[tr[o][0]]>=x) return query\_kth(tr[o][0],x);

else return query\_kth(tr[o][1],x-sz[tr[o][0]]-1);

}

int split(int a,int b){

a=query\_kth(rt,a);b=query\_kth(rt,b+2);

splay(a,rt);splay(b,tr[rt][1]);

int x=tr[b][0];tr[b][0]=0;fa[x]=0;

pushup(b);pushup(a);

return x;

}

void rever(int a,int b){

a=query\_kth(rt,a);b=query\_kth(rt,b+2);

splay(a,rt);splay(b,tr[rt][1]);

int x=tr[b][0];pushdown(x);

re(x);pushup(b);pushup(a);

}

void dfs(int o){

if(!o) return;

pushdown(o);

dfs(tr[o][0]);printf("%d ",num[o]);dfs(tr[o][1]);

}

int main(){

return 0;

}

### 线段树套线性基

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

using namespace std;

const int maxn=500007;

int id(int l,int r){

return (l+r)|(l!=r);

}

int s[maxn],z[20];

struct xxj{

int a[20];

void insert(int x){

for(int i=19;~i;--i){

if(x&(1<<i)){

if(a[i]) x^=a[i];

else{

a[i]=x;

break;

}

}

}

}

int query(){

int ans=0;

for(int i=19;~i;i--){

if(!(ans&z[i])) ans^=a[i];

}

return ans;

}

}a[maxn<<1];

xxj merge(xxj x,xxj y){

for(int i=19;i>=0;i--) if(y.a[i]) x.insert(y.a[i]);

return x;

}

xxj Merge(xxj A,xxj B) {

xxj All,C,D;

for (int i=19;~i;--i) {

All.a[i]=A.a[i];

D.a[i]=1<<i;

}

for(int i=19;~i;--i){

if(B.a[i]){

int v=B.a[i],k=0,can=1;

for(int j=19;~j; --j){

if(v>>j&1){

if(All.a[j]){

v^=All.a[j];

k^=D.a[j];

}

else{

can=0;

All.a[j]=v;

D.a[j]=k;

break;

}

}

}

if(can){

int v=0;

for(int j=19;~j;--j) {

if(k>>j&1)) {

v^=A.a[j];

}

}

C.insert(v);

}

}

}

return C;

}

xxj build(int l,int r){

if(l==r){

a[id(l,r)].insert(s[l]);

return a[id(l,r)];

}

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

a[id(l,r)]=merge(build(l,m),build(m+1,r));

return a[id(l,r)];

}

xxj query(int l,int r,int ql,int qr){

if(ql==l&&r==qr){

return a[id(l,r)];

}

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

if(ql<=m&&qr>m) return merge(query(l,m,ql,m),query(m+1,r,m+1,qr));

else if(ql<=m) return query(l,m,ql,qr);

else return query(m+1,r,ql,qr);

}

int main(){

for(int i=0;i<=19;i++) z[i]=(1<<i);

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

for(int i=1;i<=n;++i) scanf("%d",&s[i]);

build(1,n);

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

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

int l,r;scanf("%d%d",&l,&r);

printf("%d\n",query(1,n,l,r).query());

}

return 0;

}

### 老rmq（nlog-O(1)）

#include <stdio.h>

#include <algorithm>

#include <iostream>

using namespace std;

const int maxn=200007;

int n,st[maxn][32],val[maxn],lg[maxn];

inline void init(){

for(int i=1;i<=n;i++) st[i][0]=val[i];

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

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

st[k][i]=max(st[k][i-1],st[k+(1<<i-1)][i-1]);

}

}

for(int i=1;i<=n;i++) lg[i]=lg[i-1]+((1<<(lg[i-1]+1))==i);

}

inline int query(int l,int r){

int x=lg[r-l+1];

return max(st[l][x],st[r-(1<<x)+1][x]);

}

int main(){

return 0;

}

### 新rmq ( O(n) ~ O(1) ) (能做500w及以上)

#include <stdio.h>

#include <algorithm>

using namespace std;

const int maxn=1000007;

int root[100007],e1,a[100007],sum[2000007],ls[2000007],rs[2000007];

void pushup(int o){

sum[o]=sum[ls[o]]+sum[rs[o]];

}

void ins(int &o,int p,int l,int r,int x){

o=++e1;ls[o]=ls[p];rs[o]=rs[p];sum[o]=sum[p];

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

if(l==r){

sum[o]+=1;return;

}

x<=m?ins(ls[o],ls[p],l,m,x):ins(rs[o],rs[p],m+1,r,x);

pushup(o);

}

void change(int &o,int l,int r,int x,int y){

if(!o) o=++e1;

if(l==r){

sum[o]+=y;

}

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

if(x<=m) change(ls[o],l,m,x,y);

else change(rs[o],m+1,r,x,y);

pushup(o);

}

int query(int x,int y,int l,int r,int z){

if(l==r) return a[l];

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

if(sum[ls[y]]-sum[ls[x]]>=z) return query(ls[x],ls[y],l,m,z);

else return query(rs[x],rs[y],m+1,r,z-(sum[ls[y]]-sum[ls[x]]));

}

int main(){

return 0;

}

### 可持久化并查集

#include <stdio.h>

#include <iostream>

#include <algorithm>

using namespace std;

const int maxn=1000007;

int root[1000007],e1,n,a[1000007],ls[20000007],rs[20000007],val[20000007],fa[20000007],sz[20000007];

void insfa(int &o,int p,int l,int r,int x,int y){

o=++e1;ls[o]=ls[p];rs[o]=rs[p];fa[o]=fa[p];sz[o]=sz[p];

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

if(l==r){

fa[o]=y;return;

}

x<=m?insfa(ls[o],ls[p],l,m,x,y):insfa(rs[o],rs[p],m+1,r,x,y);

}

void inssz(int o,int l,int r,int x,int y){

if(l==r){

sz[o]=y;return;

}

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

x<=m?inssz(ls[o],l,m,x,y):inssz(rs[o],m+1,r,x,y);

}

void build(int &o,int l,int r){

if(!o) o=++e1;

if(l==r){

sz[o]=1;fa[o]=l;return;

}

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

build(ls[o],l,m);build(rs[o],m+1,r);

}

int getsz(int o,int l,int r,int x){

if(l==r) return sz[o];

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

if(x<=m) return getsz(ls[o],l,m,x);

else return getsz(rs[o],m+1,r,x);

}

int getfa(int o,int l,int r,int x){

if(l==r) return fa[o];

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

if(x<=m) return getfa(ls[o],l,m,x);

else return getfa(rs[o],m+1,r,x);

}

int find(int o,int x){

int f=getfa(root[o],1,n,x);

return f==x?x:find(o,f);

}

void merge(int o,int p,int u,int v){

int x=find(p,u),y=find(p,v);

int sx=getsz(root[p],1,n,x),sy=getsz(root[p],1,n,y);

if(sx>=sy){

insfa(root[o],root[p],1,n,y,x);

inssz(root[o],1,n,x,sx+sy);

}

else{

insfa(root[o],root[p],1,n,x,y);

inssz(root[o],1,n,y,sx+sy);

}

}

int main(){

int m;cin>>n>>m;

build(root[0],1,n);

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

int opt,x;scanf("%d%d",&opt,&x);

if(opt==1){

int y;scanf("%d",&y);

merge(i,i-1,x,y);

}

else if(opt==2){

root[i]=root[x];

}

else{

root[i]=root[i-1];

int y;scanf("%d",&y);

printf("%d\n",find(i,x)==find(i,y));

}

}

return 0;

}

### 可持久化字典树

#include <stdio.h>

#include <math.h>

#include <algorithm>

using namespace std;

const int maxn=10000007;

typedef long long ll;

int root[100007],e1,tr[maxn][2],sum[10000007];

void pushup(int o){

sum[o]=sum[tr[o][0]]+sum[tr[o][1]];

}

void ins(int &o,int p,int len,ll x){

o=++e1;

for(int i=0;i<2;i++) tr[o][i]=tr[p][i];

sum[o]=sum[p];

if(len==-1){

sum[o]+=1;return;

}

ins(tr[o][x>>len&1],tr[p][x>>len&1],len-1,x);

pushup(o);

}

ll query(int x,int y,int len,ll v){

if(len==-1) return 0;

int p=(v>>len&1)^1;

if(sum[tr[y][p]]-sum[tr[x][p]]) return (1LL<<len)+query(tr[x][p],tr[y][p],len-1,v);

else return query(tr[x][p^1],tr[y][p^1],len-1,v);

}

int main(){

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

for(int i=1;i<=n;i++) scanf("%lld",&a[i]),sum1[i+1]=sum1[i]^a[i];

n++;

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

ins(root[i],root[i-1],60,sum1[i]);

}

return 0;

}

### 可持久化数组

#include <stdio.h>

#include <iostream>

#include <algorithm>

using namespace std;

const int maxn=1000007;

int root[1000007],e1,a[1000007],ls[20000007],rs[20000007],val[20000007];

void ins(int &o,int p,int l,int r,int x,int y){

o=++e1;ls[o]=ls[p];rs[o]=rs[p];

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

if(l==r){

val[o]=y;return;

}

x<=m?ins(ls[o],ls[p],l,m,x,y):ins(rs[o],rs[p],m+1,r,x,y);

}

void build(int &o,int l,int r){

if(!o) o=++e1;

if(l==r){

val[o]=a[l];return;

}

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

build(ls[o],l,m);build(rs[o],m+1,r);

}

int query(int o,int l,int r,int x){

if(l==r) return val[o];

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

if(x<=m) return query(ls[o],l,m,x);

else return query(rs[o],m+1,r,x);

}

int main(){

int n,m;cin>>n>>m;

for(int i=1;i<=n;i++) scanf("%d",&a[i]);

build(root[0],1,n);

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

int p,opt,x;scanf("%d%d%d",&p,&opt,&x);

if(opt==1){

int y;scanf("%d",&y);

ins(root[i],root[p],1,n,x,y);

}

else{

printf("%d\n",query(root[p],1,n,x));root[i]=root[p];

}

}

return 0;

}

### 可持久化线段树

#include <stdio.h>

#include <algorithm>

using namespace std;

const int maxn=1000007;

int root[100007],e1,a[100007],sum[2000007],ls[2000007],rs[2000007];

void pushup(int o){

sum[o]=sum[ls[o]]+sum[rs[o]];

}

void ins(int &o,int p,int l,int r,int x){

o=++e1;ls[o]=ls[p];rs[o]=rs[p];sum[o]=sum[p];

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

if(l==r){

sum[o]+=1;return;

}

x<=m?ins(ls[o],ls[p],l,m,x):ins(rs[o],rs[p],m+1,r,x);

pushup(o);

}

void change(int &o,int l,int r,int x,int y){

if(!o) o=++e1;

if(l==r){

sum[o]+=y;

}

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

if(x<=m) change(ls[o],l,m,x,y);

else change(rs[o],m+1,r,x,y);

pushup(o);

}

int query(int x,int y,int l,int r,int z){

if(l==r) return a[l];

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

if(sum[ls[y]]-sum[ls[x]]>=z) return query(ls[x],ls[y],l,m,z);

else return query(rs[x],rs[y],m+1,r,z-(sum[ls[y]]-sum[ls[x]]));

}

int main(){

return 0;

}

### 莫队算法

#include <bits/stdc++.h>

#define make\_pair mp

#define pii pair<int,int>

using namespace std;

typedef long long ll;

const int maxn=200007;

const int inf=(1LL<<29);

int be[maxn],bl,blc;

ll tot[maxn],val[maxn],ans;

int a[1000007];int x;

struct point{

int l,r,id;

bool operator < (point a) const{

return be[l]!=be[a.l]?be[l]<be[a.l]:r<a.r;

}

}q[maxn];

int l,r;

void modify(int xx,int y){

if(y<0){

a[val[xx]]--;

if(!a[val[xx]]) ans--;

return;

}

//printf("%d %d %lld %d\n",xx,y,ans,y\*a[sum[xx]^x]);

a[val[xx]]++;

if(a[val[xx]]==1) ans++;

}

int main(){

int n,m;cin>>n;bl=sqrt(n);

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

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

be[i]=(i-1)/bl+1;

}

cin>>m;

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

scanf("%d%d",&q[i].l,&q[i].r);q[i].id=i;

}

sort(q+1,q+m+1);

l=1,r=0;

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

while(r>q[i].r) modify(r,-1),r--;

while(r<q[i].r) modify(r+1,1),r++;

while(l<q[i].l) modify(l,-1),l++;

while(l>q[i].l) modify(l-1,1),l--;

tot[q[i].id]=ans;

}

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

printf("%lld\n",tot[i]);

}

return 0;

}

### 线段树合并

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

#include <math.h>

#include <vector>

#include <map>

using namespace std;

typedef long long ll;

const int maxn=100007;

int e1,ls[maxn\*200],rs[maxn\*200],sum[maxn\*200],root[maxn];ll tot[maxn\*200];

int merge(int x,int y){

if(!x||!y) return x+y;

int now=++e1;

sum[now]=sum[x]+sum[y];

tot[now]=tot[x]+tot[y];

ls[now]=merge(ls[x],ls[y]);

rs[now]=merge(rs[x],rs[y]);

return now;

}

void pushup(int o){

sum[o]=sum[ls[o]]+sum[rs[o]];

tot[o]=tot[ls[o]]+tot[rs[o]];

}

void insert(int &o,int l,int r,int x,int y){

if(!o) o=++e1;

if(l==r){

sum[o]+=y;tot[o]+=l\*y;return;

}

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

if(x<=mid) insert(ls[o],l,mid,x,y);

else insert(rs[o],mid+1,r,x,y);

pushup(o);

}

ll query(int x,int l,int r,int z){

if(l==r) return min(z,sum[x])\*l;

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

if(sum[ls[x]]>=z) return query(ls[x],l,mid,z);

else return tot[ls[x]]+query(rs[x],mid+1,r,z-sum[ls[x]]);

}

int main(){

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

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

int a,b,c;scanf("%d%d%d",&a,&b,&c);

insert(root[a],1,1e7,c,1);

insert(root[b+1],1,1e7,c,-1);

}

for(int i=1;i<=100000;i++) root[i]=merge(root[i],root[i-1]);

ll last=1;

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

int x,a,b,c;scanf("%d%d%d%d",&x,&a,&b,&c);

int y=1+(1LL\*a\*last+b)%c;

printf("%lld\n",last=query(root[x],1,1e7,y));

}

return 0;

}

### 假LCT

#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

const int maxn=100007;

int sz[maxn],ans,e1,num[maxn],rt,fa[maxn],tr[maxn][2],a[maxn],rev[maxn],top,q[maxn],p[maxn];

int find(int x){

return x==p[x]?x:p[x]=find(p[x]);

}

bool isroot(int o){

return tr[fa[o]][0]!=o&&tr[fa[o]][1]!=o;

}

void pushup(int x){

if(!x) return;

sz[x]=sz[tr[x][0]]+sz[tr[x][1]]+1;

}

void re(int o){

if(!o) return;

rev[o]^=1;swap(tr[o][0],tr[o][1]);

}

void pushdown(int x){

if(!x) return;

if(rev[x]){

re(tr[x][0]);re(tr[x][1]);

rev[x]=0;

}

}

void rotate(int &x){

int y=fa[x],z=fa[y],l=(tr[y][1]==x),r=l^1;

if(!isroot(y)) tr[z][tr[z][1]==y]=x;

fa[x]=z;fa[y]=x;fa[tr[x][r]]=y;

tr[y][l]=tr[x][r];tr[x][r]=y;

pushup(y);pushup(x);

}

void splay(int &x){

top=0;q[++top]=x;

for(int i=x;!isroot(i);i=fa[i]) q[++top]=fa[i];

while(top) pushdown(q[top--]);

while(!isroot(x)){

int y=fa[x],z=fa[y];

if(!isroot(y)){

if(tr[y][0]==x^tr[z][0]==y) rotate(x);

else rotate(y);

}

rotate(x);

}

}

void access(int x){

for(int y=0;x;y=x,x=fa[x]){

splay(x);tr[x][1]=y;pushup(x);

}

}

void makeroot(int x){

access(x);splay(x);re(x);

}

void link(int x,int y){

makeroot(x);fa[x]=y;

}

void cut(int x,int y){

makeroot(x);access(y);splay(y);

tr[y][0]=fa[x]=0;pushup(y);

}

int query(int x,int y){

makeroot(x);access(y);splay(y);

return sz[y];

}

int main(){

int n,m;cin>>n>>m;

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

int x;cin>>x;p[x]=i;

sz[i]=1;if(x) link(i,x);

}

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

char s[17];scanf("%s",s+1);

if(s[1]=='Q'){

int x;cin>>x;

query(x,);

}

else if(s[1]=='M'){

int x,y;cin>>x>>y;

if(p[x]) cut(x,p[x]);

if(y) link(x,y),y=p[x];

}

}

return 0;

}

### 吉司机线段树

#include <stdio.h>

#include <algorithm>

#include <iostream>

#define make\_pair mp

#define pii pair<int,int>

using namespace std;

typedef long long ll;

const int maxn=1000007;

int a[maxn],mx[maxn<<2],mn[maxn<<2],cnt[maxn<<2],se[maxn<<2];ll sum[maxn<<2];

void read(int &x){

x=0;int f=1;

char c=getchar();

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

if(c=='-') f=-1;c=getchar();

}

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

x\*=f;

}

char q[67];int top;

void write(ll x){

if(x==0){

putchar('0');

}

else{

if(x<0) putchar('-');

x=x<0?-x:x;

while(x){

q[++top]=x%10+'0';x/=10;

}

while(top) putchar(q[top--]);

}

putchar('\n');

}

void pushup(int o){

int ls=o<<1,rs=ls|1;

if(mx[ls]>mx[rs]) mx[o]=mx[ls],cnt[o]=cnt[ls],se[o]=max(mx[rs],se[ls]);

else if(mx[rs]>mx[ls]) cnt[o]=cnt[rs],mx[o]=mx[rs],se[o]=max(mx[ls],se[rs]);

else cnt[o]=cnt[ls]+cnt[rs],mx[o]=mx[ls],se[o]=max(se[ls],se[rs]);

sum[o]=sum[ls]+sum[rs];

