Contents

[1. LŨY THỪA MA TRẬN 1](#_Toc500189544)

[2. TERNARY SEARCH 2](#_Toc500189545)

[3. MỘT SỐ ĐƯỢC GỌI LÀ KHÔNG MAY MẮN NẾU NÓ CHỨA SỐ 4 HOẶC 62. CHO ĐOẠN [L;R](0<=L<=R<=10^6). ĐẾM SỐ LƯỢNG SỐ MAY MẮN TRONG ĐOẠN TRÊN 2](#_Toc500189546)

[4. TÍNH TỔNG CỦA MỘT MẢNG TRONG ĐỌAN [L;R](1<=L<=R<=N) SỬ DỤNG BIT 2](#_Toc500189547)

[5. CHO MỘT MẢNG GỒM N PHẦN TỬ. 2 TRUY VẤN: TRUY VẤN 1: L R V - CỘNG V VÀO CÁC PHẦN TỬ TỪ L ĐẾN R VÀ TRUY VẤN 2: IN RA TỔNG CÁC PHẦN TỬ TỪ L ĐẾN R. SỬ DỤNG SEGMENT TRÊ VÀ UPDATE LAZY 3](#_Toc500189548)

[6. CHO MỘT MẢNG GỒM N PHẦN TỬ. TRUY VẤN L R K – IN RA GIÁ TRỊ NHỎ THỨ K TRONG ĐOẠN [L;R] 4](#_Toc500189549)

[7. KHOẢNG CÁCH XA NHẤT GIỮA 2 CẶP ĐỈNH TRONG 1 CÂY (TREE DIAMETER) 4](#_Toc500189550)

[8. CHO 1 MẢNG GÒM N PHẦN TỬ VÀ 3 TRUY VẤN S X Y - UPDATE GIÁ TRỊ TẠI VỊ TRÍ THỨ X THÀNH Y , R X Y – ĐẢO NGƯỢC MẢNG CON ĐOẠN [X,Y] , Q X Y – IN RA TỔNG CÁC SỐ TỪ [X,Y] (SQPLAY TREE) 5](#_Toc500189551)

[9. TÌM CÂY KHUNG NHỎ NHẤT 5](#_Toc500189552)

[10. THUẬT TOÁN HUNGARIAN 5](#_Toc500189553)

[11. THUẬT TOÁN DIJKSTRA 6](#_Toc500189554)

[12. TOPOSORT 6](#_Toc500189555)

[13. ĐẾM THÀNH PHẦN LIÊN THÔNG MẠNH – TARJAN 7](#_Toc500189556)

[14. LUỒNG CỰC ĐẠI 7](#_Toc500189557)

[15. TRIE 7](#_Toc500189558)

[16. RMQ 10](#_Toc500189559)

[17. KHỬ GUASS 10](#_Toc500189560)

[18. TÍNH C(N,M) MOD P 11](#_Toc500189561)

[19. ĐẾM NGHIỆM NGUYÊN DƯƠNG CỦA PHƯƠNG TRÌNH AX+BY+CZ=D (EXTEND - EUCLID) 11](#_Toc500189562)

[20. HÀM PHI-EULER 12](#_Toc500189563)

[21. INVERSE ELEMENT 12](#_Toc500189564)

[22. ĐỊNH LÝ THẶNG DƯ TRUNG HOA 12](#_Toc500189565)

[23. CHO 3 ĐIỂM . TÌM TÂM ĐƯỜNG TRÒN NGOẠI TIẾP 14](#_Toc500189566)

[24. SLIDING WINDOW – CỬA SỔ TRƯỢT 14](#_Toc500189567)

[25. CÓ BAO NHIÊU XÂU CON(LIÊN TỤC) KHÁC NHAU XUẤT HIỆN TRONG T(AC AUTOMATA) 15](#_Toc500189568)

[26. CHO MẢNG A GỒM N PHẦN TỬ, CÁC PHẦN TỬ THUỘC ĐOẠN [1,K]. CÓ M TRUY VẤN, MỖI TRUY VẤN CÓ DẠNG L R – TÍNH TỔNG SIGMA(I,K)C\_I^2. TRONG ĐÓ C\_I LÀ SỐ LẦN XUẤT HIỆN CỦA I TRONG ĐOẠN [L,R] 15](#_Toc500189569)

[27. CHO N CẶP ĐIỂM THỂ HIỆN N HÌNH CHỮ NHẬT. TÍNH DIỆN BAO PHỦ 15](#_Toc500189570)

[28. MILLER-RABIN TEST 15](#_Toc500189571)

[29. KMP 15](#_Toc500189571)

[30. ĐẾM KHỚP VÀ CẦU 15](#_Toc500189571)

[31. THUẬT TOÁN Z 15](#_Toc500189571)

[32. BIGNUM TEMPLATE 15](#_Toc500189571)

[33. AHO CORASICK 15](#_Toc500189571)

1. LŨY THỪA MA TRẬN

#include<bits/stdc++.h>

using namespace std;

#define ll long long

#define MAXN 110

int n;

const ll mod = 998;

struct Matrix{

ll a[MAXN][MAXN];

Matrix(ll x=0){

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

for(int j=0;j<n;j++){

a[i][j]=x\*(i==j);

}

}

}

Matrix operator \*(const Matrix &b)const{

Matrix res(0);

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

for(int j=0;j<n;j++){

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

ll &ma = res.a[i][j];

ma = (ma+a[i][k]\*b.a[k][j])%mod;

}

}

}

return res;

}

};

Matrix qpow(Matrix d,ll m){

Matrix res(1);

while(m){

if(m&1){

m--;

res=res\*d;

}

d=d\*d;

m>>=1;

}

return res;

}

int main(){

int p;

Matrix inp;

cin>>n>>p;

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

for(int j=0;j<n;j++){

scanf("%lld",&inp.a[i][j]);

}

}

Matrix res = qpow(inp,p);

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

for(int j=0;j<n;j++){

cout<<res.a[i][j];

if(j!=n-1) cout<<' ';

}

cout<<endl;

}

}

/\*

==================================+

INPUT: |

3 3

1 0 0

0 1 0

0 0 1

==================================+

OUTPUT: |

1 0 0

0 1 0

0 0 1

==================================+

\*/

1. KIỂM TRA 2 ĐỈNH BẤT KÌ TRONG ĐỒ THỊ CÓ KẾT NỐI VỚI NHAU HAY KHÔNG ?

#include<bits/stdc++.h>

using namespace std;

#define MAXN 100000

int par[MAXN];

int rankk[MAXN];

void init(int n){

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

par[i]=i;

rankk[i]=0;

}

}

int find(int x){

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

else return par[x]=find(par[x]);

}

void merge(int x,int y){

x=find(x);

y=find(y);

if(x==y) return;

if(rankk[x]<rankk[y]) par[x]=y;

else{

par[y]=x;

if(rankk[x]==rankk[y]) rankk[x]++;

}

}

int main(){

int n,m;

cin>>n>>m;

init(n);

int a,b;

while(m--){

cin>>a>>b;

merge(a,b);

}

cin>>m;

while(m--){

cin>>a>>b;

if(find(a)==find(b)) cout<<"YES"<<endl;

else cout<<"NO"<<endl;

}

}

/\*

==================================+

INPUT:

5 3

1 4

4 5

2 3

4

1 2

1 5

1 3

2 3

==================================+

OUTPUT:

NO

YES

NO

YES

==================================+

\*/

1. MỘT SỐ ĐƯỢC GỌI LÀ KHÔNG MAY MẮN NẾU NÓ CHỨA SỐ 4 HOẶC 62. CHO ĐOẠN [L,R] (0<=L<=R<=10^6). ĐẾM SỐ LƯỢNG SỐ MAY MẮN TRONG ĐOẠN TRÊN.

