ECNU ICPC

Team Reference Document

F0RE1GNERS

March 2019

目录

1 All

1.1 dsu_on_tree

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
const int N=2e5+5;
vector<int>e[N];
int n,tot;
int a[N],cnt[N],sz[N],son[N],idx[N],r[N],rth[N],num[N],ans[N];
multiset<int>s;
void dfs1(int x,int fa){
    sz[x]=1;
    idx[x]=++tot;
    rth[tot]=x;
    for(auto u:e[x]){
        if(u==fa)continue;
        dfs1(u,x);
        if(!son[x]||sz[son[x]]<sz[u])son[x]=u;
        sz[x]+=sz[u];
    }
    r[x]=tot;
}
void dfs2(int x,int fa,int f){
    for(auto u:e[x]){
        if(u==son[x]||u==fa)continue;
        dfs2(u,x,0);
    }
    if(son[x])dfs2(son[x],x,1);
    for(auto u:e[x]){
        if(u==son[x]||u==fa)continue;
        for(int i=idx[u];i<=r[u];i++){
            s.erase(cnt[a[rth[i]]]);
            num[cnt[a[rth[i]]]]-=a[rth[i]];
            cnt[a[rth[i]]]++;
            s.insert(cnt[a[rth[i]]]);
            num[cnt[a[rth[i]]]]+=a[rth[i]];
        }
    }
    s.erase(cnt[a[x]]);
    num[cnt[a[x]]]-=a[x];
    cnt[a[x]]++;
    s.insert(cnt[a[x]]);
    num[cnt[a[x]]]+=a[x];
    ans[x]=num[*s.rbegin()];
    if(!f){
        for(int i=idx[x];i<=r[x];i++){
            s.erase(cnt[a[rth[i]]]);
            num[cnt[a[rth[i]]]]-=a[rth[i]];
            cnt[a[rth[i]]]--;
            s.insert(cnt[a[rth[i]]]);
            num[cnt[a[rth[i]]]]+=a[rth[i]];
        }
    }
}
void solve(){
    cin>>n;
    for(int i=1;i<=n;i++)cin>>a[i];
    for(int i=1,u,v;i<n;i++)cin>>u>>v,e[u].push_back(v),e[v].push_back(u);
    dfs1(1,0);
    dfs2(1,0,0);
    for(int i=1;i<=n;i++)cout<<ans[i]<<" ";
}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int t=1;
    //cin>>t;
    while(t--)solve();
}
```

1.2 Lucas

```cpp
#include<bits/stdc++.h>

using namespace std;
#define int long long
const int N=2e5+5;
const int MOD=1e6+3;
const int logn=21;
const double PI=3.14159265358979323846264338327795;
int ksm(int x,int b){
  int ans=1;
  while(b){
    if(b&1)ans=ans*x%MOD;
    x=x*x%MOD;
    b>>=1;
  }
  return ans;
}
int exgcd(int a,int b,int& x,int& y){
  if(!b){x=1,y=0;return a;}
  int ret=exgcd(b,a%b,y,x);
  y-=a/b*x;
  return ret;
}
int Inv(int a,int p){
  int d,x,y;
  d=exgcd(a,p,x,y);
  if(d==1)return (x%p+p)%p;
  return -1;
}
int Cm(int n,int m,int p){
  int a=1,b=1;
  if(m>n)return 0;
  while(m){
    a=(a*n)%p;
    b=(b*m)%p;
    m--;
    n--;
  }
  return a*Inv(b,p)%p;
}
int Lucas(int n,int m,int p){
  if(m==0)return 1;
  return Cm(n%p,m%p,p)*Lucas(n/p,m/p,p)%p;
}
void solve(){
  int n,l,r,sum;
  cin>>n>>l>>r;
  int len=r-l+1;
  //cout<<len<<"\n";
  cout<<(Lucas(len+n,len,MOD)-1+MOD)%MOD<<"\n";
}
signed main(){
  ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
  int t=1;
  cin>>t;
  while(t--)
  solve();
}
```

1.3 Dinic

```cpp
#include<iostream>
#include<vector>
#include<string.h>
#include<algorithm>
#include<map>
#include<queue>
#include<stack>
#include<cmath>
#include<set>
#include<unordered_map>
#include<deque>
#include<iomanip>
#include<bitset>
#define ll long long
#define int long long
#define pii pair<int,int>
using namespace std;
const int N=2e3+5;
const int M=1e5+5;
const int INF=2e9;
//const int p=998244353;
const int MOD=1e9+7;
const double DARW=0.97;
int tot,m,n,s,t;
struct edge{
  int v,w,nxt;
}e[N<<2];
int head[N],a[N],b[N],vis[N],d[N];
set<int>ans;
void add(int u,int v,int w){
  e[tot].v=v;
  e[tot].w=w;
  e[tot].nxt=head[u];
  head[u]=tot++;
}
int bfs(){
  ans.clear();
  memset(vis,0,sizeof(vis));
  queue<int>q;
  q.push(s);
  vis[s]=1;
  d[s]=0;
  while(!q.empty()){
    int x=q.front();
    q.pop();
    for(int i=head[x];i!=-1;i=e[i].nxt){
      if(vis[e[i].v]||e[i].w==0)continue;
      ans.insert(e[i].v);
      d[e[i].v]=d[x]+1;
      vis[e[i].v]=1;
      q.push(e[i].v);
    }
  }
  //cout<<d[t]<<"\n";
  //cout<<t<<"\n";
  return vis[t];
}
int dfs(int x,int a){
  if(x==t||a==0)return a;
  int flow=0,f;
  for(int i=head[x];i!=-1;i=e[i].nxt){
    if(d[x]+1==d[e[i].v]&&(f=dfs(e[i].v,min(a,e[i].w)))>0){
      e[i].w-=f;
      e[i^1].w+=f;
      a-=f;
      flow+=f;
      if(a==0)break;
    }
  }
  return flow;
}
int Dinic(){
  int flow=0;
  while(bfs()){
```

```
    flow+=dfs(s,INF);
  }
  return flow;
}
void solve(){
  cin>>n>>n;
  int as=0;s=0,t=n+m+1;
  memset(head,-1,sizeof(head));
  string str;
  stringstream ss;
  for(int i=1;i<=n;i++){
    cin>>a[i];
    as+=a[i];
    add(0,i,a[i]);
    add(i,0,0);
    ss.clear();
    getline(cin,str);
    ss<<str;
    int x;
    while(ss>>x){
      //cout<<x<<"\n";
      add(i,x+m,INF);
      add(x+m,i,0);
    }
  }
  for(int i=1;i<=n;i++){
    cin>>b[i];
    add(i+m,n+m+1,b[i]);
    add(n+m+1,i+m,0);
  }
  as-=Dinic();
  vector<int>ans1,ans2;
  for(auto u:ans){
    if(u<=n)ans1.push_back(u);
    else ans2.push_back(u-n);
  }
  for(auto u:ans1)cout<<u<<" ";
  cout<<'\n';
  for(auto u:ans2)cout<<u<<" ";
  cout<<'\n';
  cout<<as<<"\n";
}
signed main(){
  ios::sync_with_stdio(false);
  cin.tie(0);
  cout.tie(0);
  int tt=1,k=1;
  //cin>>t;
  while(tt--){
    solve();
  }
}
```

1.4 PDSU

```
#include<iostream>
#include<vector>
#include<string.h>
#include<algorithm>
#include<map>
#include<queue>
#include<stack>
#include<cmath>
#include<set>
#include<unordered_map>
#include<deque>
#include<iomanip>
#include<bitset>
#define ll long long
#define int long long
#define pii pair<int,int>
using namespace std;
const int N=2e5+5;
const int M=1e5+5;
const int INF=2e9;
//const int p=998244353;
const int MOD=1e9+7;
const double DARW=0.97;
struct node{int l,r,x,h;}tr[N<<6];
int head[N],tot,n,m;
void build(int& p,int l,int r){
  if(!p)p=++tot;
  if(l==r){tr[p].x=l;return;}
  int mid=(l+r)>>1;
  build(tr[p].l,l,mid);
  build(tr[p].r,mid+1,r);
}
void change(int p1,int& p2,int l,int r,int x,int f,int h){
  if(!p2)p2=++tot;
  if(l==r){tr[p2].x=f,tr[p2].h=h;return;}
  int mid=(l+r)>>1;
  if(x<=mid)change(tr[p1].l,tr[p2].l,l,mid,x,f,h),tr[p2].r=tr[p1].r;
  else change(tr[p1].r,tr[p2].r,mid+1,r,x,f,h),tr[p2].l=tr[p1].l;
}
pii find(int p,int l,int r,int x){
  if(l==r)return {tr[p].x,tr[p].h};
  int mid=(l+r)>>1;
  if(x<=mid)return find(tr[p].l,l,mid,x);
  else return find(tr[p].r,mid+1,r,x);
}
pii findfa(int p,int x){
  pii y=find(p,1,n,x);
  if(x==y.first)return y;
  return findfa(p,y.first);
}
void merge(int p1,int& p2,int x,int y){
  pii px=findfa(p1,x),py=findfa(p1,y);
  if(px.second>py.second)change(p1,p2,1,n,py.first,px.first,py.second);
  else if(px.second<py.second)change(p1,p2,1,n,px.first,py.first,px.second);
  else {
    int p=0;
    change(p1,p,1,n,py.first,px.first,py.second);
    change(p,p2,1,n,px.first,px.first,px.second+1);
  }
}
void solve(){
  cin>>n>>m;
  build(head[0],1,n);
  for(int i=1;i<=m;i++){
    int op;
    cin>>op;
    if(op==1){
      int x,y;
      cin>>x>>y;
      merge(head[i-1],head[i],x,y);
    }else if(op==2){
      int k;
      cin>>k;
      head[i]=head[k];
    }else {
      int x,y;
      cin>>x>>y;
      head[i]=head[i-1];
      if(findfa(head[i],x).first==findfa(head[i],y).first)cout<<"1\n";
      else cout<<"0\n";
    }
  }
}
signed main(){
```

```
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
  int t=1,k=1;
  //cin>>t;
  while(t--){
      solve();
    }
}
```

1.5 Seg_treap

```
#include<bits/stdc++.h>
#define int long long
const int N=2e5+5;
using namespace std;
struct Node{
    Node *ch[2];
    int val,prio;
    int cnt;
    int siz;
    int to_rev=0;
    Node(int val):val(val),cnt(1),siz(1){
        ch[0]=ch[1]=nullptr;
        prio=rand();
    }
    inline int upd_siz(){
        siz=cnt;
        if(ch[0]!=nullptr)siz+=ch[0]->siz;
        if(ch[1]!=nullptr)siz+=ch[1]->siz;
        return siz;
    }
    inline void pushdown(){
        swap(ch[0],ch[1]);
        if(ch[0]!=nullptr)ch[0]->to_rev^=1;
        if(ch[1]!=nullptr)ch[1]->to_rev^=1;
        to_rev=0;
    }
    inline void cheak_tag(){
        if(to_rev)pushdown();
    }
};
struct Seg_treap{
#define siz(_) (_==nullptr?0:_->siz)
    Node* root;
    pair<Node*,Node*>split(Node* cur,int sz){
        if(cur==nullptr)return {nullptr,nullptr};
        cur->cheak_tag();
        if(sz<=siz(cur->ch[0])){
            auto temp=split(cur->ch[0],sz);
            cur->ch[0]=temp.second;
            cur->upd_siz();
            return {temp.first,cur};
        }else{
            auto temp=split(cur->ch[1],sz-siz(cur->ch[0])-1);
            cur->ch[1]=temp.first;
            cur->upd_siz();
            return {cur,temp.second};
        }
    }
    Node* merge(Node* sm,Node* bg){
        if(sm==nullptr&&bg==nullptr)return nullptr;
        if(sm!=nullptr&&bg==nullptr)return sm;
        if(sm==nullptr&&bg!=nullptr)return bg;
        sm->cheak_tag(),bg->cheak_tag();
        if(sm->prio<bg->prio){
            sm->ch[1]=merge(sm->ch[1],bg);
            sm->upd_siz();
            return sm;
        }else{
            bg->ch[0]=merge(sm,bg->ch[0]);
            bg->upd_siz();
            return bg;
        }
    }
    void insert(int val){
        auto temp=split(root,val);
        auto l_tr=split(temp.first,val-1);
        Node* new_node;
        if(l_tr.second==nullptr)new_node=new Node(val);
        Node* l_tr_combined=merge(l_tr.first,l_tr.second==nullptr?new_node:l_tr.second);
        root=merge(l_tr_combined,temp.second);
    }
    void seg_rev(int l,int r){
        auto less=split(root,l-1);
        auto more=split(less.second,r-l+1);
        more.first->to_rev=1;
        root=merge(less.first,merge(more.first,more.second));
    }
    void print(Node* cur){
        if(cur==nullptr)return;
        cur->cheak_tag();
        print(cur->ch[0]);
        cout<<cur->val<<" ";
        print(cur->ch[1]);
    }
};
Seg_treap tr;
void solve(){
    srand(time(0));
    int n,m;
    cin>>n>>m;
    for(int i=1;i<=n;i++)tr.insert(i);
    while(m--){
        int l,r;
        cin>>l>>r;
        tr.seg_rev(l,r);
    }
    tr.print(tr.root);
}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int t=1,k=1;
    //cin>>t;
    while(t--)solve();
}
```

1.6 bsgs