}

void Min(int o,int x){

if(mn[o]!=-1) mn[o]=min(mn[o],x);

else mn[o]=x;

if(mx[o]>x){

sum[o]-=1LL\*cnt[o]\*(mx[o]-x);mx[o]=x;

}

}

void pushdown(int o){

if(mn[o]!=-1){

Min(o<<1,mn[o]);Min(o<<1|1,mn[o]);mn[o]=-1;

}

}

void build(int o,int l,int r){

mn[o]=-1;

if(l==r){

sum[o]=mx[o]=a[l];se[o]=-1;cnt[o]=1;return;

}

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

build(o<<1,l,mid);build(o<<1|1,mid+1,r);

pushup(o);

}

void update(int o,int l,int r,int ql,int qr,int x){

if(x>=mx[o]) return;

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

if(l!=r) pushdown(o);

if(l==ql&&r==qr){

if(x>se[o]||l==r) Min(o,x);

else update(o<<1,l,m,ql,m,x),update(o<<1|1,m+1,r,m+1,qr,x),pushup(o);

return;

}

if(ql<=m&&qr>m) update(o<<1,l,m,ql,m,x),update(o<<1|1,m+1,r,m+1,qr,x);

else if(ql<=m) update(o<<1,l,m,ql,qr,x);

else update(o<<1|1,m+1,r,ql,qr,x);

pushup(o);

}

ll query(int o,int l,int r,int ql,int qr){

if(l!=r) pushdown(o);

if(l==ql&&r==qr) return sum[o];

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

if(ql<=m&&qr>m) return query(o<<1,l,m,ql,m)+query(o<<1|1,m+1,r,m+1,qr);

else if(ql<=m) return query(o<<1,l,m,ql,qr);

else return query(o<<1|1,m+1,r,ql,qr);

}

int query\_max(int o,int l,int r,int ql,int qr){

if(l!=r) pushdown(o);

if(l==ql&&r==qr) return mx[o];

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

if(ql<=m&&qr>m) return max(query\_max(o<<1,l,m,ql,m),query\_max(o<<1|1,m+1,r,m+1,qr));

else if(ql<=m) return query\_max(o<<1,l,m,ql,qr);

else return query\_max(o<<1|1,m+1,r,ql,qr);

}

int main(){

int t;cin>>t;

while(t--){

int n,m;cin>>n>>m;

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

build(1,1,n);

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

int opt;read(opt);int x,y,z;

read(x);read(y);

if(opt==0){

read(z);update(1,1,n,x,y,z);

}

else if(opt==1){

write(query\_max(1,1,n,x,y));

}

else{

write(query(1,1,n,x,y));

}

}

}

return 0;

}

### 大整数运算

#include <stdio.h>

#include <string.h>

#include <algorithm>

using namespace std;

const int maxn=100007;

struct bignum{

int shu[maxn];int len;

bignum(){

memset(shu,0,sizeof(shu));

}

bignum operator = (int a){

char b[30];sprintf(b+1,"%d",a);

return \*this=b;

}

bool operator < (bignum a) const {

if(len!=a.len) return len<a.len;

for(int i=a.len;i;i--) if(shu[i]!=a.shu[i]) return shu[i]<a.shu[i];

return false;

}

bool operator == (bignum a){

if(len!=a.len) return false;

for(int i=a.len;i;i--) if(shu[i]!=a.shu[i]) return false;

return true;

}

bignum operator = (char \*a){

len=strlen(a+1);for(int i=1;a[i]=='0';i++) len--;

for(int i=1;i<=len;i++) shu[i]=a[len-i+1]-'0';

return \*this;

}

bool operator > (bignum a) const{

return a<\*this;

}

void move(){

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

if(shu[i]>9){

if(i==len){

shu[++len]=shu[i]/10;

shu[i]%=10;

}

else shu[i+1]+=shu[i]/10,shu[i]%=10;

}

}

}

bignum operator + (bignum a){

int lm=max(len,a.len);bignum new1;

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

new1.shu[i]=shu[i]+a.shu[i];

}

new1.len=lm;

new1.move();

return new1;

}

bignum operator \* (bignum a){

bignum new1;

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

for(int k=1;k<=a.len;k++)

new1.shu[i+k-1]+=shu[i]\*a.shu[k];

new1.len=len+a.len-1;

new1.move();

return new1;

}

void print(){

for(int i=len;i;i--) printf("%d",shu[i]);

printf("\n");

}

};

int main(){

return 0;

}

# DP

### 数位dp

#include <stdio.h>

#include <math.h>

#include <iostream>

#include <string.h>

#include <algorithm>

using namespace std;

const double eps=1e-6;

int len,num[107],z;

long long dp[107][10][2][2];

long long dfs(int pos,int pre,int you1,int you2,int limit){

if(pos==0){

if(you2) return 1;else return 0;

}

if(pos!=z&&!limit) if(dp[pos][pre][you1][you2]!=-1) {return dp[pos][pre][you1][you2];}

else dp[pos][pre][you1][you2]=0;

int start=(pos==z)?1:0,end=limit?num[pos]:9;

long long ans=0;

for(int i=start;i<=end;i++){

ans+=dfs(pos-1,i,pos==z?0:(i>pre),you2|(you1&(i>pre)),limit&(i==end));

}

if(pos!=z&&!limit) dp[pos][pre][you1][you2]=ans;

return ans;

}

long long check(long long x){

len=0;

while(x){

num[++len]=x%10;

x/=10;

}

long long ans=0;

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

z=i;ans+=dfs(i,0,0,0,0);

}

z=len;ans+=dfs(z,0,0,0,1);

return ans;

}

bool equal(double x,double y){

return fabs(x-y)<eps;

}

int main(){

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

long long l=1,r=1;

long long z=0;double u;scanf("%lf",&u);

for(int i=0;i<=18;i++){

long long ans=-1;

while(l<r){

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

long long x=check(mid);

double u1=1.0\*x/mid;

if(u1>u||equal(u1,u)){

ans=mid;r=mid-1;

}

else l=mid+1;

}

if(ans!=-1){

printf("%lld",ans);return 0;

}

if(i<18) r\*=10;

}

printf("%lf",0);

return 0;

}

### 斜率优化

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

#include <math.h>

#include <vector>

#include <map>

using namespace std;

typedef long long ll;

const int maxn=1000007;

int ql=1,qr=0,q[maxn];

ll sum[maxn],dp[maxn];

ll gety(int x){

return -dp[x]-sum[x]\*sum[x];

}

ll getx(int x){

return sum[x];

}

bool check\_k(int a,int b,int x){

if(gety(b)-gety(a)>=x\*(getx(b)-getx(a))) return true;

else return false;

}

bool check\_poly(int a,int b,int c){

if((gety(c)-gety(b))\*(getx(b)-getx(a))>=(gety(b)-gety(a))\*(getx(c)-getx(b))) return true;

else return false;

}

//dp[i]=dp[k]+(sum[i]-sum[k])^2+m

//y=kx+b --> b=sum[i]^2+m-dp[i] k=-2\*sum[i] x=sum[k] y=-dp[k]-sum[k]^2 b --> max

int main(){

int n,m;

while(scanf("%d%d",&n,&m)==2){

ql=1,qr=0;

q[++qr]=0;

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

int x;scanf("%d",&x);sum[i]=sum[i-1]+x;

while(ql<qr&&check\_k(q[ql],q[ql+1],-2\*sum[i])) ql++;

dp[i]=dp[q[ql]]+(sum[i]-sum[q[ql]])\*(sum[i]-sum[q[ql]])+m;

while(ql<qr&&check\_poly(q[qr-1],q[qr],i)) qr--;

q[++qr]=i;

}

printf("%lld\n",dp[n]);

}

return 0;

}

# 数论

### ax=b ( mod c )

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

using namespace std;

void exgcd(long long a,long long b,long long &x,long long &y){

if(b==0){

x=1;y=0;return;

}

exgcd(b,a%b,y,x);y-=x\*(a/b);

}

long long gcd(long long x,long long y){

return y==0?x:gcd(y,x%y);

}

int main(){

long long a,b,c,x,y,z;

while(scanf("%lld%lld%lld%lld",&a,&b,&c,&z)==4){

if(b%gcd(a,c)!=0){

printf("-1\n");continue;

}

exgcd(a,c,x,y);x\*=b/gcd(a,c);

long long s=c/gcd(a,c);

printf("%lld\n",(x%s+s)%s);

}

return 0;

}

### BSGS

const int modb=76543;

int head[modb],next1[modb],from[modb],e1,to[modb];

void insert(int u,int v){

++e1;from[e1]=u;to[e1]=v;next1[e1]=head[u%modb];head[u%modb]=e1;

}

int find(int x){

for(int i=head[x%modb];i;i=next1[i])

if(from[i]==x) return to[i];

return -1;

}

ll bsgs(int a,int b,int c){

memset(head,0,sizeof(head));e1=0;

if(b==1) return 0;

int len=(int)sqrt(c+0.5);

ll x=1,p=1;

for(int i=0;i<len;i++,p=p\*a%c) insert(p\*b%c,i);

for(ll i=len;;i+=len){

x=x\*p%c;ll j=find(x);

if(j!=-1) return i-j;

if(i>c) return -1;

}

}

### 杜教筛

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

#include <math.h>

#include <vector>

#include <map>

using namespace std;

typedef long long ll;

const int maxn=5000007;

const int N=5000000;

int e1,mu[maxn],p[maxn],sum1[maxn],phi[maxn];

ll sum2[maxn];

void pre(){

phi[1]=mu[1]=1;

for(int i=2;i<=N;i++){

if(!phi[i]){

p[++e1]=i;mu[i]=-1;phi[i]=i-1;

}

for(int k=1;k<=e1&&p[k]\*i<=N;k++){

if(i%p[k]==0){

phi[i\*p[k]]=phi[i]\*p[k];break;

}

else mu[i\*p[k]]=-mu[i],phi[i\*p[k]]=phi[i]\*phi[p[k]];

}

}

for(int i=1;i<=N;i++) sum1[i]=sum1[i-1]+mu[i],sum2[i]=sum2[i-1]+phi[i];

}

int djs1(int x){

if(x<=N) return sum1[x];

if(a1.count(x)) return a1[x];

int ans=1;

for(int l=2,r;l<=x;l=r+1){

r=x/(x/l);

ans-=(r-l+1)\*djs1(x/l);

}

return a1[x]=ans;

}

ll djs2(ll x){

if(x<=N) return sum2[x];

if(a2.count(x)) return a2[x];

ll ans=x\*(x+1)/2;

for(ll l=2,r;l<=x;l=r+1){

r=x/(x/l);

ans-=(r-l+1)\*djs2(x/l);

}

return a2[x]=ans;

}

int main(){

int t;cin>>t;pre();

while(t--){

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

printf("%lld %d\n",djs2(n),djs1(n));

}

return 0;

}

### 扩展gcd

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

using namespace std;

void exgcd(int a,int b,int &x,int &y){

if(b==0){

x=1;y=0;return;

}

exgcd(b,a%b,y,x);y-=(a/b)\*x;

}

int main(){

int x,y;scanf("%d%d",&x,&y);

int a,b;exgcd(x,y,a,b);

cout<<a<<" "<<b<<endl;

return 0;

}

### 扩展中国剩余定理

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

#include <math.h>

#include <vector>

#include <map>

using namespace std;

typedef long long ll;

const int maxn=500007;

ll m[maxn],r[maxn];

ll exgcd(ll a,ll b,ll &x,ll &y){

if(b==0){

x=1;y=0;return a;

}

else{ll gcd=exgcd(b,a%b,y,x);y-=x\*(a/b);return gcd;}

}

ll powm(ll a,ll b,ll mod){

ll ans=0;

while(b){

if(b&1) ans=(ans+a)%mod;

a=(a<<1)%mod;

b>>=1;

}

return (ans+mod)%mod;

}

ll exchina(int n){

ll x,y;ll tot=m[1],ans=r[1];

for(int i=2;i<=n;i++){

ll a=tot,b=m[i],c=(r[i]-ans%m[i]+m[i])%m[i];

ll gcd=exgcd(a,b,x,y);

if(c%gcd!=0) return -1;

x=powm(x,c/gcd,b/gcd);

ans+=x\*tot;

tot\*=(b/gcd);

ans=(ans%tot+tot)%tot;

}

return ans;

}

int main(){

int n;while(scanf("%d",&n)==1){

for(int i=1;i<=n;i++) scanf("%lld%lld",&m[i],&r[i]);ll ans=exchina(n);

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

}

return 0;

}

### 高斯二项式系数

#include <bits/stdc++.h>

#define mp make\_pair

#define pii pair<int,int>

using namespace std;

typedef long long ll;

#define rint register int

const int maxn=100007;

const int inf=(1LL<<29);

const int mod=1e9+7;

int read(){

int x=0;int f=1;

char c=getchar();

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

if(c=='-') f=-1;c=getchar();

}

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

x\*=f;return x;

}

unordered\_map<int,int> ha;

int pown(int a,int b){

int ans=1;

while(b){

if(b&1) ans=(1LL\*ans\*a)%mod;

a=(1LL\*a\*a)%mod;

b>>=1;

}

return ans;

}

int now[100007],pre[100007],a[100007];

int c(int a,int b){

return 1LL\*pre[a]\*pown(pre[b],mod-2)%mod\*pown(pre[a-b],mod-2)%mod;

}

int main(){

cin.tie(0);ios\_base::sync\_with\_stdio(false);

int n=read(),b=read();

for(int i=1;i<=n;i++) a[i]=read(),ha[a[i]]++;

int x=1;pre[0]=1;

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

now[i]=(now[i-1]+x)%mod;

x=(1LL\*x\*b)%mod;

}

for(int i=1;i<=n;i++) pre[i]=(1LL\*pre[i-1]\*now[i])%mod;

int last=0,ans=1;

for(auto i:ha){

ans=(1LL\*ans\*c(last+i.second,i.second))%mod;

last+=i.second;

}

cout<<ans;

return 0;

}

### 格雷码

void gray\_code(int x){

cout<<x<<endl;e1=1;

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

for(int k=e1;k;k--) ans[++e1]=ans[k]^now[i];

}

for(int i=1;i<=e1;i++) cout<<ans[i]<<" ";

}

### 预处理逆元

inv[1]=1;

for(int i=2;i<=n;i++) inv[i]=1LL\*(mod-mod/i)\*inv[mod%i]%mod;

### 拉格朗日插值

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

#include <math.h>

#include <vector>

#include <map>

#define rint register int

using namespace std;

typedef long long ll;

const ll mod=1e9+7;

const int maxn=107;

int a[maxn][maxn],n;

struct poly{

int num[maxn],len;

poly(int x):len(x){

for(int i=0;i<=len;i++) num[i]=0;

}

inline poly operator \* (poly a){

poly new1(len+a.len);

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

for(int k=0;k<=a.len;k++){

new1.num[i+k]=(new1.num[i+k]+1LL\*num[i]\*a.num[k])%mod;

}

return new1;

}

void print(){

for(int i=0;i<=len;i++) printf("%d ",num[i]);

cout<<endl;

}

};

inline int pown(int a,int b){

int ans=1;

while(b){

if(b&1) ans=(1LL\*ans\*a)%mod;

a=(1LL\*a\*a)%mod;

b>>=1;

}

return ans;

}

int inv(int x){

return pown(x,mod-2);

}

void lagarange\_poly\_coe(int \*x,int \*y,int n){

poly coe(n);

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

poly now(0);now.num[0]=1;

int low=1;

for(rint k=0;k<=n;k++){

if(k!=i){

poly a(1);a.num[1]=1;a.num[0]=mod-x[k];

now=now\*a;

low=(1LL\*low\*(x[i]-x[k]+mod))%mod;

}

}

low=inv(low);

for(rint k=0;k<=n;k++){

now.num[k]=(1LL\*now.num[k]\*y[i])%mod;

now.num[k]=(1LL\*now.num[k]\*low)%mod;

(coe.num[k]+=now.num[k])%=mod;

}

}

for(rint i=0;i<=n;i++) cout<<coe.num[i]<<" ";

}

int yy[maxn],xx[maxn];

int main(){

xx[0]=0;yy[0]=1;xx[1]=1;yy[1]=3;xx[2]=2;yy[2]=7;

lagarange\_poly\_coe(xx,yy,2);

return 0;

}

### 整除分块

for(int l=1,r;l<=n;l=r+1){

r=n/(n/l);

ans+=(r-l+1)\*(n/l);

}

### 矩阵乘法

struct matrix{

int n,m;ll shu[107][107];

matrix(){

memset(shu,0,sizeof(shu));

}

matrix operator \* (matrix a){

matrix new1;new1.n=n;new1.m=a.m;

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

for(int k=1;k<=a.m;k++)

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

new1.shu[i][k]+=shu[i][z]\*a.shu[z][k],new1.shu[i][k]%=mod;

return new1;

}

matrix operator + (matrix a){

matrix new1;new1.n=n;new1.m=m;

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

for(int k=1;k<=m;k++){

new1.shu[i][k]=(shu[i][k]+a.shu[i][k])%mod;

}

return new1;

}

void print(){

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

for(int k=1;k<=m;k++){

printf("%d ",shu[i][k]);

}

printf("\n");

}

}

void eye(){

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

}

};

matrix powmat(matrix a,int b){

matrix ans;int first=0;

while(b){

if(b&1){

if(!first) ans=a;

else ans=ans\*a;

first=1;

}

a=a\*a;

b>>=1;

}

return ans;

}

### Min\_25筛

#include <bits/stdc++.h>

#define mp make\_pair

#define pii pair<int,int>

using namespace std;

typedef long long ll;