#include<bits/stdc++.h>

using namespace std;

int l,r;

const int MAXN = 25;

int dp[MAXN][2];

int a[MAXN];

int n,m;

int dfs(int pos,bool stat,int pre,bool limit){

if(pos==-1) return 1;

if(!limit&&dp[pos][stat]!=-1) return dp[pos][stat];

int up = limit?a[pos]:9;

int ans = 0;

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

if(pre==6&&i==2) continue;

if(i==4) continue;

ans += dfs(pos-1,i==6,i,limit&&i==a[pos]);

}

if(!limit) dp[pos][stat] = ans;

return ans;

}

int solve(int x){

int px = 0;

while(x){

a[px++] = x%10;

x/=10;

}

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

return dfs(px-1,0,-1,1);

}

int main(){

while(cin>>l>>r){

if(l==0&&r==0) break;

//cout<<solve(r)<<' '<<solve(l-1)<<endl;

cout<<solve(r)-solve(l-1)<<endl;

}

}

/\*

==================================+

INPUT: |

61 65

1 5

2 6

0 0

==================================+

OUTPUT: |

3

4

4

==================================+

\*/

1. TÍNH TỔNG CỦA MỘT MẢNG TRONG ĐOẠN [L;R](1<=L<=R<=N) SỬ DỤNG BIT

#include<bits/stdc++.h>

using namespace std;

#define ll long long

const int MAXN = 1e6+5;

inline int lowbit(int x){return x&(-x);}

int n,m;

ll t[MAXN],a[MAXN];

inline void init(){

memset(t,0,sizeof(t));

}

void add(int x,int k){

for(;x<=n;x+=lowbit(x)) t[x]+=k;

}

ll sum(int x){

ll ans = 0;

for(;x>0;x-=lowbit(x)) ans+=t[x];

return ans;

}

int main(){

cin>>n;

ll i;

for(i=1;i<=n;i++) cin>>a[i];

for(i=1;i<=n;i++) add(i,a[i]);

cin>>m;

while(m--){

int l,r;

cin>>l>>r;

cout<<sum(r)-sum(l-1)<<'\n';

}

return 0;

}

/\*

==================================+

INPUT: |

5

1 2 3 4 5

3

1 2

1 4

2 5

==================================+

OUTPUT: |

3

10

14

==================================+

\*/

1. CHO MỘT MẢNG GỒM N PHẦN TỬ. 2 TRUY VẤN: TRUY VẤN 1: L R V – CỘNG V VÀO CÁC PHẦN TỬ TỪ L ĐẾN R VÀ TRUY VẤN 2: L R. IN RA TỔNG CÁC PHẦN TỬ TỪ L ĐẾN R. SỬ DỤNG SEGMENT TREE VÀ UPDATE LAZE.

#include<iostream>

#include<algorithm>

#include<cstring>

#include<cstdio>

using namespace std;

#define MAXN 10000

struct Tree{

int sum;

int lazy;

}t[MAXN<<2];

int ans,n,m;

void pushup(int pos){

t[pos].sum=t[pos<<1].sum + t[pos<<1|1].sum;

return;

}

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

if(!t[pos].lazy) return;

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

t[pos<<1].sum+=t[pos].lazy\*(mid-l+1);

t[pos<<1|1].sum+=t[pos].lazy\*(r-(mid+1)+1);

t[pos<<1].lazy+=t[pos].lazy;

t[pos<<1|1].lazy+=t[pos].lazy;

t[pos].lazy=0;

}

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

t[pos].sum=t[pos].lazy=0;

if(l==r){

scanf("%d",&t[pos].sum);

return;

}

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

build(l,mid ,pos<<1);

build(mid+1,r,pos<<1|1);

pushup(pos);

}

void update(int L,int R,int l,int r,int pos,int v){

if(L<=l&&r<=R){

t[pos].sum+=v\*(r-l+1);

t[pos].lazy+=v;

return;

}

if(r<L||R<l) return;

pushdown(l,r,pos);

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

update(L,R,l,mid,pos<<1,v);

update(L,R,mid+1,r,pos<<1|1,v);

pushup(pos);

}

void query(int L,int R,int l,int r,int pos){

if(L<=l&&r<=R){

ans+=t[pos].sum;

return;

}

if(r<L||R<l) return;

pushdown(l,r,pos);

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

query(L,R,l,mid,pos<<1);

query(L,R,mid+1,r,pos<<1|1);

return;

}

int main(){

int l,r,v;

cin>>n;

build(1,n,1);

cin>>m;

while(m--){

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

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

}

cin>>m;

while(m--){

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

ans=0;

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

cout<<ans<<endl;

}

}

/\*

==================================+

INPUT: |

5

1 2 3 4 5

1

1 5 2

1

1 5

==================================+

OUTPUT: |

25

==================================+

\*/

1. CHO MỘT MẢNG GỒM N PHẦN TỬ. TRUY VẤN: L R K – IN RA GIÁ TRỊ NHỎ THỨ K TRONG ĐOẠN [L;R]

#include<iostream>

#include<algorithm>

using namespace std;

int cansel\_sync=(ios::sync\_with\_stdio(0),cin.tie(0),0);

#define rep(i,a,b) for(int i=(a);i<=(b);i++)

#define mid ((l+r)>>1)

const int MAXN = 2e5+5;

int n,q,m,cnt=0;

int a[MAXN],b[MAXN];

int T[MAXN];

int sum[MAXN<<5];

int L[MAXN<<5],R[MAXN<<5];

int build(int l,int r){

int pos = ++cnt;

sum[pos]=0;

if(l<r){

L[pos]=build(l,mid);

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

}

return pos;

}

int update(int pre,int l,int r,int x){

int pos = ++cnt;

L[pos]=L[pre];R[pos]=R[pre];

sum[pos] = sum[pre]+1;

if(l<r){

if(x<=mid) L[pos] = update(L[pre],l,mid,x);

else R[pos] = update(R[pre],mid+1,r,x);

}

return pos;

}

int query(int u,int v,int l,int r,int k){

if(l>=r) return l;

int x = sum[L[v]]-sum[L[u]];

if(x>=k) return query(L[u],L[v],l,mid,k);

else return query(R[u],R[v],mid+1,r,k-x);

}

int main(){

cin>>n>>q;

rep(i,1,n){

cin>>a[i];

b[i]=a[i];

}

sort(b+1,b+1+n);

m = unique(b+1,b+1+n)-b-1;

T[0] = build(1,m);

rep(i,1,n){

int t = lower\_bound(b+1,b+1+m,a[i]) - b;

T[i] = update(T[i-1],1,m,t);

}

while(q--){

int u,v,k;

cin>>u>>v>>k;

int t = query(T[u-1],T[v],1,m,k);

cout<<b[t]<<endl;

}

}

/\*

==================================+

INPUT: |

5 5

25957 6405 15770 26287 26465

2 2 1

3 4 1

4 5 1

1 2 2

4 4 1

==================================+

OUTPUT: |

6405

15770

26287

25957

26287

==================================+

\*/

1. KHOẢNG CÁCH XA NHẤT GIỮA 2 CẶP ĐỈNH TRONG 1 CÂY (TREE DIAMETER)

#include<bits/stdc++.h>

using namespace std;

#define MAXN 50010

int n;

vector<int> e[MAXN];

int dfs(int prev,int u,int \*d){

int res = u;

int siz = e[u].size();

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

int v = e[u][i];

if(v!=prev){

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

int now = dfs(u,v,d);

if(d[now]>d[res]) res = now;

}

}

return res;

}

int main(){

int u,v;

int dx[MAXN];

int dy[MAXN];

scanf("%d",&n);

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

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

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

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

}

dy[1]=0;

int x = dfs(-1,1,dy);

dx[x]=0;

int y = dfs(-1,x,dx);

dy[y]=0;

dfs(-1,y,dy);

cout<<dx[y]<<endl;

}

/\*

==================================+

INPUT: |

5

1 2

2 3

3 4

4 5

==================================+

OUTPUT: |

4

==================================+

\*/

1. CHO 1 MẢNG GỒM N PHẦN TỬ VÀ 3 TRUY VẤN: S X Y – UPDATE GIÁ TRỊ TẠI VỊ TRÍ THỨ X THÀNH Y , R X Y – ĐẢO NGƯỢC MẢNG CON ĐOẠN [X,Y] ; Q X Y – IN RA TỔNG CÁC SỐ TỪ [X,Y] (SQPLAY TREE)

#include<bits/stdc++.h>

using namespace std;

#define long long long

#define f1(i,n) for (int i=1; i<=n; i++)

#define f0(i,n) for (int i=0; i<n; i++)

struct node {

int Size, Sum, Value;

node \*ll, \*rr;

bool Reverse;

node(node\* L, int AValue, node\* R)

{ ll=L, rr=R; Sum=Value=AValue; Size=1, Reverse=false; }

} \*Nil, \*Root;

node\* access(node\* X){

if (X==Nil) return NULL;

if (X->Reverse){

swap(X->ll, X->rr);

if (X->ll!=Nil) X->ll->Reverse^=1;

if (X->rr!=Nil) X->rr->Reverse^=1;