```
#include<iostream>
#include<map>
#include<cmath>
#define ll long long
using namespace std;
ll bsgs(ll a,ll b,ll p){
    map<ll,ll>mp;
    ll cur=1,t=sqrt(p)+1,now;
    for(int B=0;B<=t;B++){
        mp[b*cur%p]=B;
        now=cur;
        cur=cur*a%p;
    }
    cur=now;
    for(int A=1;A<=t;A++){
```

```
        if(mp[now])return (ll)A*t-mp[now];
        now=now*cur%p;
    }
    return -1;
}
int main(){
    ll p,b,n;
    cin>>p>>b>>n;
    ll ans=bsgs(b,n,p);
    if(ans==-1)cout<<"no solution\n";
    else cout<<ans<<"\n";
}
```

1.7 VirtualTree

```
#include<bits/stdc++.h>
#define int long long
#define pii pair<int,int>
using namespace std;
const int N=5e5+5;
int n,tot,idx[N],rth[N],d[N],f[N][100],lg[N],w[N][100];
vector<pii>e[N];
vector<int>a;
map<int,int>mp;
bool cmp(int x,int y){return idx[x]<idx[y];}
void table_log(){
    for(int i=1;i<=n;i++)lg[i]=log2(i);
}
void dfs0(int x,int fa){
    idx[x]=++tot;
    rth[tot]=x;
    for(auto u:e[x]){
        if(u.first==fa)continue;
        d[u.first]=d[x]+1;
        f[u.first][0]=x;
        w[u.first][0]=u.second;
        for(int i=1;f[f[u.first][i-1]][i-1];i++)f[u.first][i]=f[f[u.first][i-1]][i-1],w[u.first][
            i]=min(w[u.first][i-1],w[f[u.first][i-1]][i-1]);
        dfs0(u.first,x);
    }
}
int get(int u,int v){
    int ans=1e18;
    //cout<<"?\n";
    if(d[u]<d[v])swap(u,v);
    while(d[u]!=d[v])ans=min(ans,w[u][lg[d[u]-d[v]]]),u=f[u][lg[d[u]-d[v]]];
    return ans;
}
int lca(int u,int v){
    if(d[u]<d[v])swap(u,v);
    while(d[u]!=d[v])u=f[u][lg[d[u]-d[v]]];
    //cout<<"?\n";
    if(u==v)return u;
    int t=lg[d[u]];
    while(f[u][0]!=f[v][0]){
        while(f[u][t]==f[v][t])t--;
        u=f[u][t],v=f[v][t];
    }
    return f[u][0];
}
void build(){
    vector<int>s;
    s.push_back(1);
    e[1].clear();
    for(auto x:a){
        if(x==1)continue;
        //cout<<s.back()<<"\n";
        int la=lca(s.back(),x);
        //cout<<la<<"\n";
        if(la!=s.back()){
            int lst=s.back();
            s.pop_back();
            while(s.size()&&idx[la]<idx[s.back()]){
                e[s.back()].push_back({lst,get(lst,s.back())});
                lst=s.back();s.pop_back();
            }//cout<<"?\n";
            if(idx[la]==idx[s.back()]){
                e[s.back()].push_back({lst,get(lst,s.back())});
            }else{
                e[la].clear();
                e[la].push_back({lst,get(lst,la)});
                s.push_back(la);
            }
        }
        e[x].clear();
        s.push_back(x);
    }
    int lst=s.back();
    s.pop_back();
    while(s.size())e[s.back()].push_back({lst,get(lst,s.back())}),lst=s.back(),s.pop_back();
}
int DP(int x){
    int dp=0;
    for(auto u:e[x]){
        if(mp.count(u.first))dp+=u.second;
        else dp+=min(u.second,DP(u.first));
    }
    return dp;
}
void solve(){
    cin>>n;d[1]=1;
    for(int i=1,u,v,w;i<n;i++){
        cin>>u>>v>>w;
        e[u].push_back({v,w});
        e[v].push_back({u,w});
    }
    table_log();
    dfs0(1,0);
    int q;
    cin>>q;
    while(q--){
        int k;
        cin>>k;
        a.clear();mp.clear();
        for(int i=1,z;i<=k;i++)cin>>z,a.push_back(z),mp[z]=1;
        sort(a.begin(),a.end(),cmp);
        //for(auto u:a)cout<<idx[u]<<" ";
        //cout<<"\n";
        build();
        //cout<<"?\n";
        cout<<DP(1)<<"\n";
    }

}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int t=1;
    //cin>>t;
    while(t--)solve();
}
```

1.8 st

```
#include<iostream>
```

```cpp
#include<algorithm>
#include<cmath>
using namespace std;
const int N=1e5+5;
int a[N],f[N][25];
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-48;ch=getchar();}
  return x*f;
}
void st(int n){
    for(int u=1;u<=n;u++)f[u][0]=a[u];
    for(int i=1;(1<<i)<=n;i++){
        for(int u=1;u+(1<<i)-1<=n;u++){
            f[u][i]=max(f[u][i-1],f[u+(1<<(i-1))][i-1]);
        }
    }
}
int main(){
    int n,m;
    cin>>n>>m;
    for(int i=1;i<=n;i++)a[i]=read();
    st(n);
    while(m--){
        int l,r;
        l=read();
        r=read();
        int k=log2(r-l+1);
        cout<<max(f[l][k],f[r-(1<<k)+1][k])<<"\n";
    }
}
```

1.9 AC-Automaton

```cpp
#include<iostream>
#include<vector>
#include<string.h>
#include<algorithm>
#include<map>
#include<queue>
#include<stack>
#include<cmath>
#include<set>
#include<unordered_map>
#include<deque>
#include<iomanip>
#include<bitset>
#define ll long long
#define int long long
using namespace std;
const int N=1e6+5;
const int M=1e8+5;
const int INF=1e9;
const int p=998244353;
const double DARW=0.97;
int tree[N][30],bh,f[N],fail[N],ff[N],lst_[N];
queue<int>q;
void insert(string s){
  int p=0;
  for(int i=0;i<s.length();i++){
    if(!tree[p][s[i]-'a'])tree[p][s[i]-'a']=++bh;
    p=tree[p][s[i]-'a'];
  }
  f[p]++;
}
void build(){
  for(int i=0;i<26;i++)if(tree[0][i])q.push(tree[0][i]);
  while(!q.empty()){
    int p=q.front();
    q.pop();
    for(int i=0;i<26;i++){
      if(tree[p][i])q.push(tree[p][i]),fail[tree[p][i]]=tree[fail[p]][i];
      else tree[p][i]=tree[fail[p]][i];
    }
  }
}
void solve(){
  int n;
  cin>>n;
  for(int i=1;i<=n;i++){
    string ss;
    cin>>ss;
    insert(ss);
  }
  build();
  string t;
  cin>>t;
  int ans=0;
  for(int i=0,j=0;i<t.length();i++){
    j=tree[j][t[i]-'a'];
    for(int u=j;u&&!ff[u];u=fail[u]){
      ans+=f[u],ff[u]=1;
    }
  }
  cout<<ans<<"\n";
}

signed main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int t=1,k=1;
    //cin>>t;
    while(t--){
      solve();
    }
}
```

1.10 Linklist_template

```cpp
#include<bits/stdc++.h>
using namespace std;
template<typename T>
struct Node{
  T date;
  Node* next;
  Node(const T& d):date(d),next(0){}
};
template<typename T>
class LinkList{
public:
  Node<T>* pHead;
  LinkList():pHead(0){};
  ~LinkList();
  int IsEmpty();
  void Print();
  void Insert(const T& value);
  void deletenode(const T& value);
  Node<T>* find(const T& value);
  LinkList& operator=(const LinkList& a);
};
template<typename T>
LinkList<T>::~LinkList(){
  Node<T>* pNow;
```

```
  while(pHead){
    pNow=pHead;
    pHead=pHead->next;
    delete pNow;
  }
}
template<typename T>
int LinkList<T>::IsEmpty(){
  if(pHead)return 0;//首节点存在即非空
  else return 1;
}
template<typename T>
void LinkList<T>::Print(){
  for(Node<T>* pNow=pHead;pNow;pNow=pNow->next){
    cout<<pNow->date<<" ";
  }
  cout<<"\n";
}
template<typename T>
void LinkList<T>::Insert(const T& value){
  Node<T>* p=new Node<T>(value);
  p->next=pHead;
  pHead=p;
}
template<typename T>
void LinkList<T>::deletenode(const T& value){
  Node<T> *pLst=pHead,*pNow=pHead;
  while(pNow&&pNow->date!=value)pLst=pNow,pNow=pNow->next;
    if(!pNow)return;//不存在value
    if(pNow==pHead){//删除节点为首节点
        pHead=pHead->next;
        delete pNow;
    }else {//删除节点非首节点
        pLst->next=pNow->next;
        delete pNow;
    }
}
template<typename T>
Node<T>* LinkList<T>::find(const T& value){
    Node<T> *pNow=pHead;
  while(pNow&&pNow->date!=value)pNow=pNow->next;
    return pNow;
}
template<typename T>
LinkList<T>& LinkList<T>::operator=(const LinkList<T>& a){
    if(this==&a)return *this;//为自身则返回
    this->~LinkList();//释放原来点
  pHead=0;
  if(!a.pHead)return *this;
  pHead=new Node<T>(a.pHead->date);
  pHead->next=0;
  Node<T>* p;
  for(Node<T> *pNow=a.pHead->next,*plst=pHead;pNow;pNow=pNow->next){
    p=new Node<T>(pNow->date);
    plst->next=p;
    p->next=0;
    plst=p;
  }
    return *this;
}
template<typename T>
LinkList<T> Get_same(const LinkList<T>& a,const LinkList<T>& b){
  Node<T> *it1=a.pHead,*it2=b.pHead;
  LinkList<T> c;
  while(it1&&it2){
    if(it1->date==it2->date)c.Insert(it1->date),it1=it1->next,it2=it2->next;
    else if(it1->date>it2->date)it2=it2->next;
    else it1=it1->next;
  }
  return c;
}
int main(){
  LinkList<char> a;
  LinkList<char> b;
  char a1[]{'A','C','D','G','H'},b1[]{'B','C','E','G','H','O'};
  for(int i=4;i>=0;i--)a.Insert(a1[i]);
  for(int i=5;i>=0;i--)b.Insert(b1[i]);
  LinkList<char> c=Get_same(a,b);
  c.Print();
}
```

1.11 EXKMP_z-Fuction

```
#include<iostream>
#include<vector>
#include<string.h>
#include<algorithm>
#include<map>
#include<queue>
#include<stack>
#include<cmath>
#include<set>
#include<unordered_map>
#include<deque>
#include<iomanip>
#include<bitset>
#define ll long long
#define int long long
using namespace std;
const int N=2e7+5;
const int M=1e8+5;
const int INF=1e9;
//const int p=998244353;
const int MOD=1e9+7;
const double DARW=0.97;
int z[N],p[N];
void solve(){
  string a,b;
  cin>>a>>b;
  b=" "+b;
  a=" "+a;
  int ansz=b.length(),ansp=0;
  for(int l=1,r=1,i=2;i<b.length();i++){
    if(i<=r&&z[i-l+1]<r-i+1)z[i]=z[i-l+1];
    else {
      z[i]=max(0*1ll,r-i+1);
      while(i+z[i]<b.length()&&b[i+z[i]]==b[z[i]+1])z[i]++;
      if(i+z[i]-1>r)r=i+z[i]-1,l=i;
    }
    //cout<<z[i]<<" ";
    ansz^=i*(z[i]+1);
  }
  for(int l=0,r=0,i=1;i<a.length();i++){
    if(i<=r&&z[i-l+1]<r-i+1)p[i]=z[i-l+1];
    else {
      p[i]=max(0*1ll,r-i+1);
      while(i+p[i]<a.length()&&p[i]+1<b.length()&&a[i+p[i]]==b[p[i]+1])p[i]++;
      if(i+p[i]-1>r)r=i+p[i]-1,l=i;
    }
    ansp^=i*(p[i]+1);
  }
  cout<<ansz<<"\n"<<ansp<<"\n";
}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int t=1,k=1;
    //cin>>t;
```