#define rint register int

const int maxn=500007;

const int mod=1e9+7;

const int inf=(1LL<<29);

int p[maxn],ee,e1;

int id1[maxn],id2[maxn],sqr,sum[maxn];

ll q[maxn<<1],n,g[maxn<<1],h[maxn<<1];

bool vis[maxn];

void prime(int n){

for(int i=2;i<=n;i++){

if(!vis[i]) p[++e1]=i,sum[e1]=(sum[e1-1]+i)%mod;

for(int k=1;k<=e1&&p[k]\*i<=n;k++){

vis[i\*p[k]]=1;

if(i%p[k]==0) break;

}

}

}

void pre(){

for(ll l=1,r;l<=n;l=r+1){

r=n/(n/l);q[++ee]=n/l;

if(q[ee]<=sqr) id1[q[ee]]=ee;

else id2[n/q[ee]]=ee;

g[ee]=(q[ee]&1)?((q[ee]-1)>>1)%mod\*((q[ee]+2)%mod)%mod:((q[ee]+2)>>1)%mod\*((q[ee]-1)%mod)%mod;

h[ee]=(q[ee]-1)%mod;

}

for(int i=1;i<=e1;++i)

for(int k=1;k<=ee&&1LL\*p[i]\*p[i]<=q[k];++k){

int pos=(q[k]/p[i]<=sqr)?id1[q[k]/p[i]]:id2[n/(q[k]/p[i])];

g[k]=(g[k]-p[i]\*(g[pos]-sum[i-1])%mod+mod)%mod;

h[k]=(h[k]-(h[pos]-(i-1))%mod+mod)%mod;

}

}

ll f(ll x,int y){

return x^y;

}

int S(ll x,int y){

if(x<=1||p[y]>x) return 0;

int pos=(x<=sqr)?id1[x]:id2[n/x];

ll ans=(g[pos]-h[pos]+mod-sum[y-1]+(y-1)+mod)%mod;

if(y==1) (ans+=2)%=mod;

for(int i=y;i<=e1&&1LL\*p[i]\*p[i]<=x;++i){

ll p1=p[i],p2=1LL\*p[i]\*p[i];

for(int e=1;p2<=x;++e,p1=p2,p2\*=p[i]){

(ans+=(1LL\*S(x/p1,i+1)\*f(p[i],e)%mod+f(p[i],e+1))%mod)%=mod;

}

}

return ans;

}

int main(){

// cin.tie(0);ios\_base::sync\_with\_stdio(false);

cin>>n;sqr=(int)sqrt(n+0.5);

prime(sqr);pre();

printf("%d",(S(n,1)+1)%mod);

return 0;

}

### 线性筛

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

using namespace std;

const int maxn=10000007;

const int mod=998244353;

int mu[maxn],phi[maxn],sum[maxn];

int len,p[2000007];

int main(){

int n;cin>>n;mu[1]=1;phi[1]=1;

for(int i=2;i<=n;i++){

if(!phi[i]) p[++len]=i,mu[i]=-1,phi[i]=i-1;

for(int k=1;k<=len&&i\*p[k]<=n;k++){

if(i%p[k]!=0) phi[i\*p[k]]=phi[i]\*phi[p[k]],mu[i\*p[k]]=mu[i]\*mu[p[k]];

else {phi[i\*p[k]]=phi[i]\*p[k],mu[i\*p[k]]=0;break;}

}

}

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

if(mu[i]) for(int k=2\*i;k<=n;k+=i) sum[k]=(sum[k]+1LL\*phi[k/i]\*mu[i])%mod;

}

int ans=0;

for(int i=2;i<=n;i++) sum[i]=(sum[i]+sum[i-1])%mod;

for(int i=1;i<=n;i++) ans=(ans+mod+mu[i])%mod;

printf("%d",((sum[n]<<1)%mod+ans)%mod);

return 0;

}

### Pohlig\_Hellman

#include <bits/stdc++.h>

#define mp make\_pair

#define pii pair<int,int>

using namespace std;

typedef long long ll;

#define rint register int

const int maxn=100007;

const int inf=(1LL<<29);

int read(){

int x=0;int f=1;

char c=getchar();

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

if(c=='-') f=-1;c=getchar();

}

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

x\*=f;return x;

}

const int modb=20;

int head[modb],next1[80007],e1;

ll from[80007],to[80007];

inline void insert(ll u,ll v){

++e1;from[e1]=u;to[e1]=v;next1[e1]=head[u%modb];head[u%modb]=e1;

}

inline ll find(ll x){

for(int i=head[x%modb];i;i=next1[i])

if(from[i]==x) return to[i];

return -1;

}

inline ll powm(ll a,ll b,ll mod){

/\*ll ans=0;

while(b){

// printf("a");

if(b&1) ans=(ans+a)%mod;

a=(a<<1)%mod;

b>>=1;

}

return (ans+mod)%mod;\*/

return (a\*b-(ll)((long double)a\*b/mod)\*mod+mod)%mod;

}

inline ll bsgs(ll a,ll b,ll c,ll mod){

memset(head,0,sizeof(head));e1=0;

if(b==1) return 0;

int len=(int)sqrt(c+0.5);

ll x=1,p=1;

for(ll i=0;i<len;i++,p=powm(p,a,mod)) insert(powm(p,b,mod),i);

for(ll i=len;;i+=len){

x=powm(x,p,mod);ll j=find(x);

if(j!=-1) return i-j;

if(i>c) return -1;

}

}

//copy from wiki

ll m[maxn],r[maxn];

inline ll exgcd(ll a,ll b,ll &x,ll &y){

if(b==0){

x=1;y=0;return a;

}

else{ll gcd=exgcd(b,a%b,y,x);y-=x\*(a/b);return gcd;}

}

inline ll exchina(int n){

ll x,y;ll tot=m[1],ans=r[1];

for(int i=2;i<=n;i++){

ll a=tot,b=m[i],c=(r[i]-ans%m[i]+m[i])%m[i];

ll gcd=exgcd(a,b,x,y);

if(c%gcd!=0) return -1;

x=powm(x,c/gcd,b/gcd);

ans+=x\*tot;

tot\*=(b/gcd);

ans=(ans%tot+tot)%tot;

}

return ans;

}

inline ll pown(ll a,ll b,ll mod){

ll ans=1;

while(b){

if(b&1) ans=powm(ans,a,mod);

a=powm(a,a,mod);

b>>=1;

}

return ans;

}

inline ll pohlig\_hellman(ll a,ll b,ll p,ll e,ll mod){

ll r=pown(a,pown(p,e-1,mod),mod),x=0;

for(ll i=0;i<e;++i){

ll h=pown(powm(pown(pown(a,mod-2,mod),x,mod),b,mod),pown(p,e-1-i,mod),mod);

// printf("%lld %lld %lld %lld\n",r,h,p,mod);

ll d=bsgs(r,h,p,mod);

x=x+pown(p,i,mod)\*d;

}

return x;

}

inline ll pohlig\_hellman\_ex(ll a,ll b,ll p[],ll e[],int len,ll mod){

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

ll aa=pown(a,(mod-1)/pown(p[i],e[i],mod),mod);

ll bb=pown(b,(mod-1)/pown(p[i],e[i],mod),mod);

m[i]=pown(p[i],e[i],mod);

r[i]=pohlig\_hellman(aa,bb,p[i],e[i],mod);

}

return exchina(len);

}

ll pp[5],e[3];

inline bool prime\_root(ll x,ll p[],int len,ll mod){

for(register int i=1;i<=len;i++){

if(pown(x,mod/p[i],mod)==1) return false;

}

return true;

}

int main(){

//freopen("testA.in","r",stdin);

//freopen("testA.out","w",stdout);

//cin.tie(0);ios\_base::sync\_with\_stdio(false);

int t;cin>>t;

while(t--){

ll p,a,b;cin>>p>>a>>b;

e[1]=e[2]=0;

ll x=p-1;

ll g=0;

while(x%2==0){

x/=2;e[1]++;

}

while(x%3==0){

x/=3;e[2]++;

}

ll lef=-1,rig=-1;

int len=1;

if(e[1]&&!e[2]){

pp[1]=2;

}

else if(!e[1]&&e[2]){

pp[1]=3;e[1]=e[2];//lef=pohlig\_hellman\_ex(g,a,pp,e,1,p);rig=pohlig\_hellman\_ex(g,b,pp,e,1,p);

}

else{

pp[1]=2;pp[2]=3;len=2;//lef=pohlig\_hellman\_ex(g,a,pp,e,1,p);rig=pohlig\_hellman\_ex(g,b,pp,e,1,p);

}

for(int i=2;;i++){

if(prime\_root(i,pp,len,p)){

g=i;break;

}

}

lef=pohlig\_hellman\_ex(g,a,pp,e,len,p);rig=pohlig\_hellman\_ex(g,b,pp,e,len,p);

if(lef==-1||rig==-1) printf("-1\n");

else{

ll xx,yy;

ll gcd=exgcd(lef,p-1,xx,yy);

ll bg=(p-1)/gcd;

if(rig%gcd){printf("-1\n");continue;}

xx=powm(xx,rig/gcd,bg);

xx=(xx%bg+bg)%bg;

// printf("%lld %lld %lld %lld %lld %lld\n",xx,bg,lef,rig,g,powm(lef,xx,p-1));

printf("%lld\n",xx);

}

}

return 0;

}

### Pollard\_rho

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

#include <string.h>

#include <algorithm>

#include <math.h>

#include <vector>

#include <map>

using namespace std;

typedef long long ll;

vector<ll> fac;

ll gcd(ll a,ll b){

if(a<0) return gcd(-a,b);

return b==0?a:gcd(b,a%b);

}

ll powm(ll a,ll b,ll mod){

ll ans=0;

while(b){

if(b&1) ans=(ans+a)%mod;

a=(a<<1)%mod;

b>>=1;

}

return ans;

}

ll pown(ll a,ll b,ll mod){

ll ans=1;

while(b){

if(b&1) ans=powm(ans,a,mod);

a=powm(a,a,mod);

b>>=1;

}

return ans;

}

bool miller\_rabin(ll n){

if(n==2) return true;

if(n<2||!(n%2)) return false;

long long ind=n-1,last,x;

int cnt=0;

while(ind%2==0){

++cnt;

ind>>=1;

}

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

x=pown(rand()%(n-2)+2,ind,n);last=x;

for(int k=1;k<=cnt;k++){

x=powm(x,x,n);

if(x==1&&last!=1&&last!=n-1) return false;

last=x;

}

if(x!=1) return false;

}

return true;

}

ll pollard\_rho(ll n,ll c){

ll l=1,r=2;

ll x=rand()%n,last=x;

while(1){

l++;x=(powm(x,x,n)+c)%n;

ll d=gcd(last-x,n);

if(1<d&&d<n) return d;

if(last==x) return n;

if(l==r){

last=x;

r<<=1;

}

}

}

void find(ll n){

if(miller\_rabin(n)){

fac.push\_back(n);return;

}

ll p=n;

while(p==n) p=pollard\_rho(p,rand()%(n-1)+1);;

find(p);

find(n/p);

}

int main(){

ll n;

while(scanf("%lld",&n)==1){

fac.clear();if(n>1) find(n);

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

}

return 0;

}

### 二次剩余

#include <bits/stdc++.h>

#define mp make\_pair

#define pii pair<int,int>

using namespace std;

typedef long long ll;

#define rint register int

const int maxn=100007;

const int inf=(1LL<<29);

const int mod=998244353;

struct cn{

int x,y,w;

cn operator \* (cn a){

cn ans;

ans.x=(1LL\*x\*a.x%mod+1LL\*y\*a.y%mod\*w%mod)%mod;

ans.y=(1LL\*x\*a.y%mod+1LL\*y\*a.x%mod)%mod;

ans.w=w;

return ans;

}

int operator ^ (int b){

cn ans,x=\*this;ans.x=1;ans.y=0;ans.w=w;

while(b){

if(b&1) ans=ans\*x;

x=x\*x;

b>>=1;

}

//cout<<ans.x<<" "<<ans.y<<endl;

return ans.x;

}

};

int pown(int a,int b){

int ans=1;

while(b){

if(b&1) ans=(1LL\*ans\*a)%mod;

a=(1LL\*a\*a)%mod;

b>>=1;

}

return ans;

}

int sqrt\_mod(int n){

if(n==0) return 0;

if(pown(n,(mod-1)/2)==mod-1) return -1;

int a,w;

while(1){

a=rand()%mod;

w=(1LL\*a\*a-n+mod)%mod;

if(pown(w,(mod-1)/2)==mod-1) break;

}

cn x;x.x=a;x.y=1;x.w=w;

return x^((mod+1)/2);

}

int main(){

printf("%d\n",sqrt\_mod(2));

return 0;

}

# 博弈

### SG函数

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

using namespace std;

const int maxn=107;

int vis[maxn],sg[maxn];

int main(){

int n;cin>>n;

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

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

for(int j=1;j<=i;j++) vis[sg[i-j]]=1;

for(int j=1;j<i;j++) vis[sg[j]^sg[i-j]]=1;

for(int j=1;;j++){

if(!vis[j]){

sg[i]=j;break;

}

}

printf("%d %d\n",i,sg[i]);

}

return 0;

}

### 尼姆博弈

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

using namespace std;

const int maxn=200007;

int a[maxn];

int main(){

int n;

while(cin>>n){

if(!n) return 0;

for(int i=1;i<=n;i++) scanf("%d",&a[i]);

int sum=0;

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

if(sum==0){

printf("No\n");

}

else{

printf("Yes\n");

int cnt=0;

for(int i=1;i<=n;i++) if((a[i]^sum)<a[i]){

printf("%d %d\n",a[i],a[i]^sum);

}

}

}

return 0;

}

### 威佐夫博弈

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <math.h>

#include <algorithm>

using namespace std;

const double eps=1e-5;

double equal(double x,double y){

return fabs(x-y)<eps;

}

int main(){

int a,b;

while(scanf("%d%d",&a,&b)==2){

if((int)(abs(a-b)\*(sqrt(5.0)+1)/2.0)==min(a,b)){

printf("0\n");

}

else printf("1\n");

}

return 0;

}

# 图论

### Km

#include <stdio.h>

#include <algorithm>

#define mp make\_pair

#define sqr(x) (x)\*(x)

using namespace std;

typedef pair<int,int> pii;

typedef long long ll;

const int maxn=507;

const int inf=2e9;

int read(){

int x=0,f=1;

char ch=getchar();

while(ch<'0'||ch>'9') {if(ch=='-') f=-1;ch=getchar();}

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

return x\*f;

}

int n,wx[maxn],wy[maxn],w[maxn][maxn],vis[maxn],p[maxn],slk[maxn],pre[maxn];

inline ll km(){

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

wx[i]=-inf;wy[i]=p[i]=0;

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

wx[i]=max(wx[i],w[i][k]);

}

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

for(int k=0;k<=n;k++) vis[k]=pre[k]=0,slk[k]=inf;

int now,nex=-1;

for(p[now=0]=i;p[now];now=nex){

int d=inf;

vis[now]=1;

int u=p[now];

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

if(!vis[k]){

int f;

if((f=wx[u]+wy[k]-w[u][k])<slk[k]) slk[k]=f,pre[k]=now;

if(slk[k]<d) d=slk[k],nex=k;

}

}

for(int k=0;k<=n;k++){

if(vis[k]) wx[p[k]]-=d,wy[k]+=d;

else slk[k]-=d;

}

}

for(;now;now=pre[now]) p[now]=p[pre[now]];

}

ll ans=0;

for(int i=1;i<=n;i++) ans+=wx[i]+wy[i];

return ans;

}

int main(){

while(scanf("%d",&n)==1){

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

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

scanf("%d",&w[i][k]);

printf("%lld\n",km());

}

return 0;

}

### 二分图染色

int vis[maxn],flag;

int dfs(int u){

if(!vis[u]) vis[u]=1;

for(int i=0;i<g[u].size();i++){

int v=g[u][i];

if(!vis[v]){

vis[v]=((vis[u]-1)^1)+1;dfs(v);

}

else if(vis[v]!=((vis[u]-1)^1)+1) flag=1;

}

}

### Dinic

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

#include <math.h>

#include <vector>

#include <queue>

using namespace std;

typedef long long ll;

const int maxn=100007;

const ll inf=(1LL<<60);

int cur[maxn],vis[maxn],head[maxn],d[maxn];

int e1,to[maxn<<1],next1[maxn<<1],from[maxn<<1];ll cap[maxn<<1],n;

int s,t;

bool bfs(){

for(int i=1;i<=n;i++) vis[i]=0,d[i]=-1;

queue<int> q;

q.push(s);d[s]=0;vis[s]=1;

while(!q.empty()){

int u=q.front();q.pop();

for(int i=head[u];i;i=next1[i]){

int v=to[i];

if(!vis[v]&&cap[i]){

vis[v]=1;

d[v]=d[u]+1;q.push(v);

}

}

}

return vis[t];

}

ll dfs(int u,ll now){

if(u==t||now==0) return now;

ll ans=0,f;

for(int &i=cur[u];i;i=next1[i]){

int v=to[i];

if(d[u]+1==d[v]&&(f=dfs(v,min(now,cap[i])))>0){

cap[i]-=f;

cap[i^1]+=f;

ans+=f;

now-=f;

if(!now) break;

}

}

return ans;

}

ll dinic(){

ll ans=0;

while(bfs()){

for(int i=1;i<=n;i++) cur[i]=head[i];

ans+=dfs(s,inf);

}

return ans;

}

void addedge(int u,int v,ll cap1){

++e1;from[e1]=u;next1[e1]=head[u];head[u]=e1;to[e1]=v;cap[e1]=cap1;

}

void add(int u,int v,ll cap1){

addedge(u,v,cap1);addedge(v,u,0);

}

void init(){

for(int i=1;i<=n;i++) head[i]=0;for(int i=1;i<=e1;i++) to[i]=0;e1=1;

}

int main(){

int cnt;cin>>cnt;

while(cnt--){

int nn,m;cin>>nn>>m;

}

return 0;