X->Reverse=false;

}

return X;

}

node\* update(node\* X){

if (!access(X)) return NULL;

X->Size=X->ll->Size+X->rr->Size+1;

X->Sum=X->ll->Sum+X->rr->Sum+X->Value;

return X;

}

ostream& operator << (ostream& cout, node\* X){

if (!access(X)) return cout;

return cout << "(" << X->ll << X->Value << X->rr << ")";

}

node\* create(int ll, int rr, int a[]){

if (ll>rr) return Nil;

int mm = (ll+rr)/2;

return update(new node(create(ll, mm-1, a), a[mm], create(mm+1, rr, a)));

}

// Splay, Split, Merge

stack<node\*> Updates;

void update\_now(stack<node\*> &st=Updates)

{ while (st.size()) { update(st.top()); st.pop(); } }

node\* link(node\* L, node\* X, node\* R)

{ X->ll=L, X->rr=R, update(X); return X; }

node\* llzig(node\* X, node\* L)

{ return link(L->ll, L, link(L->rr, X, X->rr)); }

node\* rrzig(node\* X, node\* R)

{ return link(link(X->ll, X, R->ll), R, R->rr); }

node\* get\_splay(node\* X, int Index){

node Header(Nil, 0, Nil);

node \*Left=&Header, \*Right=&Header;

while (access(X)->ll->Size != Index) {

if (access(X->ll) && Index<X->ll->ll->Size)

{ X=llzig(X, X->ll); continue; }

if (access(X->rr) && Index>X->ll->Size+1+X->rr->ll->Size+1)

{ X=rrzig(X, X->rr); continue; }

if (Index < X->ll->Size){

node \*P=X; X=X->ll;

P->ll=Nil, update(P);

Right->ll=P, Updates.push(Right);

Right=Right->ll;

}

else {

Index -= X->ll->Size+1;

node \*P=X; X=X->rr;

P->rr=Nil, update(P);

Left->rr=P, Updates.push(Left);

Left=Left->rr;

}

}

Left->rr=X->ll, Updates.push(Left);

Right->ll=X->rr, Updates.push(Right);

X->ll=Nil, X->rr=Nil, update(X);

update\_now();

X->ll=Header.rr, X->rr=Header.ll, update(X);

return X;

}

void split(node\* X, int Index, node\* &L, node\* &R){

if (Index==X->Size) { L=X, R=Nil; return; }

X=access(get\_splay(X, Index));

L=X->ll, R=X;

X->ll=Nil, update(X);

}

node\* split(node\* &X, int Index){

node \*L, \*R;

split(X, Index, L, R);

return X=L, R;

}

node\* merge(node\* L, node\* R){

if (R==Nil) return L;

R=access(get\_splay(R, 0));

return R->ll=L, update(R);

}

#define N 100005

int n, m, a[N];

main(){

Root = Nil = new node(0, 0, 0);

Nil->Size=0; Nil->ll=Nil->rr=Nil;

scanf("%d", &n);

f1(i,n) scanf("%d", &a[i]);

Root = create(1, n, a);

// cout << Root << endl;

scanf("%d", &m);

while (m--){

int x, y; char c;

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

if (c=='S') {

Root=get\_splay(Root, x-1);

if (access(Root)) Root->Value=y, update(Root);

}

else if (c=='R') {

node\* R = split(Root, y);

node\* M = split(Root, x-1);

if (access(M)) M->Reverse=true, update(M);

Root = merge(Root, merge(M, R));

}

else if (c=='Q') {

node\* R = split(Root, y);

node\* M = split(Root, x-1);

if (access(M)) cout << M->Sum << endl;

Root = merge(Root, merge(M, R));

}

// cout << Root << endl;

}

}

/\*

==================================+

INPUT: |

5

1 2 3 4 5

4

S 1 10

R 1 5

Q 1 2

Q 2 3

==================================+

OUTPUT: |

9

7

==================================+

\*/

1. TÌM CÂY KHUNG NHỎ NHẤT (TỨC LÀ ĐỒ THỊ ĐI QUA TẤT CẢ CÁC ĐỈNH VÀ CÓ TỔNG CHI PHÍ BÉ NHẤT)

**if(**d**[**v**]<** d**[**u**])** FORE**(**i**,**17**,**0**)if(**d**[**parent**[**u**][**i**]]>=** d**[**v**])** u **=** parent**[**u**][**i**];**

#include<bits/stdc++.h>

#define rep(i,a,b) for(int i=(a);i<=(b);i++)

#define ll long long

using namespace std;

const int MAXN = 5050;

const int MAXM = 2e5+5;

struct Edge{

int u,v,k;

friend bool operator < (Edge a,Edge b){

return a.k<b.k;

}

}edges[MAXM];

int fa[MAXN];

int rankk[MAXN];

int find(int x){

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

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

}

void merge(int a,int b){

a = find(a);b = find(b);

if(a==b) return;

if(rankk[a]>rankk[b]) fa[b] = a;

else{

fa[a] = b;

if(rankk[a]==rankk[b])rankk[b]++;

}

}

int n,m;

int main(){

cin>>n>>m;

rep(i,1,m) cin>>edges[i].u>>edges[i].v>>edges[i].k;

rep(i,1,n) fa[i] = i,rankk[i] = 0;

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

ll res=0;

rep(i,1,m){

if(find(edges[i].u)!=find(edges[i].v)){

res+=edges[i].k;

merge(edges[i].u,edges[i].v);

}

}

bool flag = 1;

rep(i,2,n){

if(find(n)!=find(n-1)) {flag = 0;break;}

}

if(!flag) cout<<"orz"<<endl;

else cout<<res<<endl;

}

/\*

==================================+

INPUT: |

4 5

1 2 2

1 3 2

1 4 3

2 3 4

3 4 3

==================================+

OUTPUT: |

7

==================================+

\*/

1. THUẬT TOÁN HUNGARIAN : N NGƯỜI BÁN, M NGƯỜI MUA. 1 NGƯỜI BÁN CHỈ ĐƯỢC KẾT HỢP VỚI 1 NGƯỜI MUA VÀ NGƯỢC LẠI (CÓ CHI PHÍ). IN RA TỔNG CHI PHÍ LỚN NHẤT.

#include<iostream>

#include<cstdio>

#include<cstring>

#include<queue>

using namespace std;

const int maxn=510;

const int maxm=100010;

const int inf=0x3f3f3f3f;

struct Node

{

int to;

int capa;

int cost;

int next;

}edge[maxm];

int cnt;

int source,sink;

int head[maxn];

int num[maxn];

int dis[maxn];

int rec[maxn];

int pre[maxn];

bool vis[maxn];

void init()

{

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

memset(num,0,sizeof(num));

cnt=0;

}

void add(int u,int v,int capa,int cost)

{

edge[cnt].to=v;

edge[cnt].capa=capa;

edge[cnt].cost=cost;

edge[cnt].next=head[u];

head[u]=cnt++;

edge[cnt].to=u;

edge[cnt].capa=0;

edge[cnt].cost=-cost;

edge[cnt].next=head[v];

head[v]=cnt++;

return;

}

bool spfa()

{

memset(dis,inf,sizeof(dis));

memset(vis,false,sizeof(vis));

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

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

queue<int> que;

que.push(source);

dis[source]=0;

vis[source]=true;

while(!que.empty())

{

int node=que.front();

que.pop();

vis[node]=false;

for(int i=head[node];~i;i=edge[i].next)

{

int v=edge[i].to;

if(edge[i].capa>0&&dis[v]>dis[node]+edge[i].cost)

{

dis[v]=dis[node]+edge[i].cost;

rec[v]=i;

pre[v]=node;

if(!vis[v])

{

vis[v]=true;

que.push(v);

}

}

}

}

return dis[sink]!=inf;

}

int mcmf()

{

int maxflow=0;

int mincost=0;

while(spfa())

{

int flow=inf;

int node=sink;

while(node!=source)

{

flow=min(flow,edge[rec[node]].capa);

node=pre[node];

}

maxflow+=flow;

node=sink;

while(node!=source)

{

mincost+=flow\*edge[rec[node]].cost;

edge[rec[node]].capa-=flow;

edge[rec[node]^1].capa+=flow;

node=pre[node];

}

}

return mincost;

}

int main()

{

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

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

int test;

scanf("%d",&test);

while(test--)

{

init();

int n,m;

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

int u,v,w;

source=0;

sink=210;