```
    while(t--){
      solve();
    }
}
```

1.12 0__to__latex

```
#include <iostream>
#include <fstream>
#include <string>
#include <filesystem>

namespace fs = std::filesystem;

// 仅在下划线前面添加反斜杠进行转义
std::string escape_filename(const std::string& filename) {
    std::string safe_filename = filename;
    size_t pos = 0;
    while ((pos = safe_filename.find('_', pos)) != std::string::npos) {
        safe_filename.insert(pos, "\\"); // 在下划线前面插入反斜杠
        pos += 2; // 更新位置，跳过新插入的字符
    }
    return safe_filename;
}

int main() {
    const std::string output_filename = "listings.tex";
    std::ofstream outfile(output_filename);

    if (!outfile.is_open()) {
        std::cerr << "Failed to open output file." << std::endl;
        return 1;
    }

    outfile << "% Generated LaTeX code for C++ file listings" << std::endl;
    outfile << "\\section{All}\n";
    for (const auto& entry : fs::directory_iterator(".")) {
        const auto& path = entry.path();
        if (entry.is_regular_file() && path.extension() == ".cpp") {
            std::string filename = path.filename().string();
            std::string basename = escape_filename(filename); // 转义文件名

            // 构造子章节标题，这里简单地使用转义后的文件名
            std::string subsection_title = basename.substr(0, basename.size() - 4);

            // 输出到LaTeX文件

            outfile << "\\subsection{" << subsection_title << "}\n";
            outfile << "\\raggedbottom\\lstinputlisting[style=cpp]{assets/" << basename << "}\n";
                    // 假设.cpp文件在assets目录下
            outfile << "\\hrulefill\n\n";
        }
    }

    outfile.close();
    std::cout << "LaTeX listings generated in " << output_filename << std::endl;
    return 0;
}
```

1.13 List

```
#include<iostream>
#include<algorithm>
using namespace std;
struct Node{
  int date;
  Node *next;
  Node(const int& d):date(d),next(0){}
};

bool cmp1(int a,int b){return a<b;}
class List{
  Node *first,*last;
  unsigned listSize;
public:
  List():first(0),last(0),listSize(0){}
  List(int a[],int num);
  List(const List &a);
  unsigned size() const{return listSize;}
  bool empty()const{return !listSize;}
  void push_back(const int& x);
  void push_front(const int& x);
  bool pop_back();
  bool pop_front();
  bool remove(const int& x);
  void remove_all(const int& x);
  void clear();
  void insert(Node *pos,const int& x);
  void erase(const int& w);
  void erase(Node *pos);
  void print()const;
  Node* find(const int& x)const;
  List& operator=(const List& a);
  List& operator+=(const List& a);
  friend List operator+(const List& a,const List&b);
  void listsort(bool (*cmp)(int,int)=cmp1);
  ~List();
};
void List::listsort(bool (*cmp)(int,int)){//根据传入函数排序 默认升序
  int *arr=new int[listSize];//用于储存链表数据，方便排序
  Node *p=first;
  for(int i=0;i<listSize;++i,p=p->next)arr[i]=p->date;//存入数组
  sort(arr,arr+listSize,cmp);//排序
  p=first;
  for(int i=0;i<listSize;++i,p=p->next)p->date=arr[i];//更新链表
  delete[] arr;
}
List operator+(const List& a,const List&b){
  List c(a);
  c+=b;
  return c;
}
List& List::operator+=(const List& a){
  for(Node *p=a.first;p;p=p->next){
    Node *now=new Node(p->date);
    last->next=now;
    last=now;
  }
  listSize+=a.listSize;
  return *this;
}
List& List::operator=(const List& a){
  if(this==&a)return *this;
  new(this) List(a);
  return *this;
}
void List::remove_all(const int& x){//移除所有为x的数
  while(remove(x));
}
void List::erase(Node *pos){//移除指针为pos的数
  if(pos==first)pop_front();//若指定为首地址则利用pop_front函数
  else {
    Node *p=first;
    while(p->next!=pos)p=p->next;//找到pos前指针
    p->next=pos->next;
    if(pos==last)last=p;
```

```
    delete pos;
    --listSize;
  }
}
void List::erase(const int& w){//移除第x个的数
  if(w>listSize)return;//若w大于链表大小则返回
  if(w==1)pop_front();
  else {
    Node *now=first->next,*before=first;
    for(int i=2;i<=w;i++)before=now,now=now->next;//找到第x数所对应指针和第x-1数所对//应指针
    before->next=now->next;
    if(now==last)last=before;
    delete now;
    --listSize;
  }
}
List::List(int a[],int num){//将数组转换为链表
  listSize=num;
  if(!num){
      first=0;
      last=0;
  }else {
    Node *p=new Node(a[0]);
    first=last=p;
    for(int i=1;i<num;i++){
      p=new Node(a[i]);
      last->next=p;
      last=p;
    }
  }
}
List::List(const List &a){//拷贝
  listSize=a.listSize;
  if(!listSize){
    first=last=0;
    return;
  }
  Node *p=new Node(a.first->date),*now=a.first->next;
  first=last=p;
  for(;now;now=now->next){
    p=new Node(now->date);
    last->next=p;
    last=p;
  }
}
void List::push_back(const int& x){//在链表末添加x
  Node *p=new Node(x);
  if(!listSize)first=last=p;
  else {
    last->next=p;
    last=p;
  }
  ++listSize;
}
void List::push_front(const int& x){//将x置入链表首
  Node *p=new Node(x);
  if(!listSize)first=last=p;
  else {
    p->next=first;
    first=p;
  }
  ++listSize;
}
bool List::pop_back(){//移除链表最后一个数
  if(!listSize)return 0;
  if(listSize==1){
    delete[] first;
    first=last=0;
  }else {
    Node *now=first;
    while(now->next!=last)now=now->next;
    last=now;
    delete now->next;
    now->next=0;
  }
  --listSize;
  return 1;
}
bool List::pop_front(){//移除链表第一个数
  if(!listSize)return 0;
  if(listSize==1){
    delete first;
    first=last=0;
  }else {
    Node *now=first;
    first=first->next;
    delete now;
  }
  --listSize;
  return 1;
}
bool List::remove(const int& x){//移除第一个值为x的数
  Node *now=first,*before=first;
  while(now&&now->date!=x)before=now,now=now->next;//找到第一个值为x的数
  if(now){//若存在值为x的数  不存在则返回0
    if(listSize==1){
      delete now;
      first=last=0;
    }else {
      if(now==first){
        first=first->next;
        delete now;
      }
      else if(now==last){
        last=before;
        delete now;
        last->next=0;
      }
      else{
        before->next=now->next;
        delete now;
      }
    }
    --listSize;
    return 1;
  }
  return 0;
}
void List::clear(){//清空链表
  this->~List();
  first=last=0;
  listSize=0;
}
void List::insert(Node *pos,const int& x){//在pos后插入x
  Node *p=new Node(x);
  p->next=pos->next;
  pos->next=p;
  if(pos==last)last=p;
  ++listSize;
}
List::~List(){
  for(Node *p=first;first!=0;p=first){
    first=first->next;
    delete p;
  }
}
void List::print()const{
  for(Node* now=first;now;now=now->next)cout<<now->date<<" ";
  cout<<"\n";
}
Node* List::find(const int& x)const{//寻找第一个值为x的数  未找到则返回0
  for(Node* p=first;p;p=p->next)
```

```
    if(p->date==x)return p;
  return 0;
}
int main(){
  int n;
  cin>>n;
  List a;
  while(n--){
    int num;
    cin>>num;
    a.push_front(num);
  }
  a.print();
  List b;
  b=a;
  b.print();
  cout<<a.find(1)<<'\n';
  cout<<b.find(1)<<"\n";
}
```

1.14 Left partial tree

```
#include<iostream>
#include<vector>
#include<string.h>
#include<algorithm>
#include<map>
#include<queue>
#include<stack>
#include<cmath>
#include<set>
#include<unordered_map>
#include<deque>
#include<iomanip>
#include<bitset>
#define ll long long
#define int long long
using namespace std;
const int N=1e6+5;
const int M=1e8+5;
const int INF=1e9;
const int p=998244353;
const double DARW=0.97;
struct node{int ch[2],d,val;}t[N];
int f[N],ff[N];
int find(int x){
  if(f[x]==x)return x;
  return f[x]=find(f[x]);
}
int& rs(int x){return t[x].ch[t[t[x].ch[1]].d<t[t[x].ch[0]].d];}
int merge(int x,int y){
  if(!x||!y)return x|y;
  if(t[x].val>t[y].val||(t[x].val==t[y].val&&x>y))swap(x,y);
  f[y]=find(x);
  rs(x)=merge(rs(x),y);
  t[x].d=t[rs(x)].d+1;
  return x;
}
void solve(){
  int n,m;
  cin>>n;
  for(int i=1;i<=n;i++){
    cin>>t[i].val;t[i].d=1;
    f[i]=i;ff[i]=1;
  }
  cin>>m;
  while(m--){
    char op;
    int x,y;
    cin>>op;
    if(op=='M'){
      cin>>x>>y;
      if(!ff[x]||!ff[y]||find(x)==find(y))continue;
      merge(find(x),find(y));
    }else {
      cin>>x;
      if(!ff[x]){
        cout<<"0\n";
        continue;
      }
      int k=find(x);
      cout<<t[k].val<<"\n";
      f[t[k].ch[0]]=t[k].ch[0];
      f[t[k].ch[1]]=t[k].ch[1];
      f[k]=merge(t[k].ch[0],t[k].ch[1]);
      ff[k]=0;
    }
  }
}

signed main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int t=1,k=1;
    //cin>>t;
    while(t--){
      solve();
    }
}
```

1.15 FastRead

```
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-48;ch=getchar();}
  return x*f;
}
```

1.16 Inv_FFT

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
const double PI=acos(-1);
const int N=5e6+5;
const int MOD=998244353;
int ksm(int x,int k){
    int ans=1;
    while(k){
        if(k&1)ans=x*ans%MOD;
        x=x*x%MOD;
        k>>=1;
    }
    return ans;
}
namespace Poly{
    int len,Lim=1,Alen,Blen,rev[N],ans[N];
    struct Complex{
        double r,i;
        Complex(){r=0,i=0;}
```

```
        Complex(double real,double imag):r(real),i(imag){}
    };
    inline Complex operator +(Complex A,Complex B){return Complex(A.r+B.r,A.i+B.i);}
    inline Complex operator -(Complex A,Complex B){return Complex(A.r-B.r,A.i-B.i);}
    inline Complex operator *(Complex A,Complex B){return Complex(A.r*B.r-A.i*B.i,A.r*B.i+A.i*B.r
        );}
    inline void init(int Alen,int Blen){
        while(Lim<=Alen+Blen)Lim<<=1,++len;
        for(int i=0;i<Lim;i++)rev[i]=(rev[i>>1]>>1)|((i&1)<<(len-1));
    }
    inline void FFT(vector<Complex>& a,int type){
        for(int i=0;i<Lim;i++)if(i<rev[i])swap(a[i],a[rev[i]]);
        for(int m=2;m<=Lim;m<<=1){
            Complex wn=Complex(cos(2.0*PI/m),sin(2.0*PI/m));
            for(int i=0;i<Lim;i+=m){
                Complex w=Complex(1,0);
                for(int j=0;j<m/2;j++){
                    Complex t=w*a[i+j+m/2];
                    Complex u=a[i+j];
                    a[i+j]=u+t;
                    a[i+j+m/2]=u-t;
                    w=w*wn;
                }
            }
        }
        if(!type){
            reverse(a.begin()+1,a.end());
            for(int i=0;i<Lim;i++)a[i].r/=Lim;
        }
    }
    inline vector<Complex> mul(vector<Complex> f,vector<Complex> g){
        int lf=f.size(),lg=g.size();
        init(lf,lg);
        f.resize(Lim),g.resize(Lim);
        vector<Complex> h(Lim);
        FFT(f,1);FFT(g,1);
        for(int i=0;i<Lim;i++)h[i]=f[i]*g[i];
        FFT(h,0);
        return h;
    }
    vector<Complex> Inv(int n,vector<Complex> A){
        vector<Complex> B(n);
        B[0].r=ksm(A[0].r,MOD-2,MOD);
        int deg=1;
        while(deg<(n<<1)){
            deg<<=1;
            vector<Complex> C=(deg<=n?vector<Complex>(A.begin(),A.begin()+deg):A);
            init(deg,B.size());
            C.resize(Lim);B.resize(Lim);
            FFT(C,1);FFT(B,1);
            for(int i=0;i<Lim;++i)B[i]=B[i]*(Complex(2,0)-C[i]*B[i]);
            FFT(B,0);B.resize(deg);
        }
        B.resize(n);
        return B;
    }
}
int n;
void solve(){
    cin>>n;
    vector<Poly::Complex>A(n);
    for(int i=0;i<n;i++)cin>>A[i].r;
    vector<Poly::Complex>ans=Poly::Inv(n,A);
    for(int i=0;i<n;i++)cout<<(int)(ans[i].r+0.5)<<" ";

}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int t=1;
    // cin>>t;
    while(t--){
        solve();
    }
}
```