}

### 原始对偶(dijkstra费用流)

#include <bits/stdc++.h>

#define mp make\_pair

#define sqr(x) (x)\*(x)

using namespace std;

typedef pair<int,int> pii;

typedef long long ll;

const int maxn=100007;

const ll inf=1<<29;

int read(){

int x=0,f=1;

char ch=getchar();

while(ch<'0'||ch>'9') {if(ch=='-') f=-1;ch=getchar();}

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

return x\*f;

}

int from[maxn<<1],head[maxn<<1],to[maxn<<1],nex[maxn<<1],e1,n,vis[maxn],s,t;

ll cost[maxn<<1],cap[maxn<<1],mc,mf,delta,d[maxn];

inline void addedge(int u,int v,ll c,ll w){

++e1;nex[e1]=head[u];head[u]=e1;

from[e1]=u;to[e1]=v;cap[e1]=c;cost[e1]=w;

}

inline void add(int u,int v,ll c,ll w){

addedge(u,v,c,w);

addedge(v,u,0,-w);

}

inline void init(){

for(int i=1;i<=n;i++) head[i]=0;

e1=1;mc=mf=0;delta=0;

}

//如果边权为正(不管反向边),则改为dijkstra

//复杂度 普通 KM

//普通 FMlog+NM NMlog

//pbds FM+NM NM

bool spfa(){

queue<int> q;for(int i=1;i<=n;i++) d[i]=inf,vis[i]=0;

d[s]=0;q.push(s);vis[s]=1;

while(!q.empty()){

int u=q.front();q.pop();vis[u]=0;

for(int i=head[u];i;i=nex[i]){

int v=to[i];

if(cap[i]&&d[v]>d[u]+cost[i]){

d[v]=d[u]+cost[i];

if(!vis[v]){

vis[v]=1;

q.push(v);

}

}

}

}

return d[t]!=inf;

}

/\*inline bool dijkstra(){

for(int i=1;i<=n;++i) d[i]=inf,vis[i]=0;

priority\_queue<pair<ll,int>> q;d[s]=0;q.push(mp(0,s));

while(!q.empty()){

int u=q.top().second;q.pop();

if(vis[u]) continue;

vis[u]=1;

for(int i=head[u];i;i=nex[i]){

int v=to[i];

if(cap[i]&&d[v]>d[u]+cost[i]){

d[v]=d[u]+cost[i];

q.push(mp(-d[v],v));

}

}

}

return d[t]!=inf;

}\*/

#include <ext/pb\_ds/priority\_queue.hpp>

typedef pair<ll,int> pli;

typedef \_\_gnu\_pbds::priority\_queue<pli,greater<pli>,\_\_gnu\_pbds::pairing\_heap\_tag> pq;

pq::point\_iterator it[maxn];

inline bool dijkstra(){

for(int i=1;i<=n;++i) d[i]=inf,it[i]=0;

pq q;d[s]=0;

it[s]=q.push(mp(0LL,s));

while(!q.empty()){

int u=q.top().second;q.pop();

for(int i=head[u];i;i=nex[i]){

int v=to[i];

if(cap[i]&&d[v]>d[u]+cost[i]){

d[v]=d[u]+cost[i];

if(it[v]==NULL) it[v]=q.push(mp(d[v],v));

else q.modify(it[v],mp(d[v],v));

}

}

}

return d[t]!=inf;

}

inline ll dfs(int u,ll now){

if(u==t||now==0) return now;

vis[u]=1;

ll ans=0,f;

for(int i=head[u];i;i=nex[i]){

int v=to[i];

if(!vis[v]&&cap[i]&&!cost[i]&&(f=dfs(v,min(now,cap[i])))>0){

cap[i]-=f;cap[i^1]+=f;

ans+=f;now-=f;

if(!now) break;

}

}

return ans;

}

inline void augment(){

for(int i=2;i<=e1;i++) cost[i]+=d[from[i]]-d[to[i]];

delta+=d[t];

ll now=0;

while(memset(vis,0,sizeof(vis)),now=dfs(s,inf)){

mf+=now;

mc+=now\*delta;

printf("%lld %lld\n",mf,mc);

}

}

inline void primal\_dual(){

if(!dijkstra()) return;

augment();

while(dijkstra()) augment();

}

int main(){

int m;

n=read(),m=read(),s=read(),t=read();init();

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

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

add(u,v,x,y);

}

primal\_dual();

printf("%lld %lld",mf,mc);

return 0;

}

### 浮点数版Dinic

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

#include <math.h>

#include <vector>

#include <queue>

using namespace std;

typedef long long ll;

const int maxn=250007;

const double eps=1e-8;

const double inf=1e9;

int cur[maxn],vis[maxn],head[maxn],d[maxn];

int e1,to[maxn<<1],next1[maxn<<1],from[maxn<<1];double cap[maxn<<1];int n;

int s,t;

bool bfs(){

for(int i=1;i<=n;i++) vis[i]=0,d[i]=-1;

queue<int> q;

q.push(s);d[s]=0;vis[s]=1;

while(!q.empty()){

int u=q.front();q.pop();

for(int i=head[u];i;i=next1[i]){

int v=to[i];

if(!vis[v]&&cap[i]>0){

vis[v]=1;

d[v]=d[u]+1;q.push(v);

}

}

}

return vis[t];

}

double dfs(int u,double now){

if(u==t||now==0) return now;

double ans=0,f;

for(int &i=cur[u];i;i=next1[i]){

int v=to[i];

if(d[u]+1==d[v]&&(f=dfs(v,min(now,cap[i])))>0){

cap[i]-=f;

cap[i^1]+=f;

ans+=f;

now-=f;

if(now==0) break;

}

}

return ans;

}

double dinic(){

double ans=0;

while(bfs()){

for(int i=1;i<=n;i++) cur[i]=head[i];

ans+=dfs(s,inf);

}

return ans;

}

void addedge(int u,int v,double cap1){

++e1;from[e1]=u;next1[e1]=head[u];head[u]=e1;to[e1]=v;cap[e1]=cap1;

}

void add(int u,int v,double cap1){

addedge(u,v,cap1);addedge(v,u,0);

}

void init(){

for(int i=1;i<=n;i++) head[i]=0;for(int i=1;i<=e1;i++) to[i]=0;e1=1;

}

int main(){

int tt;cin>>tt;

int nn,m;scanf("%d%d",&nn,&m);

return 0;

}

### Erdos-gallai定理

**原题：给你n个点的度序列，让你求出第n+1个点的个数使得度序列可图化**

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

using namespace std;

const int maxn=500007;

int a[maxn],n;

bool cmp(int a,int b){

return a>b;

}

int eg(int x){

int z[maxn];for(int i=0;i<=n;i++) z[i]=0;

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

long long sum1=0,sum2=0,sum3=0;

int ok=0;

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

int val=0,pos;

if(ok){

val=a[i-1];

}

else{

if(x>a[i]) val=x,ok=1;

else val=a[i];

}

sum1+=val;sum3+=z[i-1];

sum2-=min(i-1,val);

sum2+=max(0LL,n+1-i-sum3);

if(sum1>sum2+1LL\*i\*(i-1)){

if(ok) return 1;

else return -1;

}

}

return 0;

}

int main(){

scanf("%d",&n);

long long sum=0;

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

scanf("%d",&a[i]);sum+=a[i];

}

sort(a+1,a+n+1,cmp);

int p=sum%2;

int l=0,r=(n-p)/2,ans1=-1;

while(l<=r){

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

int x=eg(m\*2+p);

if(x==-1) l=m+1;

else{

r=m-1;

if(x==0) ans1=m;

}

}

l=0,r=(n-p)/2;int ans2=-1;

while(l<=r){

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

int x=eg(m\*2+p);

if(x==1) r=m-1;

else{

l=m+1;

if(x==0) ans2=m;

}

}

if(ans1==-1||ans2==-1) printf("-1");

else{

for(int i=ans1;i<=ans2;i++) printf("%d ",i\*2+p);

}

return 0;

}

### 输出欧拉路/回路

#include <bits/stdc++.h>

#define mp make\_pair

#define pii pair<int,int>

using namespace std;

typedef long long ll;

#define rint register int

const int maxn=200007;

const int inf=(1LL<<29);

map<int,int> head,de;

int to[maxn<<1],next1[maxn<<1],vis[maxn<<1];

int ans[maxn],e1,a[maxn],b[maxn],ee;

void addedge(int u,int v){

++e1;next1[e1]=head[u];head[u]=e1;to[e1]=v;

}

void euler\_path(int u){

for(int &i=head[u];i;i=next1[i]){

if(vis[i]) continue;

vis[i]=vis[((i-1)^1)+1]=1;

euler\_path(to[i]);

}

ans[++ee]=u;

}

int main(){

cin.tie(0);ios\_base::sync\_with\_stdio(false);

int n,m;cin>>n;

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

for(int i=1;i<n;i++) cin>>b[i];

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

if(a[i]>b[i]){

cout<<-1;return 0;

}

addedge(a[i],b[i]);addedge(b[i],a[i]);

de[a[i]]++;de[b[i]]++;

}

int sum=0,now=0;

for(auto it=de.begin();it!=de.end();++it){

if((\*it).second&1){

sum++;now=(\*it).first;

}

}

if(sum==0||sum==2){

euler\_path(sum==0?(\*de.begin()).first:now);

if(ee!=n) cout<<-1;

else for(int i=1;i<=ee;i++) cout<<ans[i]<<" ";

}

else{

cout<<-1;

}

return 0;

}

### 最小生成树

#include <stdio.h>

#include <algorithm>

using namespace std;

const int maxn=10007;

int from[maxn],a[maxn],fa[maxn],head[maxn],to[maxn],next[maxn],e1,w[maxn];

void addedge(int u,int v,int x){

to[++e1]=v;from[e1]=u;next[e1]=head[u];head[u]=e1;w[e1]=x;

}

bool cmp(int x,int y){

return w[x]<w[y];

}

int find(int x){

if(fa[x]==x) return x;

else fa[x]=find(fa[x]);

}

void kruscal(){

for(int i=1;i<=e1;i++) a[i]=i;

sort(a+1,a+e1+1,cmp);

for(int i=1;i<=e1;i++){

int x=find(from[i]),y=find(to[i]);

if(x!=y){

fa[x]=y;ans+=w[i];

}

}

}

int main(){

return 0;

}

### 假次小生成树

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

#include <math.h>

#include <vector>

#include <map>

using namespace std;

typedef long long ll;

const int maxn=200007;

struct edge{

int to,w;

edge(int to,int w):to(to),w(w){

}

};

int fa[maxn],a[maxn<<1],e1,head[maxn],to[maxn<<1],next[maxn<<1],from[maxn<<1],w[maxn<<1],mx[maxn][17];

int cmp(int x,int y){

return w[x]<w[y];

}

vector<edge> g[maxn];

void addedge(int u,int v,int w1){

++e1;next[e1]=head[u];head[u]=e1;to[e1]=v;from[e1]=u;w[e1]=w1;

}

int find(int x){

return x==fa[x]?x:fa[x]=find(fa[x]);

}

int lg[maxn],tp[maxn],dep[maxn],st[maxn][27];

int lca(int u,int v){

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

int ans=0;

for(int i=16;i>=0;i--) if(dep[st[u][i]]>=dep[v]) ans=max(mx[u][i],ans),u=st[u][i];

if(u==v) return ans;

for(int i=16;i>=0;i--) if(st[u][i]!=st[v][i]) ans=max(ans,max(mx[u][i],mx[v][i])),u=st[u][i],v=st[v][i];

return max(ans,mx[u][0]);

}

void dfs(int u,int fa,int val){

st[u][0]=fa;dep[u]=dep[fa]+1;mx[u][0]=val;

for(int i=1;i<=16;i++) st[u][i]=st[st[u][i-1]][i-1],mx[u][i]=max(mx[u][i-1],mx[st[u][i-1]][i-1]);

for(int i=0;i<g[u].size();i++){

int v=g[u][i].to;

if(v!=fa) dfs(v,u,g[u][i].w);

}

}

int vis[maxn<<1];

int main(){

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

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

int u,v,w;scanf("%d%d%d",&u,&v,&w);addedge(u,v,w);

}

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

for(int i=1;i<=e1;i++) a[i]=i;

sort(a+1,a+e1+1,cmp);

ll ans=0;

for(int i=1;i<=e1;i++){

int u=from[i],v=to[i],x=find(u),y=find(v);

if(x!=y){

vis[i]=1;fa[x]=y;ans+=w[i];

}

}

ll pre=ans;ans=(1LL<<60);

for(int i=1;i<=e1;i++){

if(vis[i]){

g[from[i]].push\_back(edge(to[i],w[i]));

g[to[i]].push\_back(edge(from[i],w[i]));

}

}

for(int i=1;i<=e1;i++){

if(vis[i]) continue;

int ans1=lca(from[i],to[i]);

ans=min(ans,pre-ans1+w[i]);

}

printf("%lld",ans);

return 0;

}

### Lca

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

#include <math.h>

#include <vector>

#include <map>

using namespace std;

typedef long long ll;

const int maxn=100007;

int lg[maxn],tp[maxn],dep[maxn],st[maxn][27],head[maxn],to[maxn<<1],next1[maxn<<1],e1;

int lca(int u,int v){

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

for(int i=16;i>=0;i--) if(dep[st[u][i]]>=dep[v]) u=st[u][i];

if(u==v) return u;

for(int i=16;i>=0;i--) if(st[u][i]!=st[v][i]) u=st[u][i],v=st[v][i];

return st[u][0];

}

void dfs(int u,int fa){

st[u][0]=fa;dep[u]=dep[fa]+1;

for(int i=1;i<=16;i++) st[u][i]=st[st[u][i-1]][i-1];

tp[u]=fa?tp[fa]:u;

for(int i=head[u];i;i=next1[i]){

int v=to[i];

if(v!=fa) dfs(v,u);

}

}

void addedge(int u,int v){

++e1;next1[e1]=head[u];head[u]=e1;to[e1]=v;

}

int main(){

return 0;

}

### 矩阵树定理

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

#include <math.h>

#include <vector>

#include <map>

using namespace std;

typedef long long ll;

const ll mod=1e9;

const double eps=1e-7;

ll a[maxn][maxn],n;

ll gauss\_ll(){

ll ans=1;

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

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

while(a[k][i]){

ll t=a[i][i]/a[k][i];

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

a[i][z]=(a[i][z]-(a[k][z]\*t)%mod+mod)%mod;

swap(a[k],a[i]);

ans=-ans;

}

ans=(ans\*a[i][i])%mod;

if(!ans) return 0;

}

return (ans+mod)%mod;

}

double gauss\_double(){

double ans=1.0;

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

int t=i;

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

if(fabs(a[k][i])>fabs(a[t][i])) t=k;

}

if(t!=i) swap(a[t],a[i]),ans=-ans;

ans\*=a[i][i];

if(fabs(ans)<eps) return 0;

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

double t=a[k][i]/a[i][i];

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

a[k][z]-=z\*a[i][z];

}

}

}

return ans;

}

void add(int u,int v){

if(u>v) return;

a[u][u]++;a[v][v]++;

a[u][v]--;a[v][u]--;

}

int main(){

return 0;

}

### Tarjan

#include<bits/stdc++.h>

using namespace std;

const int maxn=100007;

typedef long long ll;

stack <int> s;

vector<int> g[maxn];

int clock1,num;

int scc[maxn],dfn[maxn],low[maxn],scnt;

int e1,head[maxn],to[maxn<<1],next1[maxn<<1];

void tarjan(int u){

dfn[u]=low[u]=++clock1;s.push(u);

for(int i=0;i<g[u].size();i++){

int v=g[u][i];

if(!dfn[v]){

tarjan(v);

low[u]=min(low[u],low[v]);

}else if(!scc[v]){

low[u]=min(low[u],dfn[v]);

}

}

if(low[u]==dfn[u]){

scnt++;int now;

do{

now=s.top();s.pop();

scc[now]=scnt;

}while(!s.empty()&&now!=u);

}

}

void addedge(int u,int v){

++e1;next1[e1]=head[u];head[u]=e1;to[e1]=v;

}

int in[maxn],n;

void init(){

while(!s.empty()) s.pop();

for(int i=1;i<=scnt;i++) sz[i]=head[i]=in[i]=0,gg[i].clear();

for(int i=1;i<=n;i++) dfn[i]=low[i]=scc[i]=0,g[i].clear();

e1=0;clock1=0;scnt=0;

ans.clear();

}

int main(){

int m;cin>>n>>m;

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

int x;scanf("%d",&x);g[i].push\_back(x);

}

for(int i=1;i<=n;i++) if(!dfn[i]) tarjan(i);

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

for(int k=0;k<g[i].size();k++)

if(scc[g[i][k]]!=scc[i]) addedge(scc[i],scc[g[i][k]]),in[scc[i]]++;

}

return 0;

}

### 最小树形图

#include<iostream>

using namespace std;

#include<cstdio>

#include<cstring>

#define MAXN 1005

#define INF 0x7f7f7f7f

typedef \_\_int64 type;

struct node//边的权和顶点

{

    int u, v;

    type w;

}edge[MAXN \* MAXN];

int pre[MAXN], id[MAXN], vis[MAXN], n, m, pos;

type in[MAXN];//存最小入边权,pre[v]为该边的起点

type Directed\_MST(int root, int V, int E)

{

    type ret = 0;//存最小树形图总权值

    while(true)

    {

        int i;

        //1.找每个节点的最小入边

        for( i = 0; i < V; i++)

            in[i] = INF;//初始化为无穷大

        for( i = 0; i < E; i++)//遍历每条边

        {

            int u = edge[i].u;

            int v = edge[i].v;

            if(edge[i].w < in[v] && u != v)//说明顶点v有条权值较小的入边  记录之

            {

                pre[v] = u;//节点u指向v

                in[v] = edge[i].w;//最小入边

                if(u == root)//这个点就是实际的起点

                    pos = i;