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

{

add(source,i,1,0);

}

while(~scanf("%d%d%d",&u,&v,&w))

{

if(!u&&!v&&!w) break;

add(u,v+n,1,-w);

}

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

{

add(n+i,sink,1,0);

}

printf("%d\n",-mcmf());

}

return 0;

}

/\*

==================================+

INPUT:

3

3 2

1 1 10

2 1 19

2 2 11

3 2 1

0 0 0

4 4

1 1 6

1 2 6

2 1 8

2 3 9

2 4 8

3 2 8

4 3 7

0 0 0

3 2

1 1 10

2 1 21

2 2 11

3 2 1

0 0 0

==================================+

OUTPUT: |

21

29

22

==================================+

\*/

1. THUẬT TOÁN DIJKSTRA – SỬ DỤNG PRIORITY QUEUE – TÌM ĐƯỜNG ĐI NGẮN NHẤT TỪ 1 ĐỈNH ĐẾN TẤT CẢ CÁC ĐỈNH CÒN LẠI.

#include<bits/stdc++.h>

using namespace std;

template<class T>inline void read(T &x){x=0;char o,f=1;while(o=getchar(),o<48)if(o==45)f=-f;do x=(x<<3)+(x<<1)+(o^48);while(o=getchar(),o>47);x\*=f;}

int cansel\_sync=(ios::sync\_with\_stdio(0),cin.tie(0),0);

#define ll long long

#define rep(i,a,b) for(int i=(a);i<=(b);i++)

#define repb(i,a,b) for(int i=(a);i>=b;i--)

#define INF 0x3f3f3f3f

#define cendl printf("\n")

ll gcd(ll a,ll b){ while(b^=a^=b^=a%=b); return a; }

//#define INF 0x7fffffff

#define LINF 1ll<<60

const int MAXN = 2e5+10;

typedef pair<int,ll> pli;

vector<pli> e[MAXN];

bool vis[MAXN];

ll d[MAXN];

int n,m,s,t;

void dijkstra(){

rep(i,1,n){

d[i] = LINF;

vis[i] = 0;

}

d[s] = 0;

priority\_queue< pli,vector<pli>,greater<pli> >q;

q.push(make\_pair(0,s));

while(!q.empty()){

int now = q.top().second;

q.pop();

if(vis[now])continue;

vis[now] = 1;

int siz = e[now].size();

for(auto x:e[now]){

int v = x.second;

if(d[v]>d[now]+x.first){

d[v] = d[now] + x.first;

q.push(make\_pair(d[v],v));

}

}

}

}

int main(){

cin>>n>>m>>s;

int u,v;

ll dis;

rep(i,1,m){

cin>>u>>v>>dis;

e[u].push\_back(make\_pair(dis,v));

}

dijkstra();

rep(i,1,n){

cout<<d[i];

if(i!=n) cout<<' ';

}

}

/\*

==================================+

INPUT:

3 3 1

1 2 1

1 3 5

2 3 1

==================================+

OUTPUT: |

0 1 2

==================================+

\*/

1. CHO N NHIỆM VỤ VÀ M MỐI QUAN HỆ. MỖI QUAN HỆ CÓ DẠNG U V - THỂ HIỆN NHIỆM VỤ U ĐƯỢC THỰC HIỆN TRƯỚC NHIỆM VỤ V. IN RA THỨ TỰ CÁC NHIỆM VỤ CẦN LÀM (TOPOSORT)

#include<bits/stdc++.h>

using namespace std;

template<class T>inline void read(T &x){x=0;char o,f=1;while(o=getchar(),o<48)if(o==45)f=-f;do x=(x<<3)+(x<<1)+(o^48);while(o=getchar(),o>47);x\*=f;}

int cansel\_sync=(ios::sync\_with\_stdio(0),cin.tie(0),0);

#define ll long long

#define rep(i,a,b) for(int i=(a);i<=(b);i++)

#define repb(i,a,b) for(int i=(a);i>=b;i--)

#define INF 0x3f3f3f3f

#define cendl printf("\n")

ll gcd(ll a,ll b){ while(b^=a^=b^=a%=b); return a; }

//#define INF 0x7fffffff

const int MAXN = 110;

vector<int> e[MAXN];

int n,m;

int indg[MAXN];

void solve(){

for(int i=1;i<=n;i++) {indg[i]=0;e[i].clear();}

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

cin>>u>>v;

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

indg[v]++;

}

queue<int> q;

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

if(indg[i]==0) q.push(i);

vector<int> res;

int u;

while(!q.empty()){

u = q.front();

q.pop();

res.push\_back(u);

for(auto v:e[u]){

if(--indg[v]==0)q.push(v);

}

}

if(res.size()==n){

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

cout<<res[i]<<' ';

}

cout<<endl;

}

}

int main(){

while(cin>>n>>m&&!(n==0&&m==0)) solve();

}

/\*

==================================+

INPUT:

5 4

1 2

2 3

1 3

1 5

0 0

==================================+

OUTPUT: |

1 4 2 5 3

==================================+

\*/

1. ĐẾM THÀNH PHẦN LIÊN THÔNG MẠNH – TARJAN

#include<bits/stdc++.h>

using namespace std;

template<class T>inline void read(T &x){x=0;char o,f=1;while(o=getchar(),o<48)if(o==45)f=-f;do x=(x<<3)+(x<<1)+(o^48);while(o=getchar(),o>47);x\*=f;}

int cansel\_sync=(ios::sync\_with\_stdio(0),cin.tie(0),0);

#define ll long long

#define rep(i,a,b) for(int i=(a);i<=(b);i++)

#define repb(i,a,b) for(int i=(a);i>=b;i--)

#define INF 0x3f3f3f3f

#define cendl printf("\n")

ll gcd(ll a,ll b){ while(b^=a^=b^=a%=b); return a; }

//#define INF 0x7fffffff

const int MAXN = 2e6+5;

int n;

vector<int> e[MAXN];

int dfn[MAXN],low[MAXN];

int dfncnt;

int s[MAXN],tp;

bool in[MAXN];

int scc[MAXN],sc;

int sz[MAXN];

int indg[MAXN];

void tarjan(int u){

low[u]=dfn[u]=++dfncnt;

s[++tp]=u;

in[u]=1;

for(auto v:e[u]){

if(!dfn[v]){

tarjan(v);

low[u] = min(low[u],low[v]);

}

else if(in[v])

low[u] = min(low[u],dfn[v]);

}

if(dfn[u]==low[u]){

sc++;

sz[sc] = 0;

while(s[tp]!=u){

scc[s[tp]] = sc;

sz[sc]++;

in[s[tp]] = 0;

tp--;

}

scc[u] = sc;

sz[sc]++;

in[u] = 0;

tp--;

}

}

void reset(){

tp = sc = dfncnt =0;

rep(i,1,n){

in[i] = dfn[i] = 0;

e[i].clear();

}

}

int main(){

cin>>n;

reset();

int v;

rep(u,1,n){

while(cin>>v&&v!=0) e[u].push\_back(v);

}

rep(u,1,n)

if(!dfn[u]) tarjan(u);

rep(i,1,sc) indg[i] = 0;

rep(u,1,n){

for(auto v:e[u]){

if(scc[u]!=scc[v]) indg[scc[v]]++;

}

}

int res = 0;

rep(i,1,sc){

if(indg[i]==0) res++;

}

cout<<res<<endl;

}

/\*

==================================+

INPUT:

5

2 3 4 0

4 5 0

0

0

1 0

==================================+

OUTPUT: |

1 4 2 5 3

==================================+

\*/

1. LUỒNG CỰC ĐẠI

#include<bits/stdc++.h>

using namespace std;

#define MAXN 1005

#define INF 0x7fffffff

#define rep(i,a,b) for(int i=(a);i<=(b);i++)

#define ll long long

struct Edge{

int from,to;

ll cap,flow;

};

int n,m,s,t;

vector<Edge> edges;

vector<int> G[MAXN];

int d[MAXN];

int cur[MAXN];

void add\_edge(int from,int to,ll cap){

edges.push\_back((Edge){from,to,cap,0});

edges.push\_back((Edge){to,from,0,0});

int siz = edges.size();

G[from].push\_back(siz-2);

G[to].push\_back(siz-1);