1.17 Inv_NTT

```
#include<bits/stdc++.h>
#define ll long long

using namespace std;
const int N=3e6+5;
inline int ksm(int x,int k,int mod){
    int ans=1;
    while(k){
        if(k&1)ans=1ll*ans*x%mod;
        x=1ll*x*x%mod;
        k>>=1;
    }
    return ans;
}
namespace Poly{
    const int MOD=998244353,G=3,INVG=332748118;
    int lim,len,rev[N],invlim;
    inline void init(int l1,int l2){
        lim=1,len=0;
        while(lim<=l1+l2)lim<<=1,len++;
        for(int i=0;i<lim;i++)rev[i]=(rev[i>>1]>>1)|((i&1)<<(len-1));
        invlim=ksm(lim,MOD-2,MOD);
    }
    inline void NTT(vector<int> &f,int type){
        for(int i=0;i<lim;i++)if(i<rev[i])swap(f[i],f[rev[i]]);
        for(int m=2;m<=lim;m<<=1){
            int wn=ksm(type?G:INVG,(MOD-1)/m,MOD);
            for(int i=0;i<lim;i+=m){
                int w=1;
                for(int j=0;j<m/2;j++){
                    int u=f[i+j],v=1ll*w*f[i+j+m/2]%MOD;
                    f[i+j]=(u+v)%MOD,f[i+j+m/2]=(u-v+MOD)%MOD;
                    w=1ll*wn*w%MOD;
                }
            }
        }
        if(!type){
            for(int i=0;i<lim;i++)f[i]=1ll*f[i]*invlim%MOD;
        }
    }
    inline vector<int> mul(vector<int> f,vector<int> g){
        int lf=f.size(),lg=g.size();
        init(lf,lg);
        f.resize(lim),g.resize(lim);
        vector<int> h(lim);
        NTT(f,1);NTT(g,1);
        for(int i=0;i<lim;i++)h[i]=1ll*f[i]*g[i]%MOD;
        NTT(h,0);
        return h;
    }
    vector<int> Inv(int n,vector<int> A){
        vector<int> B(n);
        B[0]=ksm(A[0],MOD-2,MOD);
        int deg=1;
        while(deg<(n<<1)){
            deg<<=1;
            vector<int> C=(deg<=n?vector<int>(A.begin(),A.begin()+deg):A);
            init(deg,B.size());
            C.resize(lim);B.resize(lim);
            NTT(C,1);NTT(B,1);
```

```
            for(int i=0;i<lim;++i)B[i]=B[i]*(2-1ll*C[i]*B[i]%MOD+MOD)%MOD;
            NTT(B,0);B.resize(deg);
        }
        B.resize(n);
        return B;
    }
}
int n;
void solve(){
    cin>>n;
    vector<int> f(n);
    for(int i=0;i<n;i++)cin>>f[i];
    vector<int> g=Poly::Inv(n,f);
    for(int i=0;i<n;i++)cout<<g[i]<<" \n"[i==n-1];
}
int main(){
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    int t=1;
    // cin>>t;
    while(t--){
        solve();
    }
}
```

1.18 Exgcd

```
#include<iostream>
#include<algorithm>
#define ll long long
using namespace std;
const int N=3e6 +5;
ll inv[N];
void exgcd(ll a,ll b,ll &x,ll &y){
    if(!b)x=1,y=0;
    else exgcd(b,a%b,y,x),y-=a/b*x;
}
int main(){
    ll n,p;
    cin>>n>>p;inv[1]=1;
    for(ll i=2;i<=n;i++){
       inv[i]=(((-p/i)*inv[p%i])%p+p)%p;
    }
    for(int i=1;i<=n;i++){
        cout<<inv[i]<<"\n";
    }
}
```

1.19 CSC_dp

```
#include<iostream>
#include<vector>
#include<string.h>
#include<algorithm>
#include<map>
#include<queue>
#include<stack>
#include<cmath>
#include<set>
#include<unordered_map>
#include<deque>
#include<iomanip>
#include<bitset>
#include<functional>
#include<bits/stdc++.h>
#define ll long long
#define int long long
#define pii pair<int,int>
using namespace std;
const int N=5e5+5;
const int M=5e6+5;
const int INF=1e9;
//const int p=998244353;
const int MOD=1e9+7;
const double DARW=0.97;
const double eps=1e-12;
int dp[N],a[20];
int gcd(int a,int b){
    if(!b)return a;
    return gcd(b,a%b);
}
void solve(){
    int n,l,r;
    cin>>n>>l>>r;
    for(int i=1;i<=n;i++)cin>>a[i];
    for(int i=1;i<a[1];i++)dp[i]=2e18;
    dp[0]=0;
    for(int i=2;i<=n;i++){
        for(int j=0,lim=gcd(a[i],a[1]);j<lim;j++){
            for(int u=j,c=0;c<2;c+=(u==j)){
                int p=(u+a[i])%a[1];
                dp[p]=min(dp[p],dp[u]+a[i]);
                u=p;
            }
        }
    }
    int ans=0;
    for(int i=0;i<a[1];i++){
        if(r>=dp[i])ans+=(r-dp[i])/a[1]+1;
        if(l>dp[i])ans-=(l-1-dp[i])/a[1]+1;
    }
    cout<<ans<<"\n";
}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
  int t=1,k=1;
  //cin>>t;
  while(t--)solve();
}
```

1.20 splay

```
#include<iostream>
#include<algorithm>
#include<cstring>
#include<cmath>
#include<map>
#include<stack>
#include<queue>
#include<vector>
#define ll long long
const int N=1e5+5;
using namespace std;
int root,tot;
struct splay{
    int ch[2],fa,val,cnt,size;
}a[N];
void maintain(int x){
    a[x].size=a[a[x].ch[0]].size+a[a[x].ch[1]].size+a[x].cnt;
}
```

```
bool get(int x){
    return a[a[x].fa].ch[1]==x;
}
void clear(int x){
    a[x].ch[0]=a[x].ch[1]=a[x].fa=a[x].val=a[x].cnt=a[x].size=0;
}
void rotate(int x){
    int y=a[x].fa,z=a[y].fa,chk=get(x);
    a[y].ch[chk]=a[x].ch[chk^1];
    if(a[x].ch[chk^1])a[a[x].ch[chk^1]].fa=y;
    a[y].fa=x;
    a[x].ch[chk^1]=y;
    a[x].fa=z;
    if(z)a[z].ch[y==a[z].ch[1]]=x;
    maintain(y);
    maintain(x);
}
void splay(int x){
    for(int f=a[x].fa;f=a[x].fa,f;rotate(x)){
        if(a[f].fa)rotate(get(x)==get(f)?f:x);
    }
    root=x;
}
void insert(int k){
    if(!root){
        a[++tot].val=k;
        a[tot].cnt++;
        root=tot;
        maintain(root);
        return;
    }
    int cur=root,f=0;
    while(1){
        if(a[cur].val==k){
            a[cur].cnt++;
            maintain(cur);
            maintain(f);
            splay(cur);
            return;
        }
        f=cur;
        cur=a[f].ch[a[f].val<k];
        if(!cur){
            a[++tot].val=k;
            a[tot].cnt++;
            a[tot].fa=f;
            a[f].ch[a[f].val<k]=tot;
            maintain(tot);
            maintain(f);
            splay(tot);
            return;
        }
    }
}
int rnk(int x){
    int res=0,cur=root;
    while(1){
        if(x<a[cur].val)cur=a[cur].ch[0];
        else {
            res+=a[a[cur].ch[0]].size;
            if(x==a[cur].val){
                splay(cur);
                return res+1;
            }
            res+=a[cur].cnt;
            cur=a[cur].ch[1];
        }
    }
}
int kth(int k){
    int cur=root;
    while(1){
        if(a[cur].ch[0]&&k<=a[a[cur].ch[0]].size)cur=a[cur].ch[0];
        else {
            k-=a[a[cur].ch[0]].size+a[cur].cnt;
            if(k<=0){
                splay(cur);
                return a[cur].val;
            }
            cur=a[cur].ch[1];
        }

    }
}
int pre(){
    int cur=a[root].ch[0];
    if(!cur)return cur;
    while(a[cur].ch[1])cur=a[cur].ch[1];
    splay(cur);
    return cur;
}
int nxt(){
    int cur=a[root].ch[1];
    if(!cur)return cur;
    while(a[cur].ch[0])cur=a[cur].ch[0];
    splay(cur);
    return cur;
}
void del(int k){
    rnk(k);
    if(a[root].cnt>1){
        a[root].cnt--;
        maintain(root);
        return;
    }
    if(!a[root].ch[0]&&!a[root].ch[1]){
        clear(root);
        root=0;
        return;
    }
    if(!a[root].ch[0]){
        int cur=root;
        root=a[root].ch[1];
        a[root].fa=0;
        clear(cur);
        return;
    }
    if(!a[root].ch[1]){
        int cur=root;
        root=a[root].ch[0];
        a[root].fa=0;
        clear(cur);
        return;
    }
    int cur=root;
    int x=pre();
    a[a[cur].ch[1]].fa=root;
    a[root].ch[1]=a[cur].ch[1];
    clear(cur);
    maintain(root);
}
int main(){
    int n;
    cin>>n;
    for(int i=1;i<=n;i++){
        int x,k;
        cin>>x>>k;
        if(x==1){
            insert(k);
        }else if(x==2){
            del(k);
        }else if(x==3){
```

```
            cout<<rnk(k)<<"\n";
        }else if(x==4){
            cout<<kth(k)<<"\n";
        }else if(x==5){
            insert(k);
            cout<<a[pre()].val<<"\n";
            del(k);
        }else {
            insert(k);
            cout<<a[nxt()].val<<"\n";
            del(k);
        }
    }
}
```

1.21 TCS

```
#include<iostream>
#include<vector>
#include<string.h>
#include<algorithm>
#include<map>
#include<queue>
#include<stack>
#include<cmath>
#include<set>
#include<unordered_map>
#include<deque>
#include<iomanip>
#include<bitset>
#define ll long long
#define int long long
#define pii pair<int,int>
using namespace std;
const int N=1e5+5;
const int M=1e5+5;
const int INF=2e9;
//const int p=998244353;
const int MOD=1e9+7;
const double DARW=0.97;
int n,m,r,p,dfn;
int a[N],tr[N<<2],f[N],son[N],sz[N],idx[N],rnk[N],top[N],bj[N<<2],dep[N];
vector<int>e[N];
void dfs1(int x,int fa){
  f[x]=fa;
  sz[x]=1;
  dep[x]=dep[fa]+1;
  for(auto u:e[x]){
    if(u==fa)continue;
    dfs1(u,x);
    sz[x]+=sz[u];
    if(sz[u]>sz[son[x]])son[x]=u;
  }
}
void dfs2(int x,int fa){
  idx[x]=++dfn;
  rnk[dfn]=x;
  top[x]=fa;
  if(!son[x])return;
  dfs2(son[x],fa);
  for(auto u:e[x]){
    if(!idx[u])dfs2(u,u);
  }
}
void pushdown(int l,int r,int bh){
  if(l==r)return;
  int mid=(l+r)>>1;
  (tr[bh<<1]+=(mid-l+1)*bj[bh])%=p;
  (tr[(bh<<1)|1]+=(r-mid)*bj[bh])%=p;
  (bj[bh<<1]+=bj[bh])%=p;
  (bj[(bh<<1)|1]+=bj[bh])%=p;
  bj[bh]=0;
}
void build(int l,int r,int bh){
  if(l==r){tr[bh]=a[rnk[l]]%p;return;}
  int mid=(l+r)>>1;
  build(l,mid,bh<<1);
  build(mid+1,r,(bh<<1)|1);
  tr[bh]=(tr[bh<<1]+tr[(bh<<1)|1])%p;
}
void add(int l,int r,int x,int y,int bh,int k){
  if(x<=l&&y>=r){(tr[bh]+=(r-l+1)*k)%=p,(bj[bh]+=k)%=p;return;}
  pushdown(l,r,bh);
  int mid=(l+r)>>1;
  if(x<=mid)add(l,mid,x,y,bh<<1,k);
  if(y>mid)add(mid+1,r,x,y,(bh<<1)|1,k);
  tr[bh]=(tr[bh<<1]+tr[(bh<<1)|1])%p;
}
int find(int l,int r,int x,int y,int bh){
  if(x<=l&&y>=r)return tr[bh];
  pushdown(l,r,bh);
  int mid=(l+r)>>1,ans=0;
  if(x<=mid)(ans+=find(l,mid,x,y,bh<<1))%=p;
  if(y>mid)(ans+=find(mid+1,r,x,y,(bh<<1)|1))%=p;
  return ans;
}
void add(int x,int y,int z){
  while(top[x]!=top[y]){
    if(dep[top[x]]<dep[top[y]])swap(x,y);
    add(1,n,idx[top[x]],idx[x],1,z);
    x=f[top[x]];
  }
  if(dep[x]>dep[y])swap(x,y);
  add(1,n,idx[x],idx[y],1,z);
}
int find(int x,int y){
  int ans=0;
  while(top[x]!=top[y]){
    if(dep[top[x]]<dep[top[y]])swap(x,y);
    (ans+=find(1,n,idx[top[x]],idx[x],1))%=p;
    x=f[top[x]];
  }
  if(dep[x]>dep[y])swap(x,y);
  (ans+=find(1,n,idx[x],idx[y],1))%=p;
  return ans;
}
void solve(){
  cin>>n>>m>>r>>p;
  for(int i=1;i<=n;i++)cin>>a[i];
  for(int i=1,u,v;i<n;i++)cin>>u>>v,e[u].push_back(v),e[v].push_back(u);
  dfs1(r,0);
  dfs2(r,r);
  build(1,n,1);
  for(int i=1;i<=m;i++){
    int op;
    cin>>op;
    if(op==1){
      int x,y,z;
      cin>>x>>y>>z;
      add(x,y,z);
    }else if(op==2){
      int x,y;
      cin>>x>>y;
      cout<<find(x,y)<<"\n";
    }else if(op==3){
      int x,z;
      cin>>x>>z;
      add(1,n,idx[x],idx[x]+sz[x]-1,1,z);
    }else {
```

```
        int x;
        cin>>x;
        cout<<find(1,n,idx[x],idx[x]+sz[x]-1,1)<<"\n";
    }
  }
}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
  int t=1,k=1;
  //cin>>t;
  while(t--){
      solve();
    }
}
```