            }

        }

        for( i = 0; i < V; i++)//判断是否存在最小树形图

        {

            if(i == root)

                continue;

            if(in[i] == INF)

                return -1;//除了根以外有点没有入边,则根无法到达它  说明它是独立的点 一定不能构成树形图

        }

        //2.找环

        int cnt = 0;//记录环数

        memset(id, -1, sizeof(id));

        memset(vis, -1, sizeof(vis));

        in[root] = 0;

        for( i = 0; i < V; i++) //标记每个环

        {

            ret += in[i];//记录权值

            int v = i;

            while(vis[v] != i && id[v] == -1 && v != root)

            {

                vis[v] = i;

                v = pre[v];

            }

            if(v != root && id[v] == -1)

            {

                for(int u = pre[v]; u != v; u = pre[u])

                    id[u] = cnt;//标记节点u为第几个环

                id[v] = cnt++;

            }

        }

        if(cnt == 0)

            break; //无环   则break

        for( i = 0; i < V; i++)

            if(id[i] == -1)

                id[i] = cnt++;

            //3.建立新图   缩点,重新标记

            for( i = 0; i < E; i++)

            {

                int u = edge[i].u;

                int v = edge[i].v;

                edge[i].u = id[u];

                edge[i].v = id[v];

                if(id[u] != id[v])

                    edge[i].w -= in[v];

            }

            V = cnt;

            root = id[root];

    }

    return ret;

}

int main()

{

    int i;

    while(scanf("%d%d", &n, &m) != EOF)

    {

        type sum = 0;

        for( i = 0; i < m; i++)

        {

            scanf("%d%d%I64d", &edge[i].u, &edge[i].v, &edge[i].w);

            edge[i].u++; edge[i].v++;

            sum += edge[i].w;

        }

      sum ++;

        for( i = m; i < m + n; i++)//增加超级节点0,节点0到其余各个节点的边权相同（此题中 边权要大于原图的总边权值）

        {

            edge[i].u = 0;

            edge[i].v = i - m + 1;

            edge[i].w = sum;

        }

        type ans = Directed\_MST(0, n + 1, m + n);

        //n+1为总结点数,m+n为总边数

        //ans代表以超级节点0为根的最小树形图的总权值,

        //将ans减去sum,如果差值小于sum,说明节点0的出度只有1,说明原图是连通图

        //如果差值>=sum,那么说明节点0的出度不止为1,说明原图不是连通图

        if(ans == -1 || ans - sum >= sum)

            puts("impossible");

        else

            printf("%I64d %d\n",ans - sum, pos - m);

        puts("");

    }

    return 0;

}

### 差分约束

#include <stdio.h>

#include <algorithm>

#include <string.h>

#include <math.h>

#include <queue>

#define mp make\_pair

#define pii pair<int,int>

using namespace std;

typedef long long ll;

#define rint register int

const int maxn=230007;

const ll inf=(1LL<<60);

//d[i]-d[j]>=w add(j,i,w) maxlen min(d[i])

//d[i]-d[j]<=w add(j,i,w) minlen max(d[i])

int read(){

int x=0;int f=1;

char c=getchar();

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

if(c=='-') f=-1;c=getchar();

}

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

x\*=f;return x;

}

int n,e1,nex[maxn],head[maxn],to[maxn],w[maxn],inq[maxn],cnt[maxn],s;

ll d[maxn];

void addedge(int u,int v,int x){

++e1;nex[e1]=head[u];head[u]=e1;to[e1]=v;w[e1]=x;

}

bool spfa(){

queue<int> q;q.push(s);

for(int i=1;i<=n;i++) inq[i]=cnt[i]=0,d[i]=-inf;

inq[s]=1;d[s]=0;

while(!q.empty()){

int u=q.front();q.pop();inq[u]=0;

// printf("asd%d\n",u);

for(int i=head[u];i;i=nex[i]){

int v=to[i];

// printf("qq%d %d\n",u,w[i]);

if(d[v]<d[u]+w[i]){

d[v]=d[u]+w[i];

if(!inq[v]){

inq[v]=1;q.push(v);

cnt[v]++;

if(cnt[v]>n){

return false;

}

}

}

}

}

return true;

}

void init(){

for(int i=1;i<=n;i++) head[i]=0;e1=0;

for(int i=1;i<n;i++) addedge(s,i,0);

}

int a[maxn];

int main(){

// cin.tie(0);ios\_base::sync\_with\_stdio(false);

int N,m;

while(scanf("%d%d",&N,&m)==2){

n=N+1;s=n;init();

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

char S[10];scanf("%s",S+1);int u=read(),v=read(),x;

if(S[1]=='P'){

x=read();

addedge(v,u,x);

addedge(u,v,-x);

}

else{

addedge(v,u,1);

}

}

if(spfa()) printf("Reliable\n");

else printf("Unreliable\n");

}

return 0;

}

### Dfs版spfa

#include <stdio.h>

#include <queue>

#include <algorithm>

#include <string.h>

#include <math.h>

#define mp make\_pair

#define pii pair<int,int>

#pragma GCC optimize(2)

using namespace std;

typedef long long ll;

#define rint register int

const int maxn=200007;

const double inf = 1e9;

//d[i]-d[j]>=w add(j,i,w) maxlen min(d[i])

//d[i]-d[j]<=w add(j,i,w) minlen max(d[i])

int read(){

int x=0;int f=1;

char c=getchar();

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

if(c=='-') f=-1;c=getchar();

}

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

x\*=f;return x;

}

double d[maxn],w[maxn];

int n,e1,nex[maxn],head[maxn],to[maxn],vis[maxn];

inline void addedge(int u,int v,double x){

++e1;nex[e1]=head[u];head[u]=e1;to[e1]=v;w[e1]=x;

}

inline bool dfs(int u){

vis[u]=1;

for(int i=head[u];i;i=nex[i]){

int v=to[i];

if(d[v]<d[u]+w[i]){

d[v]=d[u]+w[i];

if(vis[v]) return true;

if(dfs(v)) return true;

}

}

vis[u]=0;

return false;

}

inline void init(){

for(int i=1;i<=n;i++) head[i]=0;e1=0;

}

int main(){

n=26\*26+5;init();

//无超级原点，d设为0保证从正权值开始更新

for(rint i=1;i<=n;++i) d[i]=vis[i]=0;

int OK=0;

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

if(dfs(k)){

OK=1;break;

}

}

return 0;

}

### 割点

#include <stdio.h>

#include <stack>

#include <vector>

#include <string>

#include <iostream>

#include <algorithm>

using namespace std;

const int maxn=100007;

typedef long long ll;

vector<int> g[maxn];

int n,cl,dfn[maxn],low[maxn],cut[maxn];

void tarjan(int u,int fa){

int rc=0;

dfn[u]=low[u]=++cl;

for(int i=0;i<g[u].size();i++){

int v=g[u][i];

if(!dfn[v]){

rc++;

tarjan(v,u);

low[u]=min(low[u],low[v]);

if(low[v]>=dfn[u]) cut[u]=1;

}

else if(v!=fa) low[u]=min(low[u],dfn[v]);

}

if(!fa&&rc==1) cut[u]=0;

}

void init(){

for(int i=1;i<=n;i++) g[i].clear(),dfn[i]=cut[i]=0;

cl=0;

}

int read(){

int x=0,f=1;

char ch=getchar();

while(ch<'0'||ch>'9') {if(ch=='-') f=-1;ch=getchar();}

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

return x\*f;

}

int main(){

n=read();init():

for(int i=1;i<=n;i++) if(!dfn[i]) tarjan(i,0);

int num=0;

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

if(cut[i]) num++;

}

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

return 0;

}

### 边双连通

#include <stdio.h>

#include <stack>

#include <vector>

#include <string>

#include <iostream>

#include <sstream>

#include <algorithm>

using namespace std;

const int maxn=800007;

typedef long long ll;

int n,cl,dfn[maxn],low[maxn];

int head[maxn],to[maxn<<1],nex[maxn<<1],e1;

int bans=0,bcc[maxn],bcnt,br[maxn<<1];

inline void tarjan(int u,int fa){

dfn[u]=low[u]=++cl;

for(register int i=head[u];i;i=nex[i]){

int v=to[i];

if(!dfn[v]){

tarjan(v,i);

low[u]=min(low[u],low[v]);

if(low[v]>dfn[u]){

bans++;

br[i]=br[((i-1)^1)+1]=1;

}

}

else if(!fa||i-1!=((fa-1)^1)) low[u]=min(low[u],dfn[v]);

//能够处理重边

}

}

//边双连通

inline void dfs(int u){

dfn[u]=1;

bcc[u]=bcnt;

for(int i=head[u];i;i=nex[i]){

int v=to[i];

if(br[i]) continue;

if(!dfn[v]) dfs(v);

}

}

vector<int> g[maxn];

inline void addedge(int u,int v){

++e1;nex[e1]=head[u];head[u]=e1;to[e1]=v;

}

inline void init(){

for(register int i=1;i<=n;++i) head[i]=dfn[i]=0;

for(int i=1;i<=e1;i++) br[i]=0;

for(int i=1;i<=bcnt;i++) g[i].clear();

cl=e1=bans=bcnt=0;

}

inline int read(){

int x=0,f=1;

char ch=getchar();

while(ch<'0'||ch>'9') {if(ch=='-') f=-1;ch=getchar();}

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

return x\*f;

}

int main(){

int m;

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

init();

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

int u,v;scanf("%d%d",&u,&v);

addedge(u,v);addedge(v,u);

}

bans=0;

tarjan(1,0);

for(int i=1;i<=n;++i) dfn[i]=0;

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

if(!dfn[i]){

++bcnt;dfs(i);

}

}

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

for(int k=head[i];k;k=nex[k]){

int v=to[k];

if(i<v&&bcc[i]!=bcc[v]){

g[bcc[i]].push\_back(bcc[v]);

g[bcc[v]].push\_back(bcc[i]);

}

}

}

return 0;

}

# 多项式相关

### 带了假MTT的分治NTT

#include<bits/stdc++.h>

#define ll long long

using namespace std;

const int maxn=300007;

const ll Mod=1337006139375617;

const int mod=100003,G=3;

int rev[maxn<<2];

ll fm(ll a,ll b,ll mod) {

return (a\*b - (ll)((long double)a/mod\*b)\*mod+mod)%mod;

}

int read(){

int x=0;int f=1;

char c=getchar();

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

if(c=='-') f=-1;c=getchar();

}

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

x\*=f;return x;

}

ll pown(ll a,ll b,ll mod){

if(b<0){

b=-b;a=pown(a,mod-2,mod);

}

ll ans=1;

while(b){

if(b&1) ans=fm(ans,a,mod);

a=fm(a,a,mod);

b>>=1;

}

return ans;

}

void ntt(vector<ll> &a,int n,int dft){

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

if(i<rev[i]) swap(a[i],a[rev[i]]);

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

ll wn=pown(G,dft\*(Mod-1)/(i\*2),Mod);

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

ll wnk=1;

for(int j=k;j<k+i;j++){

ll x=a[j],y=fm(wnk,a[j+i],Mod);

a[j]=(x+y)%Mod;a[j+i]=(x-y+Mod)%Mod;

wnk=fm(wnk,wn,Mod);

}

}

}

if(dft==-1){

ll inv1=pown(n,Mod-2,Mod);

for(int i=0;i<n;i++) a[i]=fm(a[i],inv1,Mod);

}

}

int e1;

struct poly{

int len;vector<ll> a;

poly(vector<ll> a):a(a){

len=a.size()-1;

}

poly(){

len=0;

}

void change(){

while(len&&a[len]==0) len--,a.pop\_back();

}

poly operator \* (poly x){

poly nex;nex.len=len+x.len;

int bit=1;

while((1<<bit)<=nex.len) bit++;

for(int i=0;i<(1<<bit);i++) rev[i]=(rev[i>>1]>>1)|((i&1)<<(bit-1));

vector<ll> A,B;

for(int i=0;i<=len;i++) A.push\_back(a[i]);

for(int i=0;i<=x.len;i++) B.push\_back(x.a[i]);

for(int i=len+1;i<(1<<bit);i++) A.push\_back(0);

for(int i=x.len+1;i<(1<<bit);i++) B.push\_back(0);

ntt(A,1<<bit,1);ntt(B,1<<bit,1);

for(int i=0;i<(1<<bit);i++) A[i]=fm(A[i],B[i],Mod);

ntt(A,1<<bit,-1);

for(int i=0;i<=nex.len;i++) nex.a.push\_back(A[i]%mod);

return nex;

}

void print(){

for(int i=0;i<=len;i++) cout<<a[i]<<" ";

cout<<"\n";

}

}pp[100007];

unordered\_map<int,int> aa;

poly ntt(int l,int r){

if(l==r) return pp[l];

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

return ntt(l,m)\*ntt(m+1,r);

}

int zz[100007],val[100007];

int pre[100007];

int c(int x,int y){

return 1LL\*pre[x]\*pown(pre[y],mod-2,mod)%mod\*pown(pre[x-y],mod-2,mod)%mod;

}

int main(){

int n,q,a;cin>>n>>a>>q;

zz[0]=1;

pre[0]=1;

for(int i=1;i<=n;i++) pre[i]=1LL\*pre[i-1]\*i%mod;

for(int i=1;i<mod-1;i++) zz[i]=1LL\*zz[i-1]\*a%mod;

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

val[i]=read();

pp[i].len=1;

pp[i].a.push\_back(1);

pp[i].a.push\_back(zz[val[i]%100002]);

}

poly A=ntt(1,n);

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

int r=read();

printf("%d\n",1LL\*(A.a[r]-c(n,r)+mod)%mod\*pown(a-1,mod-2,mod)%mod);

}

return 0;

}

### FFT

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

#include <math.h>

#include <vector>

#include <map>

#include <complex>

using namespace std;

#pragma GCC optimize(3)

typedef long long ll;

const double PI=acos(-1.0);

struct cd{

double real,unreal;

cd(double real=0.0,double unreal=0.0):real(real),unreal(unreal){

}

cd operator + (cd a){

return cd(real+a.real,unreal+a.unreal);

}

cd operator - (cd a){

return cd(real-a.real,unreal-a.unreal);

}

cd operator \* (cd a){

return cd(real\*a.real-unreal\*a.unreal,real\*a.unreal+unreal\*a.real);

}

cd operator = (ll a){

return \*this=cd(a,0);

}

cd operator = (complex<double> a){

return \*this=cd(a.real(),a.imag());

}

cd operator / (ll a){

return cd(real/a,unreal/a);

}

};

const int maxn=200007;

int read(){

int x=0;int f=1;

char c=getchar();

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

if(c=='-') f=-1;c=getchar();

}

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

x\*=f;return x;

}

int rev[maxn<<1];

void fft(cd \*a,int n,int dft){

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

if(i<rev[i]) swap(a[i],a[rev[i]]);

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

cd wn;wn=exp(complex<double>(0,dft\*PI/i));

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

cd wnk(1,0);

for(register int j=k;j<k+i;++j){

cd x=a[j],y=wnk\*a[j+i];

a[j]=x+y;a[j+i]=x-y;wnk=wnk\*wn;

}

}

}

if(dft==-1){

for(int i=0;i<n;i++) a[i]=a[i]/n;

}

}

cd A[maxn<<1],B[maxn<<1],C[maxn<<1];

void fttmul(cd \*a,cd \*b,cd \*c,int len){

int bit=1;

while((1<<bit)<=len) bit++;

for(int i=0;i<(1<<bit);i++) rev[i]=(rev[i>>1]>>1)|((i&1)<<(bit-1));

for(int i=len/2+1;i<=(1<<bit);i++) a[i]=b[i]=c[i]=0;

fft(a,1<<bit,1);fft(b,1<<bit,1);

for(int i=0;i<(1<<bit);i++) c[i]=a[i]\*b[i];

fft(c,1<<bit,-1);

}

int main(){

}

### FWT

#include <bits/stdc++.h>

#define make\_pair mp

#define pii pair<int,int>

using namespace std;

typedef long long ll;

const int mod=1e9+7;

const int maxn=100007;

int inv2;

int powm(int a,int b){

int ans=1;

while(b){

if(b&1) ans=(1LL\*ans\*a)%mod;

a=(1LL\*a\*a)%mod;

b>>=1;

}

return ans;

}

void fwt\_or(int \*a,int n,int opt)

{

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

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

for(int j=k;j<k+i;++j)

if(opt==1)a[j+i]=(a[j]+a[j+i])%mod;

else a[j+i]=(a[j+i]-a[j]+mod)%mod;

}

void fwt\_and(int \*a,int n,int opt)

{

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

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

for(int j=k;j<i+k;++j)

if(opt==1)a[j]=(a[j]+a[j+i])%mod;

else a[j]=(a[j]-a[j+i]+mod)%mod;

}

void fwt\_xor(int \*a,int n,int opt)

{

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

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

for(int j=k;j<i+k;++j)

{

int x=a[j],y=a[j+i];

a[j]=(x+y)%mod;a[j+i]=(x-y+mod)%mod;

if(opt==-1) a[j]=1LL\*a[j]\*inv2%mod,a[j+i]=1LL\*a[j+i]\*inv2%mod;

}

}

int main(){

int n;cin>>n;

inv2=powm(2,mod-2);

return 0;

}

### 不带任意模的MTT

#include <bits/stdc++.h>

#define mp make\_pair

#define pii pair<int,int>

using namespace std;

typedef long long ll;

#define rint register int

const int maxn=200007;

const int mod1=479\*(1<<21)+1,G=3,mod2=998244353;

int rev[maxn<<2];

ll pown(ll a,ll b,int mod){

if(b<0){

b=-b;a=pown(a,mod-2,mod);

}

ll ans=1;

while(b){

if(b&1) ans=(ans\*a)%mod;

a=(a\*a)%mod;

b>>=1;

}

return ans;

}

void ntt(ll \*a,int n,int dft,int mod){

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

if(i<rev[i]) swap(a[i],a[rev[i]]);

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

ll wn=pown(G,dft\*(mod-1)/(i\*2),mod);