}

bool bfs(){

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

queue<int> q;

q.push(s);

d[s]=0;//dist(s) = 0

while(!q.empty()){

int x=q.front();

q.pop();

rep(i,0,G[x].size()-1){

Edge &e = edges[G[x][i]];

if(d[e.to]==-1&&e.cap>e.flow){

d[e.to]=d[x]+1;

q.push(e.to);

}

}

}

return d[t]!=-1;

}

ll dfs(int x,ll a){

if(x==t||a==0) return a;

ll flow = 0,f;

for(int &i=cur[x];i<G[x].size();i++){

Edge &e = edges[G[x][i]];

if(d[x]+1==d[e.to]&&(f=dfs(e.to,min(a,e.cap-e.flow)))>0){

e.flow+=f;

edges[G[x][i]^1].flow-=f;

flow+=f;

a-=f;

if(a==0) break;

}

}

return flow;

}

ll Maxflow(){

ll flow=0;

while(bfs()){

memset(cur,0,sizeof(cur));

flow+=dfs(s,INF);

}

return flow;

}

int main(){

int u,v;

ll cap;

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

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

scanf("%d%d%lld",&u,&v,&cap);

add\_edge(u,v,cap);

}

cout<<Maxflow()<<endl;

}

/\*

==================================+

INPUT:

4 5 4 3

4 2 30

4 3 20

2 3 20

2 1 30

1 3 40

==================================+

OUTPUT: |

50

==================================+

\*/

1. TRIE

#include <stdio.h>

#include <vector>

using namespace std;

// trie

class trie {

public :

struct node {

int a[64];

int value;

int& operator[] (int i){ return a[i%64]; }

node() { for (int i=0; i<64; i++) a[i]=0; value=0; }

};

vector <node> a;

int& operator[] (char \*s){

int pos=0, i, c;

for (i=0; c=s[i]; i++)

{

if (a[pos][c]==0) {

a.push\_back(node());

a[pos][c] = a.size()-1;

}

pos=a[pos][c];

}

return a[pos].value;

}

void clear(){ a.clear(); a.push\_back(node()); }

trie(){ clear(); }

};

trie tr;

// main

main(){

int cnt=0;

char s[2309];

for(;;){

gets(s);

if (tr[s]==0) tr[s]=++cnt;

printf("%s = %d\n", s, tr[s]);

}

}

/\*

==================================+

INPUT:

DDat

TuanGay

DDat

LinhChoCon

==================================+

OUTPUT: |

1

2

1

3

==================================+

\*/

1. RMQ: CHO MỘT DÃY A GỒM N SỐ NGUYÊN VÀ K CẶP SỐ U V . VỚI MỖI CẶP SỐ (U,V) IN RA SỐ LỚN NHẤT TRONG KHOẢNG A[U..V]

#include <math.h>

#include <stdio.h>

int a[230997];

int M[230997][23];

int n, m;

int max(int a, int b) { return a > b ? a : b; }

int main() {

int i, k;

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

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

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

for (i = 1; i <= n; i++) M[i][0] = a[i];

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

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

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

int u, v;

while (m--) {

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

k = log2(v - u + 1);

printf("%d\n", max(M[u][k], M[v - (1 << k) + 1][k]));

}

}

/\*

==================================+

INPUT:

5 3

3 5 6 1 2

1 3

3 5

2 4

==================================+

OUTPUT: |

6

6

6

==================================+

\*/

1. KHỬ GAUSS (KIỂM TRA HỆ PHƯƠNG TRÌNH N ẨN CÓ NGHIỆM HAY KHÔNG VÀ IN RA)

#include<bits/stdc++.h>

using namespace std;

#define rep(i,a,b) for(int i=(a);i<=(b);i++)

#define repb(i,a,b) for(int i=(a);i>=b;i--)

const int MAXN = 1004;

double a[MAXN][MAXN];

int n,pl;

int main(){

scanf("%d",&n);

rep(i,1,n)

rep(j,1,n+1) scanf("%lf",&a[i][j]);

rep(i,1,n){

pl = i;

while(a[pl][i]==0&&pl<=n) pl++;

if(pl==n+1){

cout<<"No Solution"<<endl;return 0;

}

rep(j,1,n+1) swap(a[i][j],a[pl][j]);

double k = a[i][i];

rep(j,1,n+1) a[i][j]/=k;

rep(ii,1,n){

if(ii==i) continue;

double ki = a[ii][i];

rep(m,1,n+1) a[ii][m]-=ki\*a[i][m];

}

}

//rep(i,1,n) printf("%.2lf\n",a[i][n+1]);

rep(i,1,n) cout<<fixed<<setprecision(2)<<a[i][n+1]<<'\n';

}

/\*

==================================+

INPUT:

3

1 3 4 5

1 4 7 3

9 3 2 2

==================================+

OUTPUT: |

-0.97

5.18

-2.39

==================================+

\*/

1. TÍNH C(N,M) MOD P .

#include<iostream>

using namespace std;

#define rep(i,a,b) for(int i=(a);i<=(b);i++)

#define ll long long

const int MAXN = 3e5+5;

const int med = 998244353;

ll jc[MAXN];

ll qpow(ll d,ll c){

ll res = 1;

while(c){

if(c&1) res=res\*d%med;

d=d\*d%med;c>>=1;

}return res;

}

inline ll niyuan(ll x){return qpow(x,med-2);}

void initjc(){

jc[0] = 1;

rep(i,1,MAXN-1) jc[i] = jc[i-1]\*i%med;

}

inline int C(int n,int m){

if(n<m) return 0;

return jc[n]\*niyuan(jc[n-m])%med\*niyuan(jc[m])%med;

}

int main(){

initjc();

int n,m;

while(cin>>n>>m) cout<<C(n,m)<<endl;

}

/\*

==================================+

INPUT:

5 2

5 3

0 0

==================================+

OUTPUT: |

10

10

==================================+

\*/

1. ĐẾM NGHIỆM NGUYÊN DƯƠNG CỦA PHƯƠNG TRÌNH: AX+BY+CZ=D (SỬ DỤNG EXTEND-EULER)

#include <stdio.h>

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

// ax + by = gcd(x,y)

int flag = 0;

long long t, la = 1, lb = 0, ra = 0, rb = 1;

while(x%y) {

if(flag == 0)

la -= x/y\*ra, lb -= x/y\*rb;

else

ra -= x/y\*la, rb -= x/y\*lb;

t = x, x = y, y = t%y;

flag = 1 - flag;

}

if(flag == 0)

a = ra, b = rb;

else

a = la, b = lb;

return y;

}

long long countSolution(long long n1, long long n2, long long n) {

long long a, b, g;

g = exgcd(n1, n2, a, b); // a\*n1 + b\*n2 = gcd(n1,2)

if(n%g) return 0;

long long k = n/g, k1, k2;

a \*= k, b \*= k;// a\*n1 + b\*n2 = n

// (a+F)\*n1 + (b+G)\*n2 = n => Fn1 + Gn2 = 0,

//F = lcm(n1, n2)/n1 \* i, G = lcm(n1, n2)/n2 \* i

k1 = n1\*n2/g/n1, k2 = n1\*n2/g/n2;

if(a < 0) { // adjust a >= 0

k = -(a/k1) + (a%k1 != 0);

a += k\*k1, b -= k\*k2;

}

if(b < 0) { // adjust b >= 0

k = -(b/k2) + (b%k2 != 0);

a -= k\*k1, b += k\*k2;

}

if(a < 0 || b < 0) return 0;

long long x1, x2, y1, y2;

// minimize a, maximize b

k = a/k1;

a -= k\*k1;

b += k\*k2;

x1 = a, y1 = b;

// maximize a, minimize b

k = b/k2;

a += k\*k1;

b -= k\*k2;

x2 = a, y2 = b;

return (x2 - x1) / k1 + 1;

}

int main() {

int testcase, cases = 0;

long long A, B, C, P, ret, ta, tb;

scanf("%d", &testcase);

while (testcase--) {

scanf("%lld %lld %lld %lld", &A, &B, &C, &P);

long long g = exgcd(exgcd(A, B, ta, tb), C, ta, tb);

P /= g, A /= g, B /= g, C /= g;

ret = 0;

for (long long i = 0; P - C \* i >= 0; i++)

ret += countSolution(A, B, P - C \* i);

printf("Case %d: %lld\n", ++cases, ret);

}

return 0;

}

/\*

1

202 203 200 606

\*/

/\*

==================================+

INPUT:

1

202 203 200 606

==================================+

OUTPUT: |

Case 1: 2

==================================+

\*/

1. HÀM PHI-EULER

int euler\_phi(int n){

int sqr = sqrt(n+0.5);

int res = n;