1.22 NTT

```
#include<bits/stdc++.h>
#define ll long long

using namespace std;
const int N=3e6+5;
inline int ksm(int x,int k,int mod){
    int ans=1;
    while(k){
        if(k&1)ans=1ll*ans*x%mod;
        x=1ll*x*x%mod;
        k>>=1;
    }
    return ans;
}
namespace Poly{
    const int MOD=998244353,G=3,INVG=332748118;
    int lim,len,rev[N],invlim;
    inline void init(int l1,int l2){
        lim=1,len=0;
        while(lim<=l1+l2)lim<<=1,len++;
        for(int i=0;i<lim;i++)rev[i]=(rev[i>>1]>>1)|((i&1)<<(len-1));
        invlim=ksm(lim,MOD-2,MOD);
    }
    inline void NTT(vector<int> &f,int type){
        for(int i=0;i<lim;i++)if(i<rev[i])swap(f[i],f[rev[i]]);
        for(int m=2;m<=lim;m<<=1){
            int wn=ksm(type?G:INVG,(MOD-1)/m,MOD);
            for(int i=0;i<lim;i+=m){
                int w=1;
                for(int j=0;j<m/2;j++){
                    int u=f[i+j],v=1ll*w*f[i+j+m/2]%MOD;
                    f[i+j]=(u+v)%MOD,f[i+j+m/2]=(u-v+MOD)%MOD;
                    w=1ll*wn*w%MOD;
                }
            }
        }
        if(!type){
            for(int i=0;i<lim;i++)f[i]=1ll*f[i]*invlim%MOD;
        }
    }
    inline vector<int> mul(vector<int> f,vector<int> g){
        int lf=f.size(),lg=g.size();
        init(lf,lg);
        f.resize(lim),g.resize(lim);
        vector<int> h(lim);
        NTT(f,1);NTT(g,1);
        for(int i=0;i<lim;i++)h[i]=1ll*f[i]*g[i]%MOD;
        NTT(h,0);
        return h;
    }
}
int n,m;
void solve(){
    cin>>n>>m;
    vector<int> f(n+1),g(m+1);
    for(int i=0;i<=n;i++)cin>>f[i];
    for(int i=0;i<=m;i++)cin>>g[i];
    vector<int> h=Poly::mul(f,g);
    for(int i=0;i<=n+m;i++)cout<<h[i]<<" \n"[i==n+m];
}
int main(){
    int t=1;
    // cin>>t;
    while(t--){
        solve();
    }
}
```

1.23 FHQ_treap

```
#include<bits/stdc++.h>
#define int long long
const int N=1e5+5;
using namespace std;

struct Node{
    Node *ch[2];
    int val,prio;
    int cnt;
    int siz;
    Node(int val):val(val),cnt(1),siz(1){
        ch[0]=ch[1]=nullptr;
        prio=rand();
    }
    Node(Node *node){
        val=node->val,prio=node->prio,cnt=node->cnt,siz=node->siz;
    }
    inline void upd_siz(){
        siz=cnt;
        if(ch[0]!=nullptr)siz+=ch[0]->siz;
        if(ch[1]!=nullptr)siz+=ch[1]->siz;
    }
};
struct none_rot_treap{
#define _3 second.second
#define _2 second.first
    vector<Node*>rt;
    none_rot_treap(){rt.push_back(nullptr);}
    pair<Node*,Node*> split(Node* cur,int key){
        if(cur==nullptr)return {nullptr,nullptr};
        if(cur->val<=key){
            auto temp=split(cur->ch[1],key);
            cur->ch[1]=temp.first;
            cur->upd_siz();
            return {cur,temp.second};
        }else{
            auto temp=split(cur->ch[0],key);
            cur->ch[0]=temp.second;
            cur->upd_siz();
            return {temp.first,cur};
        }
    }
    tuple<Node*,Node*,Node*>split_by_rk(Node* cur,int rk){
        if(cur==nullptr)return {nullptr,nullptr,nullptr};
        int ls_siz=cur->ch[0]==nullptr?0:cur->ch[0]->siz;
        if(rk<=ls_siz){
            Node *l,*mid,*r;
```

```
            tie(l,mid,r)=split_by_rk(cur->ch[0],rk);
            cur->ch[0]=r;
            cur->upd_siz();
            return {l,mid,cur};
        }else if(rk<=ls_siz+cur->cnt){
            Node *lt=cur->ch[0];
            Node *rt=cur->ch[1];
            cur->ch[0]=cur->ch[1]=nullptr;
            return {lt,cur,rt};
        }else{
            Node *l,*mid,*r;
            tie(l,mid,r)=split_by_rk(cur->ch[1],rk-ls_siz-cur->cnt);
            cur->ch[1]=l;
            cur->upd_siz();
            return {cur,mid,r};
        }
    }
    Node* merge(Node* u,Node* v){
        if(u==nullptr&&v==nullptr)return nullptr;
        if(u!=nullptr&&v==nullptr)return u;
        if(v!=nullptr&&u==nullptr)return v;
        if(u->prio<v->prio){
            Node* temp=new Node(u);
            temp->ch[0]=u->ch[0];
            temp->ch[1]=merge(u->ch[1],v);
            temp->upd_siz();
            return temp;
        }else{
            Node* temp=new Node(v);
            temp->ch[1]=v->ch[1];
            temp->ch[0]=merge(u,v->ch[0]);
            temp->upd_siz();
            return temp;
        }
    }
    void insert(int val){
        auto temp=split(root,val);
        auto l_tr=split(temp.first,val-1);
        Node *new_node;
        if(l_tr.second==nullptr){
            new_node=new Node(val);
        }else{
            l_tr.second->cnt++;
            l_tr.second->upd_siz();
        }
        Node *l_tr_combined=merge(l_tr.first,l_tr.second==nullptr?new_node:l_tr.second);
        root=merge(l_tr_combined,temp.second);
    }
    void del(int val){
        auto temp=split(root,val);
        auto l_tr=split(temp.first,val-1);
        if(l_tr.second->cnt>1){
            l_tr.second->cnt--;
            l_tr.second->upd_siz();
            l_tr.first=merge(l_tr.first,l_tr.second);
        }else{
            if(temp.first==l_tr.second){
                temp.first=nullptr;
            }
            delete l_tr.second;
            l_tr.second=nullptr;
        }
        root=merge(l_tr.first,temp.second);
    }
    int qrank_by_val(Node *cur,int val){
        auto temp=split(cur,val-1);
        int ret=(temp.first==nullptr?0:temp.first->siz)+1;
        root=merge(temp.first,temp.second);
        return ret;
    }
    int qval_by_rank(Node *cur,int rk){
        Node *l,*mid,*r;
        tie(l,mid,r)=split_by_rk(cur,rk);
        int ret=mid->val;
        root=merge(merge(l,mid),r);
        return ret;
    }
    int qprev(int val){
        auto temp=split(root,val-1);
        int ret=qval_by_rank(temp.first,temp.first->siz);
        root=merge(temp.first,temp.second);
        return ret;
    }
    int qnex(int val){
        auto temp=split(root,val);
        int ret=qval_by_rank(temp.second,1);
        root=merge(temp.first,temp.second);
        return ret;
    }
};
none_rot_treap tr;
int a[N];
void solve(){
    srand(time(0));
    int n,m,ans=0;
    cin>>n>>m;
    for(int i=1;i<=n;i++)cin>>a[i],tr.insert(a[i]);
    int lst=0;
    while(m--){
        int op,x;
        cin>>op>>x;
        x^=lst;
        if(op==1)tr.insert(x);
        else if(op==2)tr.del(x);
        else if(op==3)ans^=(lst=tr.qrank_by_val(tr.root,x));
        else if(op==4)ans^=(lst=tr.qval_by_rank(tr.root,x));
        else if(op==5)ans^=(lst=tr.qprev(x));
        else ans^=(lst=tr.qnex(x));
    }
    cout<<ans<<"\n";
}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int t=1,k=1;
  //cin>>t;
  while(t--){
     solve();
    }
}
```

1.24 CSC_dij

```
#include<iostream>
#include<vector>
#include<string.h>
#include<algorithm>
#include<map>
#include<queue>
#include<stack>
#include<cmath>
#include<set>
#include<unordered_map>
#include<deque>
#include<iomanip>
#include<bitset>
#include<functional>
#include<bits/stdc++.h>
```

```
#define ll long long
#define int long long
#define pii pair<int,int>
using namespace std;
const int N=2e5+5;
const int M=5e6+5;
const int INF=1e9;
//const int p=998244353;
const int MOD=1e9+7;
const double DARW=0.97;
const double eps=1e-12;
int ans[N],f[N];
vector<int>t[N];
struct node{int x,val;bool operator<(const node& a)const{return val>a.val;}};
struct edge{int v,w;};
vector<edge>e[N];
priority_queue<node>q;
void dij(){
    while(q.size()&&f[q.top().x])q.pop();
    if(q.empty())return;
    node x=q.top();
    q.pop();
    f[x.x]=1;
    for(auto u:e[x.x]){
        if(u.w+ans[x.x]<ans[u.v]){
            ans[u.v]=u.w+ans[x.x];
            q.push({u.v,ans[u.v]});
        }
    }
    dij();
}
void solve(){
    int h,a,b,c;
    cin>>h>>a>>b>>c;
    if(a<b)swap(a,b);
    if(b<c)swap(b,c);
    for(int i=0;i<c;i++)e[i].push_back({(i+a)%c,a}),e[i].push_back({(i+b)%c,b}),ans[i]=2e18;
    ans[0]=1;
    q.push({0,ans[0]});
    dij();
    int as=0;
    for(int i=0;i<c;i++){
        if(h>=ans[i])
        as+=(h-ans[i])/c+1;
    }
    cout<<as<<"\n";
}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
  int t=1,k=1;
  //cin>>t;
  while(t--)solve();
}
```

1.25 BarrettReduction

```
struct Mod
{
    long long m, p;
    void init(int pp) { m = ((__int128)1 << 64) / pp; p = pp; }
    long long operator ()(long long x){
        return x - ((__int128(x) * m) >> 64) * p;
    }
} mod;
```