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

ll wnk=1;

for(int j=k;j<k+i;j++){

int x=a[j],y=(wnk\*a[j+i])%mod;

a[j]=(x+y)%mod;a[j+i]=(x-y+mod)%mod;

wnk=(wnk\*wn)%mod;

}

}

}

if(dft==-1){

int inv1=pown(n,mod-2,mod);

for(int i=0;i<n;i++) a[i]=(a[i]\*inv1)%mod;

}

}

void nttmul(ll \*a,ll \*b,int l1,int l2,int mod){

int bit=1;

while((1<<bit)<=l1+l2) bit++;

for(int i=0;i<(1<<bit);i++) rev[i]=(rev[i>>1]>>1)|((i&1)<<(bit-1));

ntt(a,1<<bit,1,mod);ntt(b,1<<bit,1,mod);

for(int i=0;i<(1<<bit);i++) a[i]=(a[i]\*b[i])%mod;

ntt(a,1<<bit,-1,mod);

}

ll m[maxn],r[maxn];

ll exgcd(ll a,ll b,ll &x,ll &y){

if(b==0){

x=1;y=0;return a;

}

else{ll gcd=exgcd(b,a%b,y,x);y-=x\*(a/b);return gcd;}

}

ll powm(ll a,ll b,ll mod){

ll ans=0;

while(b){

if(b&1) ans=(ans+a)%mod;

a=(a<<1)%mod;

b>>=1;

}

return (ans+mod)%mod;

}

ll exchina(int n){

ll x,y;ll tot=m[1],ans=r[1];

for(int i=2;i<=n;i++){

ll a=tot,b=m[i],c=(r[i]-ans%m[i]+m[i])%m[i];

ll gcd=exgcd(a,b,x,y);

if(c%gcd!=0) return -1;

x=powm(x,c/gcd,b/gcd);

ans+=x\*tot;

tot\*=(b/gcd);

ans=(ans%tot+tot)%tot;

}

return ans;

}

ll a1[200007<<2],b1[200007<<2],a2[200007<<2],b2[200007<<2];

int main(){

cin.tie(0);ios\_base::sync\_with\_stdio(false);

int n;ll x;cin>>n>>x;

nttmul(a1,b1,n,n,mod1);nttmul(a2,b2,n,n,mod2);

for(int i=n+1;i<=2\*n;i++){

m[1]=mod1;r[1]=a1[i];

m[2]=mod2;r[2]=a2[i];

printf("%lld ",exchina(2));

}

return 0;

}

### 多项式开根，求逆，除法，求模等各种操作

#include<bits/stdc++.h>

#define ll long long

using namespace std;

const int maxn=300007;

const int mod=998244353,G=3;

int rev[maxn<<2];ll tmp[maxn<<2],invb[maxn<<2],texp[maxn<<2];

ll pown(ll a,ll b){

if(b<0){

b=-b;a=pown(a,mod-2);

}

ll ans=1;

while(b){

if(b&1) ans=(ans\*a)%mod;

a=(a\*a)%mod;

b>>=1;

}

return ans;

}

struct cn{

int x,y,w;

cn operator \* (cn a){

cn ans;

ans.x=(1LL\*x\*a.x%mod+1LL\*y\*a.y%mod\*w%mod)%mod;

ans.y=(1LL\*x\*a.y%mod+1LL\*y\*a.x%mod)%mod;

ans.w=w;

return ans;

}

int operator ^ (int b){

cn ans,x=\*this;ans.x=1;ans.y=0;ans.w=w;

while(b){

if(b&1) ans=ans\*x;

x=x\*x;

b>>=1;

}

return ans.x;

}

};

int sqrt\_mod(int n){

if(n==0) return 0;

if(pown(n,(mod-1)/2)==mod-1) return -1;

int a,w;

while(1){

a=rand()%mod;

w=(1LL\*a\*a-n+mod)%mod;

if(pown(w,(mod-1)/2)==mod-1) break;

}

cn x;x.x=a;x.y=1;x.w=w;

return x^((mod+1)/2);

}

void ntt(ll \*a,int n,int dft){

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

if(i<rev[i]) swap(a[i],a[rev[i]]);

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

ll wn=pown(G,dft\*(mod-1)/(i\*2));

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

ll wnk=1;

for(int j=k;j<k+i;j++){

int x=a[j],y=(wnk\*a[j+i])%mod;

a[j]=(x+y)%mod;a[j+i]=(x-y+mod)%mod;

wnk=(wnk\*wn)%mod;

}

}

}

if(dft==-1){

int inv1=pown(n,mod-2);

for(int i=0;i<n;i++) a[i]=(a[i]\*inv1)%mod;

}

}

void derivation(ll \*a,ll \*b,int len){

for(int i=1;i<=len;i++) b[i-1]=(i\*a[i])%mod;

b[len]=0;

}

void integral(ll \*a,ll \*b,int len){

for(int i=1;i<=len;i++) b[i]=a[i-1]\*pown(i,mod-2)%mod;

b[0]=0;

}

void poly\_inv(ll \*a,ll \*b,int len){

if(!len){

b[0]=pown(a[0],mod-2);

return;

}

poly\_inv(a,b,len/2);

int bit=1;

while((1<<bit)<=(len<<1)) bit++;

for(int i=0;i<(1<<bit);i++) rev[i]=(rev[i>>1]>>1)|((i&1)<<(bit-1));

for(int i=len/2+1;i<(1<<bit);i++) b[i]=0;

for(int i=0;i<=len;i++) tmp[i]=a[i];

for(int i=len+1;i<(1<<bit);i++) tmp[i]=0;

ntt(tmp,(1<<bit),1),ntt(b,(1<<bit),1);

for(int i=0;i<(1<<bit);i++) (b[i]\*=(2-tmp[i]\*b[i]%mod+mod)%mod)%=mod;

ntt(b,(1<<bit),-1);for(int i=len+1;i<(1<<bit);i++) b[i]=0;

}

void nttmul(ll \*a,ll \*b,int l1,int l2){

int bit=1;

while((1<<bit)<=l1+l2) bit++;

for(int i=0;i<(1<<bit);i++) rev[i]=(rev[i>>1]>>1)|((i&1)<<(bit-1));

for(int i=l1+1;i<(1<<bit);i++) a[i]=0;

for(int i=l2+1;i<(1<<bit);i++) b[i]=0;

ntt(a,1<<bit,1);ntt(b,1<<bit,1);

for(int i=0;i<(1<<bit);i++) a[i]=(a[i]\*b[i])%mod;

ntt(a,1<<bit,-1);

}

int inv2=pown(2,mod-2);

void poly\_sqrt(ll \*a,ll \*b,int len){

if(!len){

b[0]=sqrt\_mod(a[0]);

b[0]=min(b[0],mod-b[0]);

return;

}

poly\_sqrt(a,b,len/2);poly\_inv(b,invb,len);

int bit=1;

while((1<<bit)<=(len<<1)) bit++;

for(int i=0;i<(1<<bit);i++) rev[i]=(rev[i>>1]>>1)|((i&1)<<(bit-1));

for(int i=len/2+1;i<(1<<bit);i++) b[i]=0;

for(int i=0;i<=len;i++) tmp[i]=a[i];

for(int i=len+1;i<(1<<bit);i++) tmp[i]=0;

ntt(b,(1<<bit),1);ntt(invb,(1<<bit),1);ntt(tmp,(1<<bit),1);

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

b[i]=(b[i]+tmp[i]\*invb[i])%mod\*inv2%mod;

}

ntt(b,(1<<bit),-1);for(int i=len+1;i<(1<<bit);i++) b[i]=0;

}

void poly\_ln(ll \*a,ll \*b,int len){

derivation(a,b,len);

poly\_inv(a,invb,len);

nttmul(invb,b,len,len);

integral(invb,b,len);

}

ll a[maxn<<2],b[maxn<<2];

void poly\_exp(ll \*a,ll \*b,int len){

if(!len){

b[0]=1;return;

}

poly\_exp(a,b,len/2);poly\_ln(b,texp,len);

int bit=1;

while((1<<bit)<=(len<<1)) bit++;

for(int i=0;i<(1<<bit);i++) rev[i]=(rev[i>>1]>>1)|((i&1)<<(bit-1));

for(int i=len/2+1;i<(1<<bit);i++) b[i]=0;

for(int i=0;i<=len;i++) tmp[i]=a[i];

for(int i=len+1;i<(1<<bit);i++) tmp[i]=0;

for(int i=len+1;i<(1<<bit);i++) texp[i]=0;

ntt(tmp,1<<bit,1);ntt(texp,1<<bit,1);ntt(b,1<<bit,1);

for(int i=0;i<(1<<bit);i++) b[i]=b[i]\*(mod+1-texp[i]+tmp[i])%mod;

ntt(b,1<<bit,-1);

for(int i=len+1;i<(1<<bit);i++) b[i]=0;

}

ll di[maxn<<2],md[maxn<<2];

void poly\_div(ll \*a,ll \*b,ll \*c,int l1,int l2){

for(int i=0;i<=l1;++i) invb[l1-i]=a[i];

for(int i=0;i<=l2;++i) texp[l2-i]=b[i];

for(int i=l2+1;i<=l1-l2;i++) texp[i]=0;

poly\_inv(texp,di,l1-l2);

nttmul(invb,di,l1,l1-l2);

for(int i=0;i<=l1-l2;i++) c[i]=invb[l1-l2-i];

}

void poly\_mod(ll \*a,ll \*b,ll \*c,int l1,int l2){

poly\_div(a,b,md,l1,l2);

for(int i=0;i<=l2;i++) di[i]=b[i];

nttmul(md,di,l1-l2,l2);

for(int i=0;i<=l1;i++) c[i]=(a[i]-md[i]+mod)%mod;

}

int main(){

int n,m;cin>>n>>m;

for(int i=0;i<=n;i++) cin>>a[i];

for(int i=0;i<=m;i++) cin>>b[i];

poly\_div(a,b,di,n,m);

for(int i=0;i<=n-m;i++) cout<<di[i]<<" ";

cout<<"\n";

poly\_mod(a,b,a,n,m);

for(int i=0;i<m;i++) cout<<a[i]<<" ";

return 0;

}

### 杜教BM

#include <cstdio>

#include <cstring>

#include <cmath>

#include <algorithm>

#include <vector>

#include <string>

#include <map>

#include <set>

#include <cassert>

#include<bits/stdc++.h>

using namespace std;

#define rep(i,a,n) for (int i=a;i<n;i++)

#define per(i,a,n) for (int i=n-1;i>=a;i--)

#define pb push\_back

#define mp make\_pair

#define all(x) (x).begin(),(x).end()

#define fi first

#define se second

#define SZ(x) ((int)(x).size())

typedef vector<int> VI;

typedef long long ll;

typedef pair<int,int> PII;

const ll mod=1000000007;

ll powmod(ll a,ll b) {ll res=1;a%=mod; assert(b>=0); for(;b;b>>=1){if(b&1)res=res\*a%mod;a=a\*a%mod;}return res;}

// head

int \_,n;

namespace linear\_seq {

const int N=10010;

ll res[N],base[N],\_c[N],\_md[N];

vector<int> Md;

void mul(ll \*a,ll \*b,int k) {

rep(i,0,k+k) \_c[i]=0;

rep(i,0,k) if (a[i]) rep(j,0,k) \_c[i+j]=(\_c[i+j]+a[i]\*b[j])%mod;

for (int i=k+k-1;i>=k;i--) if (\_c[i])

rep(j,0,SZ(Md)) \_c[i-k+Md[j]]=(\_c[i-k+Md[j]]-\_c[i]\*\_md[Md[j]])%mod;

rep(i,0,k) a[i]=\_c[i];

}

int solve(ll n,VI a,VI b) { // a 系数 b 初值 b[n+1]=a[0]\*b[n]+...

// printf("%d\n",SZ(b));

ll ans=0,pnt=0;

int k=SZ(a);

assert(SZ(a)==SZ(b));

rep(i,0,k) \_md[k-1-i]=-a[i];\_md[k]=1;

Md.clear();

rep(i,0,k) if (\_md[i]!=0) Md.push\_back(i);

rep(i,0,k) res[i]=base[i]=0;

res[0]=1;

while ((1ll<<pnt)<=n) pnt++;

for (int p=pnt;p>=0;p--) {

mul(res,res,k);

if ((n>>p)&1) {

for (int i=k-1;i>=0;i--) res[i+1]=res[i];res[0]=0;

rep(j,0,SZ(Md)) res[Md[j]]=(res[Md[j]]-res[k]\*\_md[Md[j]])%mod;

}

}

rep(i,0,k) ans=(ans+res[i]\*b[i])%mod;

if (ans<0) ans+=mod;

return ans;

}

VI BM(VI s) {

VI C(1,1),B(1,1);

int L=0,m=1,b=1;

rep(n,0,SZ(s)) {

ll d=0;

rep(i,0,L+1) d=(d+(ll)C[i]\*s[n-i])%mod;

if (d==0) ++m;

else if (2\*L<=n) {

VI T=C;

ll c=mod-d\*powmod(b,mod-2)%mod;

while (SZ(C)<SZ(B)+m) C.pb(0);

rep(i,0,SZ(B)) C[i+m]=(C[i+m]+c\*B[i])%mod;

L=n+1-L; B=T; b=d; m=1;

} else {

ll c=mod-d\*powmod(b,mod-2)%mod;

while (SZ(C)<SZ(B)+m) C.pb(0);

rep(i,0,SZ(B)) C[i+m]=(C[i+m]+c\*B[i])%mod;

++m;

}

}

return C;

}

int gao(VI a,ll n) {

VI c=BM(a);

c.erase(c.begin());

rep(i,0,SZ(c)) c[i]=(mod-c[i])%mod;

return solve(n,c,VI(a.begin(),a.begin()+SZ(c)));

}

};

int main() {

while (~scanf("%d",&n)) {

vector<int>v;

v.push\_back(1);

v.push\_back(2);

v.push\_back(4);

printf("%d\n",linear\_seq::gao(v,n-1));

}

}

### 多项式快速插值(n\*log^2)

#include<bits/stdc++.h>

using namespace std;

const int RLEN=1<<20|1;

inline char gc(){

static char ibuf[RLEN],\*ib,\*ob;

(ob==ib)&&(ob=(ib=ibuf)+fread(ibuf,1,RLEN,stdin));

return (ob==ib)?EOF:\*ib++;

}

#define gc getchar

inline int read(){

char ch=gc();

int res=0,f=1;

while(!isdigit(ch))f^=ch=='-',ch=gc();

while(isdigit(ch))res=(res+(res<<2)<<1)+(ch^48),ch=gc();

return f?res:-res;

}

#define ll long long

#define re register

#define pii pair<int,int>

#define fi first

#define se second

#define pb push\_back

#define cs const

const int mod=998244353,G=3;

inline int add(int a,int b){return a+b>=mod?a+b-mod:a+b;}

inline void Add(int &a,int b){a=add(a,b);}

inline int dec(int a,int b){return a>=b?a-b:a-b+mod;}

inline void Dec(int &a,int b){a=dec(a,b);}

inline int mul(int a,int b){return 1ll\*a\*b>=mod?1ll\*a\*b%mod:a\*b;}

inline void Mul(int &a,int b){a=mul(a,b);}

inline int ksm(int a,int b,int res=1){for(;b;b>>=1,a=mul(a,a))(b&1)?(res=mul(res,a)):0;return res;}

inline void chemx(int &a,int b){a<b?a=b:0;}

inline void chemn(int &a,int b){a>b?a=b:0;}

cs int N=(1<<17)+1,C=20;

#define poly vector<int>

#define bg begin

poly w[C+1];

int rev[N<<2];

inline void init\_rev(int lim){

for(int i=0;i<lim;i++)rev[i]=(rev[i>>1]>>1)|((i&1)\*(lim>>1));

}

inline void init\_w(){

for(int i=1;i<=C;i++)w[i].resize((1<<(i-1)));

int wn=ksm(G,(mod-1)/(1<<C));

w[C][0]=1;

for(int i=1;i<(1<<(C-1));i++)

w[C][i]=mul(w[C][i-1],wn);

for(int i=C-1;i;i--)

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

w[i][j]=w[i+1][j<<1];

}

inline void ntt(poly &f,int lim,int kd){

for(int i=0;i<lim;i++)if(i>rev[i])swap(f[i],f[rev[i]]);

for(int mid=1,l=1;mid<lim;mid<<=1,l++)

for(int i=0,a0,a1;i<lim;i+=(mid<<1))

for(int j=0;j<mid;j++){

a0=f[i+j],a1=mul(f[i+j+mid],w[l][j]);

f[i+j]=add(a0,a1),f[i+j+mid]=dec(a0,a1);

}

if(kd==-1){

reverse(f.begin()+1,f.begin()+lim);

for(int i=0,inv=ksm(lim,mod-2);i<lim;i++)Mul(f[i],inv);

}

}

inline int F(cs poly a,int x){

int p=1,res=0;

for(int i=0;i<a.size();i++,Mul(p,x))Add(res,mul(a[i],p));

return res;

}

inline poly operator +(cs poly &a,cs poly &b){

poly c(max(a.size(),b.size()),0);

for(int i=0;i<c.size();i++)c[i]=add(a[i],b[i]);

return c;

}

inline poly operator -(cs poly &a,cs poly &b){

poly c(max(a.size(),b.size()),0);

for(int i=0;i<c.size();i++)c[i]=dec(a[i],b[i]);

return c;

}

inline poly operator \*(poly a,poly b){

int deg=a.size()+b.size()-1,lim=1;

if(deg<=128){

poly c(deg,0);

for(int i=0;i<a.size();i++)

for(int j=0;j<b.size();j++)

Add(c[i+j],mul(a[i],b[j]));

return c;

}

while(lim<deg)lim<<=1;

init\_rev(lim);

a.resize(lim),ntt(a,lim,1);

b.resize(lim),ntt(b,lim,1);

for(int i=0;i<lim;i++)Mul(a[i],b[i]);

ntt(a,lim,-1),a.resize(deg);

return a;