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

if(n%i==0){

res = res/i\*(i-1);

while(n%i==0) n/=i;

}

}

if(n>1) res = res/n\*(n-1);

return res;

}

1. INVERSE ELEMENT(TÍNH A^-1 MOD P)

void init(int p){

inv[1] = 1;

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

inv[i] = (ll)(p-p/i)\*inv[p%i]%p;

}

}

1. ĐỊNH LÝ THẶNG DƯ TRUNG HOA – TÌM X NHỎ NHẤT THỎA MÃN X MOD A1 = B1 ; X MOD A2 = B2;…;X MOD AN = BN

#include<iostream>

using namespace std;

#define ll long long

#define rep(i,a,b) for(int i=(a);i<=(b);i++)

template<class T>inline void read(T &x){x=0;char o,f=1;while(o=getchar(),o<48)if(o==45)f=-f;do x=(x<<3)+(x<<1)+(o^48);while(o=getchar(),o>47);x\*=f;}

template<class T>

void wt(T x){

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

if(x >= 10) wt(x / 10);

putchar('0' + x % 10);

}

void Exgcd(ll a,ll b,ll &d,ll &x,ll &y){

if(!b){d=a;x=1;y=0;}

else{Exgcd(b,a%b,d,y,x);y-=x\*(a/b);}

}

inline ll mul(ll a,ll b,ll mo){

return ((a%mo)\*(b%mo)%mo+mo)%mo;

}

const int MAXN = 1e5+5;

ll c[MAXN],m[MAXN];

int n;

ll Excrt(ll m[],ll c[],int n){

ll mnow = m[1],cnow = c[1];

rep(i,2,n){

ll p1,p2,gcdd;

ll m1 = mnow,m2 = m[i];

ll dc = (c[i]-cnow%m2+m2)%m2;

Exgcd(m1,m2,gcdd,p1,p2);

if(dc%gcdd) {cout<<i<<endl;return -1;}

p1 = mul(p1,dc/gcdd,m2/gcdd);

cnow += p1\*m1;

mnow = m1/gcdd\*m2;

cnow = (cnow%mnow+mnow)%mnow;

}

return cnow;

}

int main(){

cin>>n;

rep(i,1,n) read(m[i]),read(c[i]);

wt(Excrt(m,c,n));

}

/\*

==================================+

INPUT:

// ai bi

3

11 6

25 9

33 17

==================================+

OUTPUT: |

809

==================================+

\*/

1. CHO 3 ĐIỂM, TÌM TÂM ĐƯỜNG TRÒN NGOẠI TIẾP

#include<bits/stdc++.h>

using namespace std;

struct point{

double x;

double y;

};

point cal(point a,point b,point c){

double x1 = a.x;double y1 = a.y;

double x2 = b.x;double y2 = b.y;

double x3 = c.x; double y3 = c.y;

double a1 = 2\*(x2-x1); double a2 = 2\*(x3-x2);

double b1 = 2\*(y2-y1); double b2 = 2\*(y3-y2);

double c1 = x2\*x2 + y2\*y2 - x1\*x1 - y1\*y1;

double c2 = x3\*x3 + y3\*y3 - x2\*x2 - y2\*y2;

double rx = (c1\*b2-c2\*b1)/(a1\*b2-a2\*b1);

double ry = (c2\*a1-c1\*a2)/(a1\*b2-a2\*b1);

return point{rx,ry};

}

int main(){

point fi,se,th;

cin>>fi.x>>fi.y;

cin>>se.x>>se.y;

cin>>th.x>>th.y;

point res;

res = cal(fi,se,th);

cout<<res.x<<" "<<res.y;

return 0;

}

/\*

==================================+

INPUT:

0 0

1 0

0 1

==================================+

OUTPUT: |

0.5 0.5

==================================+

\*/

1. SLIDING WINDOW (CỬA SỐ TRƯỢT)

ll EulerPhi(ll N)

#include<iostream>

#include<deque>

using namespace std;

struct node{

int data;

int order;

};

const int MAXN = 1e6+5;

deque<node>dq\_min;

deque<node>dq\_max;

int n,k;

int a[MAXN];

int res\_min[MAXN];

int res\_max[MAXN];

int main(){

cin>>n>>k;

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

cin>>a[i];

}

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

if(!dq\_max.empty()&&dq\_max.front().order<i-k+1) dq\_max.pop\_front();

if(!dq\_min.empty()&&dq\_min.front().order<i-k+1) dq\_min.pop\_front();

while(!dq\_max.empty()&&dq\_max.back().data<=a[i]) dq\_max.pop\_back();

dq\_max.push\_back(node{a[i],i});

while(!dq\_min.empty()&&dq\_min.back().data>=a[i]) dq\_min.pop\_back();

dq\_min.push\_back(node{a[i],i});

res\_max[i] = dq\_max.front().data;

res\_min[i] = dq\_min.front().data;

}

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

cout<<res\_min[i];

if(i!=n) cout<<' ';

}

cout<<endl;

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

cout<<res\_max[i];

if(i!=n) cout<<' ';

}

cout<<endl;

}

/\* Hai chieu\*/

/\*

deque<node> dq;

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

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

if(!dq.empty()&&dq.front().order<j-k+1) dq.pop\_front();

while(!dq.empty()&&a[i][j]>=dq.back().data) dq.pop\_back();

dq.push\_back(node{a[i][j],j});

res[i][j] = dq.front().data;

}

dq.clear();

}

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

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

if(!dq.empty()&&dq.front().order<i-k+1) dq.pop\_front();

while(!dq.empty()&&dq.back().data<=res[i][j]) dq.pop\_back();

dq.push\_back(node{res[i][j],i});

res[i][j] = dq.front().data;

}

dq.clear();

}

ll msum = 0;

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

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

msum+=res[i][j];

}

}

cout<<msum<<endl;

\*/

/\*

==================================+

INPUT:

8 3

1 3 -1 -3 5 3 6 7

==================================+

OUTPUT: |

-1 -3 -3 -3 3 3

3 3 5 5 6 7

==================================+

\*/

1. CÓ BAO NHIÊU XÂU CON (LIÊN TỤC) KHÁC NHAU XUẤT HIỆN TRONG XÂY T(AC AUTOMATA)

#include<bits/stdc++.h>

using namespace std;

int cansel\_sync=(ios::sync\_with\_stdio(0),cin.tie(0),0);

#define rep(i,a,b) for(int i=(a);i<=(b);i++)

const int MAXN = 1e6+5;

inline int idx(char c){return c-'a';}

struct Node{

int son[26],flag,fail;

}trie[MAXN\*10];

int n,cntt;

string s;

queue<int> q;

void insert(string &s){

int siz = s.size(),v,u = 1;

rep(i,0,siz-1){

v = idx(s[i]);

if(!trie[u].son[v]) trie[u].son[v] = ++cntt;

u = trie[u].son[v];

}

trie[u].flag++;

}

void getfail(){

rep(i,0,25) trie[0].son[i] = 1;

q.push(1);trie[1].fail = 0;

int u,v,ufail;

while(!q.empty()){

u = q.front();q.pop();

rep(i,0,25){

v = trie[u].son[i];

ufail = trie[u].fail;

if(!v){trie[u].son[i]=trie[ufail].son[i];continue;}

trie[v].fail = trie[ufail].son[i];

q.push(v);

}

}

}

int query(string &s){

int siz = s.size(),u = 1,v,k,ans = 0;

rep(i,0,siz-1){

v = idx(s[i]);

k = trie[u].son[v];

while(k&&trie[k].flag!=-1){

ans += trie[k].flag;trie[k].flag = -1;

k = trie[k].fail;

}

u = trie[u].son[v];

}

return ans;

}

int main(){

cntt = 1;

cin>>n;

string hc;

rep(i,1,n){

cin>>s;insert(s);

}

getfail();

cin>>s;

cout<<query(s)<<endl;

}

/\*

==================================+

INPUT:

4

a

from

ab

abc

abcd

==================================+

OUTPUT: |

2

==================================+

\*/

1. CHO MẢNG A GỒM N PHẦN TỬ, CÁC PHẦN TỬ THUỘC ĐOẠN [1;K]. CÓ M TRUY VẤN, MỖI TRUY VẤN CÓ DẠNG L R – TÍNH TỔNG SIGMA(I=1,K) C\_I^2 . TRONG ĐÓ C\_I LÀ SỐ LẦN XUẤT HIỆN CỦA I TRONG ĐOẠN [L;R]