1.26 RandomIncremeMethod

```
#include<iostream>
#include<vector>
#include<string.h>
#include<algorithm>
#include<map>
#include<queue>
#include<stack>
#include<cmath>
#include<set>
#include<unordered_map>
#include<deque>
#include<iomanip>
#include<bitset>
#include<functional>
#include<bits/stdc++.h>
#define ll long long
#define int long long
#define pii pair<int,int>
using namespace std;
const int N=3e5+5;
const int M=5e6+5;
const int INF=1e9;
//const int p=998244353;
const int MOD=1e9+7;
const double DARW=0.97;
const double eps=1e-12;
int cmp(double x,double y){
  if(x-y>eps)return 1;
  if(x-y<eps)return -1;
  return 0;
}
struct node{
  double x,y;
}a[N];
node getmid(node a,node b){return {(a.x+b.x)/2,(a.y+b.y)/2};}
double dist(node a,node b){return sqrt((a.x-b.x)*(a.x-b.x)+(a.y-b.y)*(a.y-b.y));}
struct circle{node o;double r;}c;
circle getc(node x1,node x2,node x3){
  double a=x1.x-x2.x,b=x1.y-x2.y,c=x1.x-x3.x,d=x1.y-x3.y,e=((x1.x*x1.x-x2.x*x2.x)+(x1.y*x1.y-x2.y
      *x2.y))/2,f=((x1.x*x1.x-x3.x*x3.x)+(x1.y*x1.y-x3.y*x3.y))/2;
  node o{(d*e-b*f)/(a*d-b*c),(a*f-c*e)/(a*d-b*c)};
  double r=dist(o,x1);
  return {o,r};
}
void solve(){
  int n;
  cin>>n;
  for(int i=1;i<=n;i++)cin>>a[i].x>>a[i].y;
  random_shuffle(a+1,a+n+1);
  c.o=a[1],c.r=0;
  for(int i=1;i<=n;i++){
    if(cmp(dist(a[i],c.o),c.r)==1){
      c.o=a[i],c.r=0;
      for(int j=1;j<=i-1;j++){
        if(cmp(dist(c.o,a[j]),c.r)==1){
          c.o=getmid(a[i],a[j]);
          c.r=dist(c.o,a[j]);
          for(int u=1;u<=j-1;u++){
            if(cmp(dist(c.o,a[u]),c.r)==1){
              c=getc(a[i],a[j],a[u]);
            }
          }
        }
      }
    }
  }
```

```
  cout<<fixed<<setprecision(16)<<c.r<<"\n"<<c.o.x<<" "<<c.o.y<<"\n";
}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int t=1,k=1;
    //cin>>t;
    while(t--)solve();
}
```

1.27 none_rot_treap

```
#include<bits/stdc++.h>
#define int long long
const int N=1e5+5;
using namespace std;

struct Node{
    Node *ch[2];
    int val,prio;
    int cnt;
    int siz;
    Node(int val):val(val),cnt(1),siz(1){
        ch[0]=ch[1]=nullptr;
        prio=rand();
    }
    Node(Node *node){
        val=node->val,prio=node->prio,cnt=node->cnt,siz=node->siz;
    }
    inline void upd_siz(){
        siz=cnt;
        if(ch[0]!=nullptr)siz+=ch[0]->siz;
        if(ch[1]!=nullptr)siz+=ch[1]->siz;
    }
};
struct none_rot_treap{
#define _3 second.second
#define _2 second.first
    Node* root;
    pair<Node*,Node*> split(Node* cur,int key){
        if(cur==nullptr)return {nullptr,nullptr};
        if(cur->val<=key){
            auto temp=split(cur->ch[1],key);
            cur->ch[1]=temp.first;
            cur->upd_siz();
            return {cur,temp.second};
        }else{
            auto temp=split(cur->ch[0],key);
            cur->ch[0]=temp.second;
            cur->upd_siz();
            return {temp.first,cur};
        }
    }
    tuple<Node*,Node*,Node*>split_by_rk(Node* cur,int rk){
        if(cur==nullptr)return {nullptr,nullptr,nullptr};
        int ls_siz=cur->ch[0]==nullptr?0:cur->ch[0]->siz;
        if(rk<=ls_siz){
            Node *l,*mid,*r;
            tie(l,mid,r)=split_by_rk(cur->ch[0],rk);
            cur->ch[0]=r;
            cur->upd_siz();
            return {l,mid,cur};
        }else if(rk<=ls_siz+cur->cnt){
            Node *lt=cur->ch[0];
            Node *rt=cur->ch[1];
            cur->ch[0]=cur->ch[1]=nullptr;
            return {lt,cur,rt};
        }else{
            Node *l,*mid,*r;
            tie(l,mid,r)=split_by_rk(cur->ch[1],rk-ls_siz-cur->cnt);
            cur->ch[1]=l;
            cur->upd_siz();
            return {cur,mid,r};
        }
    }
    Node* merge(Node* u,Node* v){
        if(u==nullptr&&v==nullptr)return nullptr;
        if(u!=nullptr&&v==nullptr)return u;
        if(v!=nullptr&&u==nullptr)return v;
        if(u->prio<v->prio){
            u->ch[1]=merge(u->ch[1],v);
            u->upd_siz();
            return u;
        }else{
            v->ch[0]=merge(u,v->ch[0]);
            v->upd_siz();
            return v;
        }
    }
    void insert(int val){
        auto temp=split(root,val);
        auto l_tr=split(temp.first,val-1);
        Node *new_node;
        if(l_tr.second==nullptr){
            new_node=new Node(val);
        }else{
            l_tr.second->cnt++;
            l_tr.second->upd_siz();
        }
        Node *l_tr_combined=merge(l_tr.first,l_tr.second==nullptr?new_node:l_tr.second);
        root=merge(l_tr_combined,temp.second);
    }
    void del(int val){
        auto temp=split(root,val);
        auto l_tr=split(temp.first,val-1);
        if(l_tr.second->cnt>1){
            l_tr.second->cnt--;
            l_tr.second->upd_siz();
            l_tr.first=merge(l_tr.first,l_tr.second);
        }else{
            if(temp.first==l_tr.second){
                temp.first=nullptr;
            }
            delete l_tr.second;
            l_tr.second=nullptr;
        }
        root=merge(l_tr.first,temp.second);
    }
    int qrank_by_val(Node *cur,int val){
        auto temp=split(cur,val-1);
        int ret=(temp.first==nullptr?0:temp.first->siz)+1;
        root=merge(temp.first,temp.second);
        return ret;
    }
    int qval_by_rank(Node *cur,int rk){
        Node *l,*mid,*r;
        tie(l,mid,r)=split_by_rk(cur,rk);
        int ret=mid->val;
        root=merge(merge(l,mid),r);
        return ret;
    }
    int qprev(int val){
        auto temp=split(root,val-1);
        int ret=qval_by_rank(temp.first,temp.first->siz);
        root=merge(temp.first,temp.second);
        return ret;
    }
    int qnex(int val){
```

```
        auto temp=split(root,val);
        int ret=qval_by_rank(temp.second,1);
        root=merge(temp.first,temp.second);
        return ret;
    }
};
none_rot_treap tr;
int a[N];
void solve(){
    srand(time(0));
    int n,m,ans=0;
    cin>>n>>m;
    for(int i=1;i<=n;i++)cin>>a[i],tr.insert(a[i]);
    int lst=0;
    while(m--){
        int op,x;
        cin>>op>>x;
        x^=lst;
        if(op==1)tr.insert(x);
        else if(op==2)tr.del(x);
        else if(op==3)ans^=(lst=tr.qrank_by_val(tr.root,x));
        else if(op==4)ans^=(lst=tr.qval_by_rank(tr.root,x));
        else if(op==5)ans^=(lst=tr.qprev(x));
        else ans^=(lst=tr.qnex(x));
    }
    cout<<ans<<"\n";
}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int t=1,k=1;
  //cin>>t;
  while(t--){
      solve();
    }
}
```

1.28 lucas.

```
#include<iostream>
#define ll long long
using namespace std;
ll ini[100005];
ll apow(ll a,ll b,ll p){
    ll ans=1;a%=p;
    while(b){
        if(b&1)ans=ans*a%p;
        a=a*a%p;
        b>>=1;
    }
    return ans;
}
ll C(ll n,ll m,ll p){
    if(n<m)return 0;
    return ini[n]*apow(ini[m],p-2,p)*apow(ini[n-m],p-2,p)%p;
}
ll lucas(ll n,ll m,ll p){
    return (m==0)?1:lucas(n/p,m/p,p)*C(n%p,m%p,p)%p;
}
void table(int p){
    ini[0]=1;
    ini[1]=1;
    for(int i=2;i<=p;i++)ini[i]=ini[i-1]*i%p;
}

int main(){
    int t;
    cin>>t;
    while(t--){
        int n,m,p;
        cin>>n>>m>>p;
        table(p);
        cout<<lucas(n+m,m,p)<<"\n";
    }
}
```

1.29 Binary heap

```
#include<iostream>
#include<vector>
#include<string.h>
#include<algorithm>
#include<map>
#include<queue>
#include<stack>
#include<cmath>
#include<set>
#include<unordered_map>
#include<deque>
#include<iomanip>
#include<bitset>
#define ll long long
#define int long long
using namespace std;
const int N=1e6+5;
const int M=1e8+5;
const int INF=1e9;
const int p=998244353;
const double DARW=0.97;
int n,a[N];
void up(int x){
  while(x/2&&a[x]<a[x/2]){
    if(a[x]<a[x/2])swap(a[x],a[x/2]);
    x/=2;
  }
}
void down(int x){
  while(2*x<=n){
    int t=2*x;
    if(t+1<=n&&a[t+1]<a[t])t++;
    if(a[t]>=a[x])break;
    swap(a[t],a[x]);
    x=t;
  }
}
void build(){
  for(int i=n;i>=1;i--)down(i);
}
void solve(){
  int q;
  cin>>q;
  while(q--){
    int op,x;
    cin>>op;
    if(op==1){
      cin>>x;
      a[++n]=x;
      up(n);
    }else if(op==2)cout<<a[1]<<"\n";
    else swap(a[1],a[n]),n--,down(1);
  }

}

signed main(){
```

```
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int t=1,k=1;
    //cin>>t;
    while(t--){
      solve();
    }
}
```

1.30 Linklist

```
#include<bits/stdc++.h>
using namespace std;
struct Node{
  int date;
  Node* next;
  Node(const int& d):date(d),next(0){}
};
class LinkList{
  Node* pHead;
public:
  LinkList():pHead(0){};
  ~LinkList();
  int IsEmpty();
  void Print();
  void Insert(int value);
  void deletenode(int value);
  Node* find(int value);
  LinkList& operator=(const LinkList& a);
};
LinkList::~LinkList(){
  Node* pNow;
  while(pHead){
    pNow=pHead;
    pHead=pHead->next;
    delete pNow;
  }
}
int LinkList::IsEmpty(){
  if(pHead)return 0;//首节点存在即非空
  else return 1;
}
void LinkList::Print(){
  for(Node* pNow=pHead;pNow;pNow=pNow->next){
    cout<<pNow->date<<" ";
  }
  cout<<"\n";
}
void LinkList::Insert(int value){
  Node* p=new Node(value);
  p->next=pHead;
  pHead=p;
}
void LinkList::deletenode(int value){
  Node *pLst=pHead,*pNow=pHead;
  while(pNow&&pNow->date!=value)pLst=pNow,pNow=pNow->next;
    if(!pNow)return;//不存在value
    if(pNow==pHead){//删除节点为首节点
        pHead=pHead->next;
        delete pNow;
    }else {//删除节点非首节点
        pLst->next=pNow->next;
        delete pNow;
    }
}
Node* LinkList::find(int value){
    Node *pNow=pHead;
  while(pNow&&pNow->date!=value)pNow=pNow->next;
    return pNow;
}
LinkList& LinkList::operator=(const LinkList& a){
    if(this==&a)return *this;//为自身则返回
    this->~LinkList();//释放原来点
  pHead=0;
  if(!a.pHead)return *this;
  pHead=new Node(a.pHead->date);
  pHead->next=0;
  Node* p;
  for(Node *pNow=a.pHead->next,*plst=pHead;pNow;pNow=pNow->next){
    p=new Node(pNow->date);
    plst->next=p;
    p->next=0;
    plst=p;
  }
    return *this;
}
int main(){
  LinkList a;
    int num,n;
  cout<<"the size of list:";
  cin>>n;
  while(n--){//测试输入
    cout<<"cin:";
    cin>>num;
    a.Insert(num);
    cout<<"list a:";
    a.Print();
  }
  LinkList b;
    b=a;
  cout<<"list b:";
  b.Print();
  int finda;
  cout<<"find:";
  cin>>finda;
  Node* it=a.find(finda);//测试查找、=
  cout<<"the address of findnum:";
  cout<<it<<"\n";
  it->date=0;
  cout<<"list a:";
  a.Print();
  cout<<"list b:";
  b.Print();
  int del;
  while(!a.IsEmpty()){//测试删除，判空
    cout<<"del:";
    cin>>del;
    a.deletenode(del);
    cout<<"list a:";
    a.Print();
  }

}
```