}

inline poly Inv(poly a,int deg){

poly b,c(1,ksm(a[0],mod-2));

for(int lim=4;lim<(deg<<2);lim<<=1){

b=a,b.resize(lim>>1);

init\_rev(lim);

b.resize(lim),ntt(b,lim,1);

c.resize(lim),ntt(c,lim,1);

for(int i=0;i<lim;i++)Mul(c[i],dec(2,mul(b[i],c[i])));

ntt(c,lim,-1),c.resize(lim>>1);

}c.resize(deg);return c;

}

inline poly operator /(poly a,poly b){

int lim=1,deg=a.size()-b.size()+1;

reverse(a.bg(),a.end());

reverse(b.bg(),b.end());

while(lim<deg)lim<<=1;

b=Inv(b,lim),b.resize(deg);

a=a\*b,a.resize(deg);

reverse(a.bg(),a.end());

return a;

}

inline poly operator %(poly a,poly b){

poly c=a-(a/b)\*b;

c.resize(b.size()-1);

return c;

}

inline poly deriv(poly a){

for(int i=0;i<a.size()-1;i++)a[i]=mul(a[i+1],i+1);

a.pop\_back();return a;

}

#define lc (u<<1)

#define rc ((u<<1)|1)

#define mid ((l+r)>>1)

poly f[N<<2];

int n,x[N],y[N],g[N];

inline void build(int u,int l,int r){

if(l==r){f[u].pb(mod-x[l]),f[u].pb(1);return;}

build(lc,l,mid),build(rc,mid+1,r);

f[u]=f[lc]\*f[rc];

}

inline void calc(int u,int l,int r,poly res){

if(l==r){

g[l]=mul(ksm(F(res,x[l]),mod-2),y[l]);

return;

}

calc(lc,l,mid,res%f[lc]),calc(rc,mid+1,r,res%f[rc]);

}

inline poly getans(int u,int l,int r){

if(l==r)return poly(1,g[l]);

poly ansl=getans(lc,l,mid),ansr=getans(rc,mid+1,r);

return ansl\*f[rc]+ansr\*f[lc];

}

int main(){

n=read();

init\_w();

for(int i=1;i<=n;i++)x[i]=read(),y[i]=read();

build(1,1,n);

calc(1,1,n,deriv(f[1]));

poly ans=getans(1,1,n);

for(int i=0;i<n;i++)cout<<ans[i]<<" ";

}

### 多项式多点求值

#include <cstdio>

#include <algorithm>

#include <vector>

const int mod = 998244353, G = 3;

namespace Math {

inline int pw(int base, int p) {

static int res;

for (res = 1; p; p >>= 1, base = static\_cast<long long> (base) \* base % mod) if (p & 1) res = static\_cast<long long> (res) \* base % mod;

return res;

}

inline int inv(int x) { return pw(x, mod - 2); }

}

inline void reduce(int &x) { x += x >> 31 & mod; }

#define maxn 65536

int a[maxn], ans[maxn];

namespace Poly {

#define N maxn

int rev[N], lim, s, ilim;

int Wn[N + 1];

inline void clear(register int \*l, const int \*r) {

if (l >= r) return ;

while (l != r) \*l++ = 0;

}

inline void init(const int n) {

s = -1, lim = 1; while (lim < n) lim <<= 1, ++s; ilim = Math::inv(lim);

for (register int i = 0; i < lim; ++i) rev[i] = rev[i >> 1] >> 1 | (i & 1) << s;

const int t = Math::pw(G, (mod - 1) / lim);

\*Wn = 1; for (register int \*i = Wn; i != Wn + lim; ++i) \*(i + 1) = static\_cast<long long> (\*i) \* t % mod;

}

inline void NTT(int \*A, const int op = 1) {

static int Wt[N];

for (register int i = 1; i < lim; ++i) if (i < rev[i]) std::swap(A[i], A[rev[i]]);

for (register int mid = 1; mid < lim; mid <<= 1) {

const int t = lim / mid >> 1;

\*Wt = Wn[op ? 0 : lim];

for (register int \*i = Wt, W = 0; i != Wt + mid; ++i, W += t) \*i = Wn[op ? W : lim - W];

for (register int i = 0; i < lim; i += mid << 1) {

for (register int j = 0; j < mid; ++j) {

const int X = A[i + j], Y = static\_cast<long long> (Wt[j]) \* A[i + j + mid] % mod;

reduce(A[i + j] += Y - mod), reduce(A[i + j + mid] = X - Y);

}

}

}

if (!op) for (register int \*i = A; i != A + lim; ++i) \*i = static\_cast<long long> (\*i) \* ilim % mod;

}

std::vector<int> P[N << 1], S[N << 1];

int C[N], D[N];

void DC\_NTT(int rt, int l, int r) {

if (l == r) { P[rt] = {a[l], 1}; return ; }

int mid = l + r >> 1;

DC\_NTT(rt << 1, l, mid), DC\_NTT(rt << 1 | 1, mid + 1, r);

int L = rt << 1, R = rt << 1 | 1;

int n = P[L].size(), m = P[R].size();

init(n + m - 1);

std::copy(P[L].begin(), P[L].end(), C); clear(C + n, C + lim);

std::copy(P[R].begin(), P[R].end(), D); clear(D + m, D + lim);

NTT(C), NTT(D);

for (int i = 0; i < lim; ++i) C[i] = static\_cast<long long> (C[i]) \* D[i] % mod;

NTT(C, 0);

P[rt].assign(C, C + n + m - 1);

}

int E[N];

void INV(int \*A, int \*B, int n) {

if (n == 1) {

\*B = Math::inv(\*A);

return ;

}

INV(A, B, n + 1 >> 1);

init(n + n - 1);

std::copy(A, A + n, E); clear(E + n, E + lim);

clear(B + (n + 1 >> 1), B + lim);

NTT(B), NTT(E);

for (int i = 0; i < lim; ++i) B[i] = (2 + mod - static\_cast<long long> (B[i]) \* E[i] % mod) \* B[i] % mod;

NTT(B, 0); clear(B + n, B + lim);

}

int F[N];

void DIV(int A, int n, int B, int m) {

const int len = n - m + 1;

init(len << 1);

std::reverse\_copy(S[A].begin(), S[A].end(), C); clear(C + len, C + lim);

std::reverse\_copy(P[B].begin(), P[B].end(), D); clear(D + len, D + lim);

clear(F, F + lim);

INV(D, F, len);

NTT(C), NTT(F);

for (int i = 0; i < lim; ++i) F[i] = static\_cast<long long> (F[i]) \* C[i] % mod;

NTT(F, 0);

clear(F + len, F + lim);

}

void \_\_DIVMOD(int res, int A, int n, int B, int m) {

if (n < m) {

S[res].assign(S[A].begin(), S[A].end());

return ;

}

DIV(A, n, B, m);

init(n);

std::reverse\_copy(F, F + n - m + 1, C); clear(C + n - m + 1, C + lim);

std::copy(P[B].begin(), P[B].end(), D); clear(D + m, D + lim);

NTT(C), NTT(D);

for (int i = 0; i < lim; ++i) C[i] = static\_cast<long long> (C[i]) \* D[i] % mod;

NTT(C, 0);

for (int i = 0; i < m - 1; ++i) reduce(C[i] = S[A][i] - C[i]);

S[res].assign(C, C + m - 1);

}

void DIVMOD(int res, int A) {

int n = S[A].size(), m = P[res].size();

\_\_DIVMOD(res, A, n, res, m);

}

void solve(int rt, int l, int r) {

if (l == r) {

ans[l] = S[rt][0];

return ;

}

int mid = l + r >> 1;

DIVMOD(rt << 1, rt), DIVMOD(rt << 1 | 1, rt);

solve(rt << 1, l, mid), solve(rt << 1 | 1, mid + 1, r);

}

void work(int \*f, int n, int m) {

DC\_NTT(1, 1, m);

S[0].assign(f, f + n);

DIVMOD(1, 0);

solve(1, 1, m);

}

#undef N

}

int n, m;

int f[maxn];

int main() {

scanf("%d%d", &n, &m); if (!m) return 0; ++n;

for (int i = 0; i < n; ++i) scanf("%d", f + i);

for (int i = 1; i <= m; ++i) scanf("%d", a + i), reduce(a[i] = -a[i]);

Poly::work(f, n, m);

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

return 0;

}

# 字符串

### AC自动机

#include <stdio.h>

#include <iostream>

#include <string.h>

#define ll long long

using namespace std;

const int maxn=1000007;

int you[maxn],q[maxn],tr[maxn][30],fail[maxn],rt=1,e1=1;

char s[maxn];

void ins(char \*s){

int u=rt,len=strlen(s+1);

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

if(!tr[u][s[i]-'a']) tr[u][s[i]-'a']=++e1;

u=tr[u][s[i]-'a'];

}

you[u]++;

}

void getfail(){

int ql=1,qr=0;q[++qr]=rt;fail[1]=0;

while(ql<=qr){

int u=q[ql++];

for(int i=0;i<26;i++){

if(!tr[u][i]) continue;

int v=fail[u];

while(v&&!tr[v][i]) v=fail[v];

fail[tr[u][i]]=tr[v][i];

q[++qr]=tr[u][i];

}

}

}

void acm(char \*s){

int len=strlen(s+1),u=rt,ans=0;

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

while(u&&!tr[u][s[i]-'a']) u=fail[u];

if(tr[u][s[i]-'a']) u=tr[u][s[i]-'a'];

for(int k=u;~you[k]&&k!=rt;k=fail[k]) ans+=you[k],you[k]=-1;

}

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

}

void init(){

for(int i=1;i<=e1;i++) for(int k=0;k<26;k++) tr[i][k]=0;

for(int i=1;i<=e1;i++) you[i]=0;

e1=1;

}

int main(){

int t;cin>>t;

for(int i=0;i<26;i++) tr[0][i]=1;

while(t--){

init();

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

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

scanf("%s",s+1);ins(s);

}

getfail();

scanf("%s",s+1);

acm(s);

}

return 0;

}

### 最小表示法

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

#include <math.h>

#include <vector>

using namespace std;

typedef long long ll;

int getmin(char \*s){

int ql=0,qr=1,pos=0,len=strlen(s),x;

while(ql<len&&qr<len&&pos<len){

x=s[(ql+pos)%len]-s[(qr+pos)%len];

if(!x) pos++;

else{

if(x>0) ql+=pos+1;

else qr+=pos+1;

if(ql==qr) qr++;

pos=0;

}

}

return min(ql,qr);

}

int getmax(char \*s){

int ql=0,qr=1,pos=0,len=strlen(s),x;

while(ql<len&&qr<len&&pos<len){

x=s[(ql+pos)%len]-s[(qr+pos)%len];

if(!x) pos++;

else{

if(x<0) ql+=pos+1;

else qr+=pos+1;

if(ql==qr) qr++;

pos=0;

}

}

return min(ql,qr);

}

int main(){

return 0;

}

### KMP

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

#include <math.h>

#include <vector>

#include <map>

using namespace std;

typedef long long ll;

const int maxn=100007;

int fail[maxn];

char s1[maxn],s2[maxn];

int main(){

scanf("%s%s",s1+1,s2+1);

int last=0,l1=strlen(s1+1),l2=strlen(s2+1);

for(int i=2;i<=l1;i++){

while(last&&s1[last+1]!=s1[i]) last=fail[last];

if(s1[last+1]==s1[i]) last++;

fail[i]=last;

}

last=0;int ans=0;

for(int i=1;i<=l2;i++){

while(last&&s1[last+1]!=s2[i]) last=fail[last];

if(s1[last+1]==s2[i]) last++;

if(last==l1){

last=fail[last];ans++;

}

}

printf("%d",ans?ans:-1);

return 0;

}

### Manacher

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

#include <math.h>

#include <vector>

#include <map>

using namespace std;

typedef long long ll;

const int maxn=110007;

int fail[maxn<<1];

char s[maxn<<1],sf[maxn<<1];

int main(){

while(scanf("%s",s+1)==1){

int len=strlen(s+1);

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

sf[i\*2-1]='#';

sf[i\*2]=s[i];

}

sf[0]='@';sf[len\*2+1]='#';sf[len\*2+2]='$';

int mx=0,ans=0,id=0;

for(int i=1;i<=2\*len+1;i++){

fail[i]=mx>i?min(mx-i,fail[2\*id-i]):1;

while(sf[i-fail[i]]==sf[i+fail[i]]) fail[i]++;

if(fail[i]+i>mx) mx=fail[i]+i,id=i;

ans=max(ans,fail[i]-1);

}

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

}

return 0;

}

### 回文自动机

#include <bits/stdc++.h>

#define mp make\_pair

#define pii pair<int,int>

using namespace std;

typedef long long ll;

#define rint register int

const int maxn=300007;

const int inf=(1LL<<29);

int read(){

int x=0;int f=1;

char c=getchar();

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

if(c=='-') f=-1;c=getchar();

}

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

x\*=f;return x;

}

int tr[maxn][27];

int fail[maxn],L[maxn],fa[maxn],e1,las,LL[maxn],sz[maxn],num[maxn];

void init(){

L[0]=0;fail[0]=1;L[1]=-1;fail[1]=0;e1=1;las=0;

memset(tr[0],0,sizeof(tr[0]));

memset(tr[1],0,sizeof(tr[1]));

}

char s[maxn];

int new\_node(int x){

++e1;

memset(tr[e1],0,sizeof(tr[e1]));

L[e1]=x;

return e1;

}

int val[maxn];

void ins(int c,int n){

int u=las;

while(s[n-L[u]-1]!=s[n]) u=fail[u];

if(!tr[u][c]){

int now=new\_node(L[u]+2);

int v=fail[u];

while(s[n-L[v]-1]!=s[n]) v=fail[v];

fail[now]=tr[v][c];

tr[u][c]=now;

val[now]=val[u]|(1<<c);

}

las=tr[u][c];sz[las]++;

LL[n]=L[las];

}

void count(){

for(int i=e1;i>1;i--) sz[fail[i]]+=sz[i];

}

int main(){

//cin.tie(0);ios\_base::sync\_with\_stdio(false);

cin>>s+1;

int n=strlen(s+1);

init();

for(int i=1;i<=n;i++) ins(s[i]-'a',i);

count();

ll ans=0;

for(int i=2;i<=e1;i++){

int x=0;

for(int k=0;k<26;k++) if(val[i]&(1<<k)) x++;

ans+=sz[i]\*x;

}

cout<<ans;

return 0;

}

### 后缀自动机(广义)

#include <bits/stdc++.h>

#define mp make\_pair

#define sqr(x) (x)\*(x)

using namespace std;

typedef pair<int,int> pii;

typedef long long ll;

const int maxn=250007;

const int inf=1<<29;

int read(){

int x=0,f=1;

char ch=getchar();

while(ch<'0'||ch>'9') {if(ch=='-') f=-1;ch=getchar();}

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

return x\*f;

}

int Len;char s[maxn];

struct SAM{

int e1,last;

int tr[maxn<<1][30];

int you[maxn<<1];

int len[maxn<<1],fa[maxn<<1],mn[maxn<<1];

int newnode(){

++e1;memset(tr[e1],0,sizeof(tr[e1]));

return e1;

}

inline void init(){

e1=0;last=newnode();

}

void add(int x){

if(tr[last][x]){

int p=last,q=tr[p][x];

if(len[q]==len[p]+1) last=q;

else{

int np=newnode();

len[np]=len[p]+1;

memcpy(tr[np],tr[q],sizeof(tr[q]));

fa[np]=fa[q];

for(;p&&tr[p][x]==q;p=fa[p]) tr[p][x]=np;

fa[q]=np;last=np;

}

return;

}

int now=newnode(),p;

len[now]=len[last]+1;

for(p=last;p&&!tr[p][x];p=fa[p]) tr[p][x]=now;

if(!p) fa[now]=1;

else{

int q=tr[p][x];

if(len[q]==len[p]+1) fa[now]=q;

else{

int np=newnode();

len[np]=len[p]+1;

memcpy(tr[np],tr[q],sizeof(tr[q]));

fa[np]=fa[q];

for(;p&&tr[p][x]==q;p=fa[p]) tr[p][x]=np;

fa[q]=fa[now]=np;

}

}

last=now;

}

int num[maxn],id[maxn<<1];

void topsort(){

for(int i=0;i<=Len;i++) num[i]=0;

for(int i=1;i<=Len;i++) num[i]+=num[i-1];

for(int i=1;i<=e1;i++) id[num[len[i]]--]=i;

for(int i=e1;i;i--){

you[fa[id[i]]]+=you[id[i]];

}

}

}S;

int main(){

scanf("%s",s+1);

Len=strlen(s+1);S.init();

for(int i=1;i<=Len;i++) S.add(s[i]-'a');

return 0;

}

### 后缀数组

#include <stdio.h>

#include <string.h>

#include <iostream>

#include <algorithm>

using namespace std;

const int maxn=200007;

char s[maxn];

int sa[2][maxn],rk[2][maxn],now,bit,n,a[maxn],h[maxn],v[maxn],l1;

void solve(int sa[],int rk[],int SA[],int RK[]){

for(int i=1;i<=n;i++) v[rk[sa[i]]]=i;

for(int i=n;i;i--) if(sa[i]>l1) SA[v[rk[sa[i]-l1]]--]=sa[i]-l1;

for(int i=n-l1+1;i<=n;i++) SA[v[rk[i]]--]=i;

for(int i=1;i<=n;i++) RK[SA[i]]=RK[SA[i-1]]+(rk[SA[i]]!=rk[SA[i-1]]||rk[SA[i]+l1]!=rk[SA[i-1]+l1]);

}

void getsa(){

now=0;

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

for(int i=1;i<=30;i++) v[i]+=v[i-1];

for(int i=1;i<=n;i++) sa[now][v[a[i]]--]=i;

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

for(l1=1;l1<n;l1<<=1){

solve(sa[now],rk[now],sa[now^1],rk[now^1]);now^=1;

}

}

void geth(){

l1=0;