#include<bits/stdc++.h>

using namespace std;

#define rep(i,a,b) for(int i=(a);i<=(b);i++)

#define ll long long

const int MAXN = 5e4+5;

int cnt[MAXN];

int pos[MAXN],a[MAXN];

ll ans[MAXN];

int n,m,k,res;

struct Q{

int l,r,k;

friend bool operator < (Q x,Q y){

return pos[x.l]==pos[y.l]?x.r<y.r:pos[x.l]<pos[y.l];

}

}q[MAXN];

void Add(int pos){

res -= cnt[a[pos]]\*cnt[a[pos]];

cnt[a[pos]]++;

res += cnt[a[pos]]\*cnt[a[pos]];

}

void Sub(int pos){

res -= cnt[a[pos]]\*cnt[a[pos]];

cnt[a[pos]]--;

res += cnt[a[pos]]\*cnt[a[pos]];

}

int main(){

cin>>n>>m>>k;

memset(cnt,0,sizeof(cnt));

int siz = sqrt(k);

rep(i,1,n){

cin>>a[i];

pos[i] = i/siz;

}

rep(i,1,m){

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

q[i].k = i;

}

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

res = 0;

int l = 1,r = 0;

rep(i,1,m){

while(q[i].l<l) Add(--l);

while(q[i].l>l) Sub(l++);

while(q[i].r<r) Sub(r--);

while(q[i].r>r) Add(++r);

ans[q[i].k] = res;

}

rep(i,1,m) cout<<ans[i]<<endl;

}

/\*

==================================+

INPUT:

// n m k

6 4 3

1 3 2 1 1 3

1 4

2 6

3 5

5 6

==================================+

OUTPUT: |

6

9

5

2

==================================+

\*/

1. CHO N CẶP ĐIỂM THỂ HIỆN N HÌNH CHỮ NHẬT (GÓC TRÁI DƯỚI VÀ GÓC PHẢI TRÊN). TÍNH DIỆN TÍCH ĐƯỢC BAO PHỦ BỞI CÁC HÌNH CHỮ NHẬT TRÊN.

#include<iostream>

#include<algorithm>

#include<vector>

using namespace std;

#define rep(i,a,b) for(int i=(a);i<=(b);i++)

#define ll long long

#define ls (x<<1)

#define rs (x<<1|1)

int cansel\_sync=(ios::sync\_with\_stdio(0),cin.tie(0),0);

const int MAXN = 2e5+5;

struct Line{

ll l,r,h;

int qz;

bool operator < (Line &rhs){

return h < rhs.h;

}

}line[MAXN];

int n;

ll x1,y1,x2,y2;

ll X[MAXN];

struct Segt{

int l,r;

int sum;

ll len;

}t[MAXN<<3];

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

t[x].l = l;t[x].r = r;

t[x].len = t[x].sum = 0;

if(l==r) return;

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

build(ls,l,mid);

build(rs,mid+1,r);

}

void push\_up(int x){

int l = t[x].l,r = t[x].r;

if(t[x].sum) t[x].len = X[r+1]-X[l];

else t[x].len = t[ls].len + t[rs].len;

}

void update(int x,int L,int R,int v){

int l = t[x].l,r = t[x].r;

if(X[r+1]<=L||R<=X[l]) return;

if(L<=X[l]&&X[r+1]<=R){

t[x].sum += v;

push\_up(x);

return;

}

update(ls,L,R,v);

update(rs,L,R,v);

push\_up(x);

}

int main(){

cin>>n;

rep(i,1,n){

cin>>x1>>y1>>x2>>y2;

X[2\*i-1] = x1,X[2\*i] = x2;

line[2\*i-1] = Line{x1,x2,y1,1};

line[2\*i] = Line{x1,x2,y2,-1};

}

n<<=1;

sort(line+1,line+1+n);

sort(X+1,X+1+n);

int tot = unique(X+1,X+n+1)-(X+1);

build(1,1,tot-1);

ll res = 0;

rep(i,1,n-1){

update(1,line[i].l,line[i].r,line[i].qz);

res += t[1].len\*(line[i+1].h-line[i].h);

}

cout<<res<<endl;

}

/\*

==================================+

INPUT:

2

1 1 2 2

2 2 3 3

==================================+

OUTPUT: |

6

9

5

2

==================================+

\*/

1. MILLER-RABIN TEST

ll nt[1000010];

ll nhan(ll a, ll b, ll mod)

{

if (b == 0) return 0%mod;

if (b == 1) return a%mod;

ll g = nhan(a,b/2,mod);

if (b%2) return ((g+g)%mod+a)%mod;

return (g+g)%mod;

}

ll modpow(ll a, ll b, ll mod)

{

if (b == 0) return 1%mod;

if (b == 1) return a%mod;

ll g = modpow(a,b/2,mod);

if (b%2) return nhan(nhan(g,g,mod),a,mod);

return nhan(g,g,mod);

}

bool MillerRabin(ll n, ll seed)

{

ll k = 0;

if (n < 2) return false;

if (n == 2) return true;

if (!(n & 1)) return false;

ll m = n - 1;

while (!(m & 1)) m >>= 1, k++;

ll a = seed;

a = modpow(a, m, n);

if (a == 1 || a == n - 1) return true;

for (ll j = 0; j < k - 1; j++)

{

a = modpow(a, 2, n);

if (a == 1) return false;

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

}

return false;

}

void Sieve()

{

FOR(i,2,1000000)

if (nt[i] == 0) {

nt[i] = 1;

for (ll j = i\*i; j <= 1000000; j += i) nt[j] = -1;

}

}

bool PrimalityTest(ll n)

{

if (n <= 1000000) return nt[n] == 1;

else return MillerRabin(n, 2) && MillerRabin(n, 13) && MillerRabin(n, 23) && MillerRabin(n, 1662803);

}

int main()

{

ll t, n, k;

ios::sync\_with\_stdio(0);

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

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

Sieve();

cin >> t;

while (t--) {

cin >> n >> k;

if (n < 2\*k) cout << "No\n";

else if (n == 2\*k) cout << "Yes\n";

else if (k == 1) cout << (PrimalityTest(n) ? "Yes" : "No") << endl;

else if (n % 2 == 0) cout << "Yes\n";

else {

if (n == 5) cout << "Yes\n";

else if (k == 2) cout << (PrimalityTest(n-2) ? "Yes" : "No") << endl;

else cout << "Yes\n";

}

}

return 0;

}

1. KMP

#include <stdio.h>

#include <string.h>

#include <algorithm>

#include <iostream>

using namespace std;

const int N = 1000006;

int m, n, Prev[N];

char a[N], b[N];

int main() {

scanf("%s%s", a + 1, b + 1);

m = strlen(a + 1), n = strlen(b + 1);

Prev[0] = -1;

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

Prev[i] = 0;

for (int j = Prev[i - 1]; j != -1; j = Prev[j])

if (b[j + 1] == b[i]) {

Prev[i] = j + 1;

break;

}

}

int u = 0;

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

while (u != 0 && b[u + 1] != a[i]) u = Prev[u];

if (b[u + 1] == a[i]) u++;

if (u == n) printf("%d ", i - n + 1);

}

puts("");

}

/\*

Input:

abab

ab

Output:

1

3

\*/

1. ĐẾM KHỚP VÀ CẦU

#include <stdio.h>

#include <algorithm>

#include <iostream>

#include <vector>

using namespace std;

const int N = 100005;

int n, m;

vector<int> a[N];

int CriticalEdge = 0;

bool CriticalNode[N];

int Num[N], Low[N], Time = 0;

void visit(int u, int p) {

int NumChild = 0;

Low[u] = Num[u] = ++Time;

for (int v : a[u])

if (v != p) {

if (Num[v] != 0)

Low[u] = min(Low[u], Num[v]);

else {

visit(v, u);

NumChild++;

Low[u] = min(Low[u], Low[v]);

if (Low[v] >= Num[v])

CriticalEdge++;

if (u == p) {

if (NumChild >= 2)

CriticalNode[u] = true;

} else {

if (Low[v] >= Num[u])

CriticalNode[u] = true;

}

}

}

}

int main() {

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

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

int x, y;

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

a[x].push\_back(y);

a[y].push\_back(x);

}

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

if (!Num[i]) visit(i, i);

int Count = 0;