1.31 FFT

```
#include<bits/stdc++.h>
#define int long long
#define double long double
using namespace std;
const double PI=acos(-1);
const int N=5e6+5;
namespace Poly{
    int len,Lim=1,Alen,Blen,rev[N],ans[N],p;
    void setP(int x){p=x;}
```

```
    struct Complex{
        double r,i;
        Complex(){r=0,i=0;}
        Complex(double real,double imag):r(real),i(imag){}
    };
    inline Complex operator +(Complex A,Complex B){return Complex(A.r+B.r,A.i+B.i);}
    inline Complex operator -(Complex A,Complex B){return Complex(A.r-B.r,A.i-B.i);}
    inline Complex operator *(Complex A,Complex B){return Complex(A.r*B.r-A.i*B.i,A.r*B.i+A.i*B.r
        );}
    inline void init(int Alen,int Blen){
        while(Lim<=Alen+Blen)Lim<<=1,++len;
        for(int i=0;i<Lim;i++)rev[i]=(rev[i>>1]>>1)|((i&1)<<(len-1));
    }
    inline void FFT(vector<Complex>& a,int type){
        for(int i=0;i<Lim;i++)if(i<rev[i])swap(a[i],a[rev[i]]);
        for(int m=2;m<=Lim;m<<=1){
            Complex wn=Complex(cos(2.0*PI/m),sin(2.0*PI/m));
            for(int i=0;i<Lim;i+=m){
                Complex w=Complex(1,0);
                for(int j=0;j<m/2;j++){
                    Complex t=w*a[i+j+m/2];
                    Complex u=a[i+j];
                    a[i+j]=u+t;
                    a[i+j+m/2]=u-t;
                    w=w*wn;
                }
            }
        }
        if(!type){
            reverse(a.begin()+1,a.end());
            for(int i=0;i<Lim;i++)a[i].r/=Lim;
        }
    }
    inline vector<Complex> mul(vector<Complex> f,vector<Complex> g){
        int lf=f.size(),lg=g.size();
        init(lf,lg);
        f.resize(Lim),g.resize(Lim);
        vector<Complex> h(Lim);
        FFT(f,1);FFT(g,1);
        for(int i=0;i<Lim;i++)h[i]=f[i]*g[i];
        FFT(h,0);
        return h;
    }
    inline vector<int> mul(vector<int> f,vector<int> g){
        int lf=f.size(),lg=g.size();
        vector<Complex>ff(lf);
        vector<Complex>gg(lg);
        for(int i=0;i<lf;i++)ff[i].r=f[i];
        for(int i=0;i<lg;i++)gg[i].r=g[i];
        vector<Complex> hh=mul(ff,gg);
        vector<int> h(lf+lg-1);
        for(int i=0;i<lf+lg-1;i++)h[i]=(int)(hh[i].r+0.5);
        return h;
    }
    inline vector<int> pmul(vector<int> f,vector<int> g){
        int lf=f.size(),lg=g.size();
        init(lf,lg);
        f.resize(Lim),g.resize(Lim);
        vector<Complex> v1(Lim),v2(Lim),v3(Lim),v4(Lim),h1(Lim),h2(Lim),h3(Lim);
        for(int i=0;i<Lim;i++){
            v1[i].r=f[i]>>15;
            v2[i].r=f[i]&((1<<15)-1);
            v3[i].r=g[i]>>15;
            v4[i].r=g[i]&((1<<15)-1);
        }
        FFT(v1,1);FFT(v2,1);FFT(v3,1);FFT(v4,1);
        for(int i=0;i<Lim;i++){
            h1[i]=v1[i]*v3[i];
            h2[i]=v1[i]*v4[i]+v2[i]*v3[i];
            h3[i]=v2[i]*v4[i];
        }
        FFT(h1,0);FFT(h2,0);FFT(h3,0);
        vector<int> h(lf+lg-1);
        for(int i=0,w1,w2,w3;i<lf+lg-1;i++){
            w1=(int)(h1[i].r+0.5)%p;
            w2=(int)(h2[i].r+0.5)%p;
            w3=(int)(h3[i].r+0.5)%p;
            h[i]=(w1*(1<<30)%p+w2*(1<<15)%p+w3)%p;
        }
        return h;
    }
}
int n,m,p;
void solve(){
    cin>>n>>m>>p;
    Poly::setP(p);
    vector<int>A(n+1);
    vector<int>B(m+1);
    for(int i=0;i<=n;i++)cin>>A[i];
    for(int i=0;i<=m;i++)cin>>B[i];
    vector<int>h=Poly::pmul(A,B);
    for(int i=0;i<=n+m;i++)cout<<h[i]<<" ";

}
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int t=1;
    // cin>>t;
    while(t--){
        solve();
    }
}
```

1 数学

1.1 杜教筛

求 $S(n)=\sum_{i=1}^{n}f(i)$，其中 f 是一个积性函数。

构造一个积性函数 g，那么由 $(f*g)(n)=\sum_{d|n}f(d)g(\frac{n}{d})$，得到 $f(n)=(f*g)(n)-\sum_{d|n,d<n}f(d)g(\frac{n}{d})$。

$$g(1)S(n) = \sum_{i=1}^{n}(f*g)(i)-\sum_{i=1}^{n}\sum_{d|i,d<i}f(d)g(\frac{n}{d}) \quad (1)$$

$$\overset{t=\frac{i}{d}}{=} \sum_{i=1}^{n}(f*g)(i)-\sum_{t=2}^{n}g(t)S(\lfloor\frac{n}{t}\rfloor) \quad (2)$$

当然，要能够由此计算 $S(n)$，会对 f,g 提出一些要求：

- $f*g$ 要能够快速求前缀和。
- g 要能够快速求分段和（前缀和）。
- 对于正常的积性函数 $g(1)=1$，所以不会有什么问题。

在预处理 $S(n)$ 前 $n^{\frac{2}{3}}$ 项的情况下复杂度是 $O(n^{\frac{2}{3}})$。

1.2 素性测试

- 前置：快速乘、快速幂
- int 范围内只需检查 2, 7, 61
- long long 范围 2, 325, 9375, 28178, 450775, 9780504, 1795265022
- 3E15 内 2, 2570940, 880937, 610386380, 4130785767
- 4E13 内 2, 2570940, 211991001, 3749873356
- http://miller-rabin.appspot.com/

1.3 扩展欧几里得

- 求 $ax+by=gcd(a,b)$ 的一组解
- 如果 a 和 b 互素，那么 x 是 a 在模 b 下的逆元
- 注意 x 和 y 可能是负数

1.4 类欧几里得

- $m=\lfloor\frac{an+b}{c}\rfloor$.
- $f(a,b,c,n)=\sum_{i=0}^{n}\lfloor\frac{ai+b}{c}\rfloor$：当 $a\geq c$ or $b\geq c$ 时，$f(a,b,c,n)=(\frac{a}{c})n(n+1)/2+(\frac{b}{c})(n+1)+f(a \bmod c,b \bmod c,c,n)$；否则 $f(a,b,c,n)=nm-f(c,c-b-1,a,m-1)$。
- $g(a,b,c,n)=\sum_{i=0}^{n}i\lfloor\frac{ai+b}{c}\rfloor$：当 $a\geq c$ or $b\geq c$ 时，$g(a,b,c,n)=(\frac{a}{c})n(n+1)(2n+1)/6+(\frac{b}{c})n(n+1)/2+g(a \bmod c,b \bmod c,c,n)$；否则 $g(a,b,c,n)=\frac{1}{2}(n(n+1)m-f(c,c-b-1,a,m-1)-h(c,c-b-1,a,m-1))$。
- $h(a,b,c,n)=\sum_{i=0}^{n}\lfloor\frac{ai+b}{c}\rfloor^2$：当 $a\geq c$ or $b\geq c$ 时，$h(a,b,c,n)=(\frac{a}{c})^2n(n+1)(2n+1)/6+(\frac{b}{c})^2(n+1)+(\frac{a}{c})(\frac{b}{c})n(n+1)+h(a \bmod c,b \bmod c,c,n)+2(\frac{a}{c})g(a \bmod c,b \bmod c,c,n)+2(\frac{b}{c})f(a \bmod c,b \bmod c,c,n)$；否则 $h(a,b,c,n)=nm(m+1)-2g(c,c-b-1,a,m-1)-2f(c,c-b-1,a,m-1)-f(a,b,c,n)$。

1.5 斯特灵数

- 第一类斯特灵数：绝对值是 n 个元素划分为 k 个环排列的方案数。$s(n,k)=s(n-1,k-1)+(n-1)s(n-1,k)$
- 第二类斯特灵数：n 个元素划分为 k 个等价类的方案数。$S(n,k)=S(n-1,k-1)+kS(n-1,k)$

1.6 一些数论公式

- 当 $x\geq\phi(p)$ 时有 $a^x\equiv a^{x \bmod \phi(p)+\phi(p)} \pmod p$
- $\mu^2(n)=\sum_{d^2|n}\mu(d)$
- $\sum_{d|n}\varphi(d)=n$
- $\sum_{d|n}2^{\omega(d)}=\sigma_0(n^2)$，其中 ω 是不同素因子个数
- $\sum_{d|n}\mu^2(d)=2^{\omega(d)}$

1.7 一些数论函数求和的例子

- $\sum_{i=1}^{n}i[gcd(i,n)=1]=\frac{n\varphi(n)+[n=1]}{2}$
- $\sum_{i=1}^{n}\sum_{j=1}^{m}[gcd(i,j)=x]=\sum_{d}\mu(d)\lfloor\frac{n}{dx}\rfloor\lfloor\frac{m}{dx}\rfloor$
- $\sum_{i=1}^{n}\sum_{j=1}^{m}gcd(i,j)=\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{d|gcd(i,j)}\varphi(d)=\sum_{d}\varphi(d)\lfloor\frac{n}{d}\rfloor\lfloor\frac{m}{d}\rfloor$
- $S(n)=\sum_{i=1}^{n}\mu(i)=1-\sum_{i=1}^{n}\sum_{d|i,d<i}\mu(d)\overset{t=\frac{i}{d}}{=}1-\sum_{t=2}^{n}S(\lfloor\frac{n}{t}\rfloor)$（利用 $[n=1]=\sum_{d|n}\mu(d)$）
- $S(n)=\sum_{i=1}^{n}\varphi(i)=\sum_{i=1}^{n}i-\sum_{i=1}^{n}\sum_{d|i,d<i}\varphi(i)\overset{t=\frac{i}{d}}{=}\frac{i(i+1)}{2}-\sum_{t=2}^{n}S(\frac{n}{t})$（利用 $n=\sum_{d|n}\varphi(d)$）
- $\sum_{i=1}^{n}\mu^2(i)=\sum_{i=1}^{n}\sum_{d^2|n}\mu(d)=\sum_{d=1}^{\lfloor\sqrt{n}\rfloor}\mu(d)\lfloor\frac{n}{d^2}\rfloor$
- $\sum_{i=1}^{n}\sum_{j=1}^{n}gcd^2(i,j)=\sum_{d}d^2\sum_{t}\mu(t)\lfloor\frac{n}{dt}\rfloor^2\overset{x=dt}{=}\sum_{x}\lfloor\frac{n}{x}\rfloor^2\sum_{d|x}d^2\mu(\frac{t}{x})$
- $\sum_{i=1}^{n}\varphi(i)=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}[i\perp j]-1=\frac{1}{2}\sum_{i=1}^{n}\mu(i)\cdot\lfloor\frac{n}{i}\rfloor^2-1$

1.8 斐波那契数列性质

- $F_{a+b}=F_{a-1}\cdot F_b+F_a\cdot F_{b+1}$
- $F_1+F_3+\cdots+F_{2n-1}=F_{2n},F_2+F_4+\cdots+F_{2n}=F_{2n+1}-1$
- $\sum_{i=1}^{n}F_i=F_{n+2}-1$
- $\sum_{i=1}^{n}F_i^2=F_n\cdot F_{n+1}$
- $F_n^2=(-1)^{n-1}+F_{n-1}\cdot F_{n+1}$
- $gcd(F_a,F_b)=F_{gcd(a,b)}$
- 模 n 周期（皮萨诺周期）
  - $\pi(p^k)=p^{k-1}\pi(p)$
  - $\pi(nm)=lcm(\pi(n),\pi(m)),\forall n\perp m$
  - $\pi(2)=3,\pi(5)=20$
  - $\forall p\equiv\pm1 \pmod{10},\pi(p)|p-1$
  - $\forall p\equiv\pm2 \pmod 5,\pi(p)|2p+2$

1.9 常见生成函数

- $(1+ax)^n=\sum_{k=0}^{n}\binom{n}{k}a^kx^k$
- $\frac{1-x^{r+1}}{1-x}=\sum_{k=0}^{n}x^k$
- $\frac{1}{1-ax}=\sum_{k=0}^{\infty}a^kx^k$