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

if(rk[now][i]==1) h[rk[now][i]]=0;

else{

int last=sa[now][rk[now][i]-1];

while(l1<=min(n-i,n-last)&&a[i+l1]==a[last+l1]) l1++;

h[rk[now][i]]=l1;if(l1>0) l1--;

}

}

int main(){

scanf("%s",s+1);n=strlen(s+1);

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

getsa();geth();

for(int i=1;i<=n;i++)printf("%d ",sa[now][i]);

printf("\n");

for(int i=1;i<=n;i++)printf("%d ",h[i]);

return 0;

}

# 计算几何

### 球体积交/并

#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

const double pi = acos(-1);

const int maxn=100007;

const int inf = 1e9 + 7;

typedef struct {

double x, y, z, r;

}point;

int n;

point a[maxn],s;

double dis(point p,point q) {

double ans = sqrt((p.x-q.x)\*(p.x-q.x)+(p.y-q.y)\*(p.y-q.y)+(p.z-q.z)\*(p.z-q.z));

return ans;

}

int main()

{

int T;scanf("%d", &T);

int Case = 1;

while (T--)

{

scanf("%d", &n);

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

scanf("%lf%lf%lf%lf",&a[i].x,&a[i].y,&a[i].z,&a[i].r);

}

scanf("%lf%lf%lf%lf",&s.x,&s.y,&s.z,&s.r);

double ans=0;

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

double d = dis(s,a[i]);

if(d>=s.r+a[i].r) continue;

else if(d+a[i].r<=s.r) ans+=(4.0/3)\*pi\*a[i].r\*a[i].r\*a[i].r;

else if(d+s.r<=a[i].r) ans+=(4.0/3)\*pi\*s.r\*s.r\*s.r;

else{

double co=(s.r\*s.r+d\*d-a[i].r\*a[i].r)/(2.0\*d\*s.r);

double h=s.r\*(1-co);

ans+=(1.0/3)\*pi\*(3.0\*s.r-h)\*h\*h;

co=(a[i].r\*a[i].r+d\*d-s.r\*s.r)/(2.0\*d\*a[i].r);

h=a[i].r\*(1-co);

ans+=(1.0/3)\*pi\*(3.0\*a[i].r-h)\*h\*h;

}

}

ans=-ans;

ans+=pi\*(4.0/3)\*(a[0].r\*a[0].r\*a[0].r+s.r\*s.r\*s.r);

printf("%.10lf\n",ans);

}

return 0;

}

### 计算几何板子杂烩

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <math.h>

#include <algorithm>

#define sqr(x) (x)\*(x)

using namespace std;

typedef long long ll;

const int maxn=100007;

const double eps=1e-7;

#define ppb pair<pair<point,point>,bool>

#define pp pair<point,point>

struct point{

double x,y;

point(double x=0.0,double y=0.0):x(x),y(y){

}

double operator ^ (point a){

return x\*a.y-y\*a.x;

}

double operator \* (point a){

return x\*a.x+y\*a.y;

}

inline point operator - (point a){

return point(x-a.x,y-a.y);

}

inline point operator + (point a){

return point(x+a.x,y+a.y);

}

inline point operator \* (double a){

return point(x\*a,y\*a);

}

inline bool operator == (point a){

return fabs(x-a.x)<eps&&fabs(y-a.y)<eps;

}

inline bool operator < (point a){

if(fabs(y-a.y)<eps) return x<a.x;

return y<a.y;

}

void print(){

cout<<x<<" "<<y<<" ";

}

}p[maxn],s;

bool equal0(double x){

return fabs(x)<eps;

}

bool equal\_num(double x,double y){

return fabs(x-y)<eps;

}

typedef point vec;

inline double length(point a){

return sqrt(sqr(a.x)+sqr(a.y));

}

inline bool parallel(point a,point b,point c,point d){

return equal0((b-a)^(d-c));

}

inline double dis(point a,point b){

return length(a-b);

}

inline double dot(point a,point b,point c){

return (a-c)\*(b-c);

}

inline double mul(point a,point b,point c){

return (a-c)^(b-c);

}

inline bool cmp(point a,point b){

if(fabs(mul(a,b,s))<eps) return dis(a,s)<dis(b,s);

else return mul(a,b,s)>0;

}

inline int sign(double x){

if(fabs(x)<eps) return 0;

else if(x>0) return 1;

else return -1;

}

int top;point q[maxn];

inline void sort\_by\_angle1(point \*p,int n){

int pos=1;

for(int i=2;i<=n;i++){

if(p[i].y+eps<p[pos].y||fabs(p[i].y-p[pos].y)<eps&&p[i].x<p[pos].x) pos=i;

}

swap(p[1],p[pos]);s=p[1];

sort(p+2,p+n+1,cmp);

}

inline void graham(point \*p,int n){

sort\_by\_angle1(p,n);

top=0;

if(n==1){

q[++top]=p[1];return;

}

else if(n==2){

q[++top]=p[1];q[++top]=p[2];return;

}

q[++top]=p[1];q[++top]=p[2];

for(int i=3;i<=n;i++){

while(top>=2&&mul(p[i],q[top],q[top-1])>-eps) top--;

q[++top]=p[i];

}

}

inline double tri\_s(point a,point b,point c){

return fabs((a-b)^(c-b))/2;

}

void sort\_by\_angle2(point \*p,int n,point a){

s=a;

sort(p+1,p+n+1,cmp);

}

//intersection of segment and line

bool intersect\_line\_segment(point a,point b,point c,point d){

double x=mul(b,c,a)\*mul(b,d,a);

return x<eps;

}

double dis\_point\_to\_line(point a,point b,point c){

double x=mul(b,a,c);

return fabs(x)/length(b-c);

}

bool online(point a,point b,point c){

return equal0(mul(c,b,a));

}

bool onsegment(point a,point b,point c){

return online(a,b,c)&&dot(b,c,a)<eps;

}

bool onsegment\_spec(point a,point b,point c){

return online(a,b,c)&&dot(b,c,a)<-eps;

}

bool intersect\_segment\_spec(point a,point b,point c,point d){

double x1=mul(b,c,a),x2=mul(b,d,a);

double x3=mul(d,a,c),x4=mul(d,b,c);

if(equal0(x1)&&equal0(x2)) return onsegment\_spec(a,c,d)||onsegment\_spec(b,c,d);

else if(sign(x1\*x2)<0&&sign(x3\*x4)<0) return true;

else return false;

}

bool intersect\_segment(point a,point b,point c,point d){

double x1=mul(b,c,a),x2=mul(b,d,a);

double x3=mul(d,a,c),x4=mul(d,b,c);

if(equal0(x1)&&equal0(x2)) return onsegment(a,c,d)||onsegment(b,c,d);

else if(sign(x1\*x2)>0||sign(x3\*x4)>0) return false;

else return true;

}

point get\_intersect\_point(point p1,point p2,point p3,point p4){

return p1+(p2-p1)\*(((p1-p3)^(p4-p3))/((p4-p3)^(p2-p1)));

}

//probably some eps problem===================================

inline ppb get\_intersect\_segment(point a,point b,point c,point d){

if(onsegment(a,c,d)||onsegment(b,c,d)){

if(b<a) swap(a,b);

if(d<c) swap(c,d);

point x,y;

if(onsegment(c,a,b)) x=c;else x=a;

if(onsegment(d,a,b)) y=d;else y=b;

return make\_pair(make\_pair(x,y),true);

}

else return make\_pair(make\_pair(0,0),false);

}

inline point project\_point(point p,point a,point b){

point ans;

double u=((b.x-a.x)\*(b.x-a.x)+(b.y-a.y)\*(b.y-a.y));

u = ((b.x-a.x)\*(b.x-p.x)+(b.y-a.y)\*(b.y-p.y))/u;

ans.x=u\*a.x+(1-u)\*b.x;

ans.y=u\*a.y+(1-u)\*b.y;

return ans;

}

inline ppb project\_segment(point a,point b,point c,point d){

point x=project\_point(c,a,b),y=project\_point(d,a,b);

return get\_intersect\_segment(a,b,x,y);

}

inline pp project\_line(point a,point b,point c,point d){

point x=project\_point(c,a,b),y=project\_point(d,a,b);

return make\_pair(x,y);

}

//==========================================================

struct segment{

point a,b;double ang;

double angle(){

return atan2(b.y-a.y,b.x-a.x);

}

segment(point a=point(),point b=point()):a(a),b(b){

ang=angle();

}

};

struct polygon{

segment s[21];

int num;

};

struct circle{

point o;

double r;

};

int circle\_to\_line(circle a,point b,point c){

double x=dis\_point\_to\_line(a.o,b,c);

if(fabs(x-a.r)<eps) return 0;

else if(x<a.r) return 1;

else return -1;

}

int read(){

int x=0,f=1;

char ch=getchar();

while(ch<'0'||ch>'9') {if(ch=='-') f=-1;ch=getchar();}

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

return x\*f;

}

bool cmp\_seg(segment a,segment b){

return fabs(a.ang-b.ang)<eps?((a.b-a.a)^(b.b-a.a))<eps:a.ang<b.ang;

}

bool point\_in\_poly(point a,point \*p,int n){

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

if(mul(p[i],p[i+1],a)<-eps) return false;

}

return mul(p[n],p[1],a)>-eps;

}

inline void clockwise(point \*p,int n){

double ans=0;

for(int i=2;i<=n;i++) ans+=mul(p[i-1],p[i],p[1]);

if(ans<0) reverse(p+1,p+n+1);

}

double poly\_area(point \*p,int n){

double ans=0;

for(int i=1;i<=n;i++) ans+=p[i]^p[i%n+1];

return fabs(0.5\*ans);

}

segment qs[maxn];point qp[maxn];

inline bool SI(segment \*s,int n,point \*res,int &m){

sort(s+1,s+n+1,cmp\_seg);

int ql=1,qr=0;

qs[++qr]=s[1];

for(int i=2;i<=n;i++){

if(fabs(s[i].ang-s[i-1].ang)>eps){

while(ql<qr&&mul(s[i].b,qp[qr-1],s[i].a)<-eps) --qr;

while(ql<qr&&mul(s[i].b,qp[ql],s[i].a)<-eps) ++ql;

qp[qr]=get\_intersect\_point(qs[qr].a,qs[qr].b,s[i].a,s[i].b);

qs[++qr]=s[i];

if(parallel(qs[qr-1].a,qs[qr-1].b,qs[qr].a,qs[qr].b))

return false;

}

}

while(ql<qr&&mul(qs[ql].b,qp[qr-1],qs[ql].a)<-eps) --qr;

while(ql<qr&&mul(qs[qr].b,qp[ql],qs[qr].a)<-eps) ++ql;

if(qr<=ql+1) return false;

qp[qr]=get\_intersect\_point(qs[ql].a,qs[ql].b,qs[qr].a,qs[qr].b);

m=0;for(int i=ql;i<=qr;i++) res[++m]=qp[i];

return true;

}

segment seg[maxn];

point res[maxn];int m;

int main(){

return 0;

}

# 其他

### double数组的memset

极大值memset(a,0x7f,sizeof(a));

次大值memset(a,0xfe,sizeof(a));

极小值memset(a,0x42,sizeof(a));

次小值memset(a,0x32,sizeof(a));

### 快速乘

ll fm(ll a,ll b) {

return (a\*b - (ll)((long double)a/mod\*b)\*mod+mod)%mod;

}

### FAST IO

int read(){

int x=0;int f=1;

char c=getchar();

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

if(c=='-') f=-1;c=getchar();

}

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

x\*=f;return x;

}

char q[67];int top;

void write(int x){

if(x==0){

putchar('0');

}

else{

if(x<0) putchar('-');

x=x<0?-x:x;

while(x){

q[++top]=x%10+'0';x/=10;

}

while(top) putchar(q[top--]);

}

}

#define rd(n) FastIO::read(n)

namespace FastIO {

const int SIZE = 1 << 16;

char buf[SIZE], obuf[SIZE], str[60];

int bi = SIZE, bn = SIZE, opt;

int read(char \*s) {

while (bn) {

for (; bi < bn && buf[bi] <= ' '; bi++);

if (bi < bn) break;

bn = fread(buf, 1, SIZE, stdin);

bi = 0;

}

int sn = 0;

while (bn) {

for (; bi < bn && buf[bi] > ' '; bi++) s[sn++] = buf[bi];

if (bi < bn) break;

bn = fread(buf, 1, SIZE, stdin);

bi = 0;

}

s[sn] = 0;

return sn;

}

bool read(int& x) {

int n = read(str), bf;

if (!n) return 0;

int i = 0; if (str[i] == '-') bf = -1, i++; else bf = 1;

for (x = 0; i < n; i++) x = x \* 10 + str[i] - '0';

if (bf < 0) x = -x;

return 1;

}

};

### O3优化

#pragma GCC optimize(3)

### FFT常用模数

r^2\*k+1 r k g

3 1 1 2

5 1 2 2

17 1 4 3

97 3 5 5

193 3 6 5

257 1 8 3

7681 15 9 17

12289 3 12 11

40961 5 13 3

65537 1 16 3

786433 3 18 10

5767169 11 19 3

7340033 7 20 3

23068673 11 21 3

104857601 25 22 3

167772161 5 25 3

469762049 7 26 3

998244353 119 23 3

1004535809 479 21 3

2013265921 15 27 31

2281701377 17 27 3

3221225473 3 30 5

75161927681 35 31 3

77309411329 9 33 7

206158430209 3 36 22

2061584302081 15 37 7

2748779069441 5 39 3

6597069766657 3 41 5

39582418599937 9 42 5

79164837199873 9 43 5

263882790666241 15 44 7

1231453023109121 35 45 3

1337006139375617 19 46 3

3799912185593857 27 47 5

4222124650659841 15 48 19

7881299347898369 7 50 6

31525197391593473 7 52 3

180143985094819841 5 55 6

1945555039024054273 27 56 5

4179340454199820289 29 57 3

### 模拟退火

#include <stdio.h>

#include <iostream>

#include <string.h>

#include <algorithm>

#include <math.h>

#include <vector>

#include <map>

#define sqr(x) (x)\*(x)

using namespace std;

typedef long long ll;

const double eps=1e-15;

const int maxn=10007;

double ansx,ansy;

int n;

double xx[maxn],yy[maxn],w[maxn];

double calc(double x,double y){

double tot=0;

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

tot+=sqrt(sqr(xx[i]-x)+sqr(yy[i]-y))\*w[i];

}

return tot;

}

void mnth(){

double T=200;

while(T>eps){

double nowx=(ansx+(rand()\*2-RAND\_MAX)\*T);

double nowy=(ansy+(rand()\*2-RAND\_MAX)\*T);

double delta=calc(nowx,nowy)-calc(ansx,ansy);

if(delta<0) ansx=nowx,ansy=nowy;

else if(exp(-delta/T)\*RAND\_MAX>rand()) ansx=nowx,ansy=nowy;

T\*=0.999;

}

}

int main(){

scanf("%d",&n);

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

scanf("%lf%lf%lf",&xx[i],&yy[i],&w[i]);

ansx+=xx[i];ansy+=yy[i];

}

ansx/=n;ansy/=n;

mnth();

printf("%.3lf %.3lf",ansx,ansy);

return 0;

}

# Python

### Exgcd板子

m=[0]

r=[0]

x=0

y=0

def exgcd(a,b) :

global x

global y

gcd=0

if b==0:

x=1

y=0

return a

else:

gcd=exgcd(b,a%b)

x,y=y,x

y-=x\*(a//b)

return gcd

def powm(a,b,mod):

ans=0;

while(b):

if b%2:

ans=(ans+a)%mod

a=(a\*2)%mod

b=b//2

return (ans+mod)%mod

def exchina(n):

global x

global y

tot=m[1]

ans=r[1];

i=2

while i<=n :

a=tot

b=m[i]

c=(r[i]-ans%m[i]+m[i])%m[i]

gcd=exgcd(a,b)

if c%gcd!=0:

return -1

x=powm(x,c//gcd,b//gcd)

ans+=x\*tot

tot\*=(b//gcd)

ans=(ans%tot+tot)%tot

i=i+1

return ans

n,mm=input().split(" ")

n=int(n)

mm=int(mm)

i=1

while i<=n :

xx,yy=input().split(" ")

xx=int(xx)

yy=int(yy)

m.append(xx)

r.append(yy)

i+=1

ans=exchina(n)

if ans==-1:

print("he was definitely lying")

elif ans<=mm :

print(ans);

else :

print("he was probably lying");

# JAVA

### 大整数

import java.math.BigInteger;

import java.util.Scanner;

public class gcd{

static BigInteger zero=new BigInteger("0");

static BigInteger one=new BigInteger("1");

static int num[]={0,2,3,5,7,11,13,17,19,23};

static BigInteger gcd(BigInteger a,BigInteger b){

return b.compareTo(zero)==0?a:gcd(b,a.remainder(b));

}

public static void main(String []args){

Scanner in=new Scanner(System.in);

BigInteger a;

int n=in.nextInt();

while(n>0){

n--;

a=in.nextBigInteger();

if(a.compareTo(one)==0){

System.out.println("1/1");

continue;

}

BigInteger now=new BigInteger("1"),up=new BigInteger("1");

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

BigInteger t=new BigInteger(num[i]+"");

if(!(now.multiply(t).compareTo(a)>0)){

now=now.multiply(t);

up=up.multiply(t.add(one));

}

else break;

}

BigInteger x=gcd(now,up);

now=now.divide(x);up=up.divide(x);

System.out.print(now);

System.out.print('/');

System.out.println(up);

}

}

}

# 公式及杂项

### 二分图

n-最大匹配=最小边覆盖=最小不相交链覆盖

2\*(n-最大匹配)=最大独立集

最大匹配=最小顶点覆盖

最大团=补图最大独立集

最小相交链覆盖先用floyd求传递闭包

输出最大独立集方案：从s开始遍历残量网络，S集合中的可到达点和T集合中的不可到达点是最大独立集

输出最小点覆盖方案：从s开始遍历残量网络，S集合中的不可到达点和T集合中的可到达点是最大独立集