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

if (CriticalNode[i]) Count++;

printf("%d %d\n", Count, CriticalEdge);

}

/\*

Input:

5 3

1 2

2 3

4 5

Output:

1 3

\*/

1. THUẬT TOÁN Z

// Cho một xâu S. Với mỗi vị trí I của S, goi z[i] là giá trị lớn nhất sao //cho xâu con độ dài z[i] bắt đầu tại I bằng với tiền tố đồ dài z[i] của S. //Nói cách khác, với mỗi I, tìm x=z[i] lớn nhất sao cho S[0..x-//1]==S[i..i+x-1] (phần tử đánh số từ 0)

#include <stdio.h>

#include <string.h>

#include <algorithm>

#include <iostream>

using namespace std;

#define N 500005

int n, z[N];

char a[N];

void make\_z(char a[], int n, int F[]) {

int L = -1, R = -1;

F[0] = n;

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

if (i > R) {

L = i;

R = i - 1;

while (R < n - 1 && a[R + 1] == a[R - L + 1]) R++;

F[i] = R - L + 1;

} else {

if (F[i - L] < R - i + 1)

F[i] = F[i - L];

else {

L = i;

while (R < n - 1 && a[R + 1] == a[R - L + 1]) R++;

F[i] = R - L + 1;

}

}

}

}

int main() {

gets(a);

n = strlen(a);

make\_z(a, n, z);

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

printf("%d ", z[i]);

printf("\n");

}

/\*

Input:

ab

Output:

2 0

\*/

1. BIGNUM TEMPLATE

#include <assert.h>

#include <stdio.h>

#include <algorithm>

#include <iostream>

#include <vector>

using namespace std;

#define long long long

typedef vector<int> vi;

const int BASE = 10000;

void fix(vi &a) {

a.push\_back(0);

for (int i = 0; i < a.size() - 1; ++i) {

a[i + 1] += a[i] / BASE;

a[i] %= BASE;

if (a[i] < 0) {

a[i] += BASE;

a[i + 1]--;

}

}

while (a.size() >= 2 && a.back() == 0) a.pop\_back();

}

vi operator\*(const vi &a, const vi &b) {

vi c(a.size() + b.size() + 1);

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

for (int j = 0; j < b.size(); ++j) {

c[i + j] += a[i] \* b[j];

c[i + j + 1] += c[i + j] / BASE;

c[i + j] %= BASE;

}

return fix(c), c;

}

vi to\_vi(int x) { // x < Base

assert(x < BASE);

return vi(1, x);

}

vi operator+(vi a, const vi &b) {

a.resize(max(a.size(), b.size()));

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

a[i] += b[i];

return fix(a), a;

}

vi operator-(vi a, const vi &b) {

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

a[i] -= b[i];

return fix(a), a;

}

vi operator\*(vi a, int x) { // x < BASE

assert(x < BASE);

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

a[i] \*= x;

return fix(a), a;

}

bool operator<(const vi &a, const vi &b) {

if (a.size() != b.size()) return a.size() < b.size();

for (int i = a.size() - 1; i >= 0; i--)

if (a[i] != b[i])

return a[i] < b[i];

return false;

}

vi operator/(vi a, int x) { // x < BASE

assert(x < BASE);

for (int i = (int)a.size() - 1, r = 0; i >= 0; --i, r %= x) {

r = r \* BASE + a[i];

a[i] = r / x;

}

return fix(a), a;

}

int operator%(const vi &a, int x) { //x < BASE

int r = 0;

for (int i = (int)a.size() - 1; i >= 0; --i)

r = (r \* BASE + a[i]) % x;

return r;

}

istream &operator>>(istream &cin, vi &a) {

string s;

cin >> s;

a.clear();

a.resize(s.size() / 4 + 1);

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

int x = (s.size() - 1 - i) / 4; // <- log10(BASE)=4

a[x] = a[x] \* 10 + (s[i] - '0');

}

return fix(a), cin;

}

ostream &operator<<(ostream &cout, const vi &a) {

printf("%d", a.back());

for (int i = (int)a.size() - 2; i >= 0; i--)

printf("%04d", a[i]);

return cout;

}

void test\_fib(int n) {

vi a = to\_vi(1), b = to\_vi(1);

for (int i = 1; i <= n / 2; ++i) {

a = a + b;

b = b + a;

cout << "F[" << i \* 2 + 1 << "]=" << a << endl;

cout << "F[" << i \* 2 + 2 << "]=" << b << endl;

}

}

void test\_fact(int n) {

vi P = to\_vi(1);

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

P = P \* i;

cout << i << "!= " << P << endl;

}

}

void test\_divide() {

vi a;

int x;

for (;;) {

cout << "Input a big integer and a small integer (<10000)" << endl;

if (cin >> a >> x)

;

else

break;

cout << "a=" << a << " x=" << x << endl;

vi q = a / x;

int r = a % x;

cout << "a/x=" << q << "; a%x=" << r << endl;

vi a0 = q \* to\_vi(x) + to\_vi(r);

assert(a0 == a && !(a0 < a) && !(a0 > a));

}

}

int main() {

// cout << "Press Enter to run test\_fib()" << endl;

// cin.ignore(1);

// test\_fib(100);

// cout << "Press Enter to run test\_fact()" << endl;

// cin.ignore(1);

// test\_fact(100);

// cout << "Press Enter to run test\_divide()" << endl;

// cin.ignore(1);

// test\_divide();

vi a, b;

cin >> a >> b;

if (a < b)

cout << "a<b\n";

else if (b < a)

cout << "a>b\n";

else

cout << "a=b\n";

cout << a + b << '\n';

if (a < b) swap(a, b); //a must be >= b

cout << a - b << '\n';

cout << a \* b << '\n';

cin >> a;

int x;

cin >> x;

cout << a / x << '\n';

cout << a % x << '\n';

}

/\*

Input:

4564545645645645645456465465

564564564

Output:

2 0

\*/

1. AHO – CORASICK

// Dùng thuật toán Aho Corasick để giải bài toán sau.

//Cho một số pattern và một xâu S. Một phép biến đổi là thay thế một kí tự //trong xâu S bằng một kí tự khác. Tìm cách biến đổi ít nhất để không có //pattern nào xuất hiện trong S như một xâu con liên tiếp. Biết rằng các //pattern không chứa dấu cách còn xâu S có thể có dấu cách.

#include <stdio.h>

#include <string.h>

#include <algorithm>

#include <iostream>

#include <queue>

using namespace std;

int m, n;

namespace trie {

const int N = 2003;

int a[N][128], Peak = 0; //

int Prev[N];

bool Leaf[N]; //

void clear() {

Peak = 0;

memset(a, 0, sizeof a);

memset(Prev, 0, sizeof Prev);

memset(Leaf, 0, sizeof Leaf);

}

void insert(char s[]) {

int u = 0;

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

if (!a[u][k]) a[u][k] = ++Peak;

u = a[u][k];

}

Leaf[u] = true;

}

int next(int u, char k) {

for (int i = u; i != -1; i = Prev[i])

if (a[i][k]) return a[i][k];

return 0;

}

void bfs() {

queue<int> qu;

qu.push(0);

Prev[0] = -1;

while (qu.size()) {

int u = qu.front();

qu.pop();

for (int k = 0; k < 128; k++)

if (int v = a[u][k]) {

Prev[v] = next(Prev[u], k);

Leaf[v] |= Leaf[Prev[v]];

qu.push(v);

}

}

}

}; // namespace trie

int solve(char s[]) {

using namespace trie;

int u = 0, Count = 0;

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

u = next(u, s[i]);

if (Leaf[u]) {

Count++;

u = 0;

}

}

return Count;

}

void solve() {

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

char s[123];

gets(s);

trie ::insert(s);

}

trie ::bfs();

int Sum = 0;

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

char s[123];

gets(s);

Sum += solve(s);

}

cout << Sum << endl;

}

void readln() {

do {

} while (getchar() != '\n');

}

int main() {

for (;;) {

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

readln();

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

solve();

trie ::clear();

}

}

/\*

Input:

4 6

:-)

:-(

(-:

)-:

Hello uncle John! :-) :-D

I am sad or happy? (-:-(?

I feel so happy, my head spins

(-:-)(-:-)(-:-)(-:-) :-) (-: :-)

but then sadness comes :-(

Loves you, Joanna :-)))))

3 1

:)

):

))

:):)):)):)):(:((:(((:):)

0 0

\*/

/\*

11

8

\*/