- $\frac{1}{(1-x)^2} = \sum_{k=0}^{\infty}(k+1)x^k$
- $\frac{1}{(1-x)^n} = \sum_{k=0}^{\infty}\binom{n+k-1}{k}x^k$
- $e^x = \sum_{k=0}^{\infty}\frac{x^k}{k!}$
- $\ln(1+x) = \sum_{k=0}^{\infty}\frac{(-1)^{k+1}}{k}x^k$

1.10 佩尔方程

若一个丢番图方程具有以下的形式：$x^2 - ny^2 = 1$。且 n 为正整数，则称此二元二次不定方程为**佩尔方程**。

若 n 是完全平方数，则这个方程式只有平凡解 $(\pm 1, 0)$（实际上对任意的 n，$(\pm 1, 0)$ 都是解）。对于其余情况，拉格朗日证明了佩尔方程总有非平凡解。而这些解可由 $\sqrt{n}$ 的连分数求出。

$$x = [a_0; a_1, a_2, a_3] = x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\ddots}}}}$$

设 $\frac{p_i}{q_i}$ 是 $\sqrt{n}$ 的连分数表示：$[a_0; a_1, a_2, a_3, \ldots]$ 的渐近分数列，由连分数理论知存在 i 使得 (p_i, q_i) 为佩尔方程的解。取其中最小的 i，将对应的 (p_i, q_i) 称为佩尔方程的基本解，或最小解，记作 (x_1, y_1)，则所有的解 (x_i, y_i) 可表示成如下形式：$x_i + y_i\sqrt{n} = (x_1 + y_1\sqrt{n})^i$。或者由以下的递回关系式得到：

$x_{i+1} = x_1x_i + ny_1y_i$, $y_{i+1} = x_1y_i + y_1x_i$。

通常，佩尔方程结果的形式通常是 $a_n = ka_{n-1} - a_{n-2}$（a_{n-2} 前的系数通常是 -1）。暴力 / 凑出两个基础解之后加上一个 0，容易解出 k 并验证。

1.11 Burnside & Polya

$|X/G| = \frac{1}{|G|}\sum_{g\in G}|X^g|$。$X^g$ 是 g 下的不动点数量，也就是说有多少种东西用 g 作用之后可以保持不变。

$|Y^X/G| = \frac{1}{|G|}\sum_{g\in G} m^{c(g)}$。用 m 种颜色染色，然后对于某一种置换 g，有 $c(g)$ 个置换环，为了保证置换后颜色仍然相同，每个置换环必须染成同色。

1.12 皮克定理

$2S = 2a + b - 2$

- S 多边形面积
- a 多边形内部点数
- b 多边形边上点数

1.13 莫比乌斯反演

- $g(n) = \sum_{d|n} f(d) \Leftrightarrow f(n) = \sum_{d|n}\mu(d)g(\frac{n}{d})$
- $f(n) = \sum_{n|d} g(d) \Leftrightarrow g(n) = \sum_{n|d}\mu(\frac{d}{n})f(d)$

1.14 低阶等幂求和

- $\sum_{i=1}^{n} i^1 = \frac{n(n+1)}{2} = \frac{1}{2}n^2 + \frac{1}{2}n$
- $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} = \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n$
- $\sum_{i=1}^{n} i^3 = \left[\frac{n(n+1)}{2}\right]^2 = \frac{1}{4}n^4 + \frac{1}{2}n^3 + \frac{1}{4}n^2$
- $\sum_{i=1}^{n} i^4 = \frac{n(n+1)(2n+1)(3n^2+3n-1)}{30} = \frac{1}{5}n^5 + \frac{1}{2}n^4 + \frac{1}{3}n^3 - \frac{1}{30}n$
- $\sum_{i=1}^{n} i^5 = \frac{n^2(n+1)^2(2n^2+2n-1)}{12} = \frac{1}{6}n^6 + \frac{1}{2}n^5 + \frac{5}{12}n^4 - \frac{1}{12}n^2$

1.15 一些组合公式

- 错排公式：$D_1 = 0, D_2 = 1, D_n = (n-1)(D_{n-1} + D_{n-2}) = n!(\frac{1}{2!} - \frac{1}{3!} + \cdots + (-1)^n\frac{1}{n!}) = \lfloor\frac{n!}{e} + 0.5\rfloor$
- 卡塔兰数（n 对括号合法方案数，n 个结点二叉树个数，$n \times n$ 方格中对角线下方的单调路径数，凸 $n+2$ 边形的三角形划分数，n 个元素的合法出栈序列数）：$C_n = \frac{1}{n+1}\binom{2n}{n} = \frac{(2n)!}{(n+1)!n!}$

1.16 伯努利数与等幂求和

$\sum_{i=0}^{n} i^k = \frac{1}{k+1}\sum_{i=0}^{k}\binom{k+1}{i}B_{k+1-i}(n+1)^i$。也可以 $\sum_{i=0}^{n} i^k = \frac{1}{k+1}\sum_{i=0}^{k}\binom{k+1}{i}B^+_{k+1-i}n^i$。区别在于 $B^+_1 = 1/2$。

1.17 数论分块

$f(i) = \lfloor\frac{n}{i}\rfloor = v$ 时 i 的取值范围是 $[l, r]$。

```
for (LL l = 1, v, r; l <= N; l = r + 1) {
    v = N / l; r = N / v;
}
```

1.18 博弈

- Nim 游戏：每轮从若干堆石子中的一堆取走若干颗。先手必胜条件为石子数量异或和非零。
- 阶梯 Nim 游戏：可以选择阶梯上某一堆中的若干颗向下推动一级，直到全部推下去。先手必胜条件是奇数阶梯的异或和非零（对于偶数阶梯的操作可以模仿）。
- Anti-SG：无法操作者胜。先手必胜的条件是：
  - SG 不为 0 且某个单一游戏的 SG 大于 1 。
  - SG 为 0 且没有单一游戏的 SG 大于 1。
- Every-SG：对所有单一游戏都要操作。先手必胜的条件是单一游戏中的最大 step 为奇数。
  - 对于终止状态 step 为 0
  - 对于 SG 为 0 的状态，step 是最大后继 step +1
  - 对于 SG 非 0 的状态，step 是最小后继 step +1
- 树上删边：叶子 SG 为 0，非叶子结点为所有子结点的 SG 值加 1 后的异或和。

尝试：

- 打表找规律
- 寻找一类必胜态（如对称局面）
- 直接博弈 dp

2 图论

2.1 带下界网络流

- 无源汇：$u \to v$ 边容量为 $[l,r]$，连容量 $r-l$，虚拟源点到 v 连 l，u 到虚拟汇点连 l。
- 有源汇：为了让流能循环使用，连 $T \to S$，容量 ∞。
- 最大流：跑完可行流后，加 $S' \to S$，$T \to T'$，最大流就是答案（$T \to S$ 的流量自动退回去了，这一部分就是下界部分的流量）。
- 最小流：T 到 S 的那条边的实际流量，减去删掉那条边后 T 到 S 的最大流。
- 网上说可能会减成负的，还要有限地供应 S 之后，再跑一遍 S 到 T 的。
- 费用流：必要的部分（下界以下的）不要钱，剩下的按照最大流。

2.2 二分图匹配

- 最小覆盖数 = 最大匹配数
- 最大独立集 = 顶点数 - 二分图匹配数
- DAG 最小路径覆盖数 = 结点数 - 拆点后二分图最大匹配数

2.3 差分约束

一个系统 n 个变量和 m 个约束条件组成，每个约束条件形如 $x_j - x_i \le b_k$。可以发现每个约束条件都形如最短路中的三角不等式 $d_u - d_v \le w_{u,v}$。因此连一条边 (i,j,b_k) 建图。

若要使得所有量两两的值最接近，源点到各点的距离初始成 0，跑最远路。

若要使得某一变量与其他变量的差尽可能大，则源点到各点距离初始化成 ∞，跑最短路。

2.4 三元环

将点分成度入小于 $\sqrt{m}$ 和超过 $\sqrt{m}$ 的两类。现求包含第一类点的三元环个数。由于边数较少，直接枚举两条边即可。由于一个点度数不超过 $\sqrt{m}$，所以一条边最多被枚举 $\sqrt{m}$ 次，复杂度 $O(m\sqrt{m})$。再求不包含第一类点的三元环个数，由于这样的点不超过 $\sqrt{m}$ 个，所以复杂度也是 $O(m\sqrt{m})$。

对于每条无向边 (u,v)，如果 $d_u < d_v$，那么连有向边 (u,v)，否则有向边 (v,u)。度数相等的按第二关键字判断。然后枚举每个点 x，假设 x 是三元组中度数最小的点，然后暴力往后面枚举两条边找到 y，判断 (x,y) 是否有边即可。复杂度也是 $O(m\sqrt{m})$。

2.5 四元环

考虑这样一个四元环，将答案统计在度数最大的点 b 上。考虑枚举点 u，然后枚举与其相邻的点 v，然后再枚举所有度数比 v 大的与 v 相邻的点，这些点显然都可能作为 b 点，我们维护一个计数器来计算之前 b 被枚举多少次，答案加上计数器的值，然后计数器加一。

枚举完 u 之后，我们用和枚举时一样的方法来清空计数器就好了。

任何一个点，与其直接相连的度数大于等于它的点最多只有 $\sqrt{2m}$ 个。所以复杂度 $O(m\sqrt{m})$。

2.6 支配树

- `semi[x]` 半必经点（就是 x 的祖先 z 中，能不经过 z 和 x 之间的树上的点而到达 x 的点中深度最小的）
- `idom[x]` 最近必经点（就是深度最大的根到 x 的必经点）

3 计算几何

3.1 k 次圆覆盖

一种是用竖线进行切分，然后对每一个切片分别计算。扫描线部分可以魔改，求各种东西。复杂度 $O(n^3 \log n)$。

复杂度 $O(n^2 \log n)$。原理是：认为所求部分是一个奇怪的多边形 + 若干弓形。然后对于每个圆分别求贡献的弓形，并累加多边形有向面积。可以魔改扫描线的部分，用于求周长、至少覆盖 k 次等等。内含、内切、同一个圆的情况，通常需要特殊处理。

3.2 三维凸包

增量法。先将所有的点打乱顺序，然后选择四个不共面的点组成一个四面体，如果找不到说明凸包不存在。然后遍历剩余的点，不断更新凸包。对遍历到的点做如下处理。

1. 如果点在凸包内，则不更新。
2. 如果点在凸包外，那么找到所有原凸包上所有分隔了对于这个点可见面和不可见面的边，以这样的边的两个点和新的点创建新的面加入凸包中。

4 随机素数表

42737, 46411, 50101, 52627, 54577, 191677, 194869, 210407, 221831, 241337, 578603, 625409, 713569, 788813, 862481, 2174729, 2326673, 2688877, 2779417, 3133583, 4489747, 6697841, 6791471, 6878533, 7883129, 9124553, 10415371, 11134633, 12214801, 15589333, 17148757, 17997457, 20278487, 27256133, 28678757, 38206199, 41337119, 47422547, 48543479, 52834961, 76993291, 85852231, 95217823, 108755593, 132972461, 171863609, 173629837, 176939899, 207808351, 227218703, 306112619, 311809637, 322711981, 330806107, 345593317, 345887293, 362838523, 373523729, 394207349, 409580177, 437359931, 483577261, 490845269, 512059357, 534387017, 698987533, 764016151, 906097321, 914067307, 954169327

适合哈希的素数：1572869, 3145739, 6291469, 12582917, 25165843, 50331653

NTT 素数表：$p = r2^k + 1$，原根是 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; 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.

5 心态崩了

- `(int)v.size()`
- `1LL << k`
- 递归函数用全局或者 static 变量要小心
- 预处理组合数注意上限
- 想清楚到底是要 `multiset` 还是 `set`
- 提交之前看一下数据范围，测一下边界
- 数据结构注意数组大小（2 倍，4 倍）
- 字符串注意字符集
- 如果函数中使用了默认参数的话，注意调用时的参数个数
- 注意要读完
- 构造参数无法使用自己
- 树链剖分/dfs 序，初始化或者询问不要忘记 idx, ridx
- 排序时注意结构体的所有属性是不是考虑了
- 不要把 while 写成 if
- 不要把 int 开成 char
- 清零的时候全部用 0 到 $n + 1$。
- 模意义下不要用除法
- 哈希不要自然溢出
- 最短路不要 SPFA，乖乖写 Dijkstra
- 上取整以及 GCD 小心负数
- mid 用 `l + (r - l) / 2` 可以避免溢出和负数的问题
- 小心模板自带的意料之外的隐式类型转换
- 求最优解时不要忘记更新当前最优解
- 图论问题一定要注意图不连通的问题
- 处理强制在线的时候 lastans 负数也要记得矫正
- 不要觉得编译器什么都能优化