

*Heaven's Light is Our Guide"*



**Department of Computer Science & Engineering**

**RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOG**

**Software Project-01**

**Submitted By:**

**Name: Khandoker Sefayet Alam**

**Roll:2003121**

**Department: Computer Science & Engineering  
Section-C**

**Course code: CSE 2100**

**Course name: Software development project I**

**Submitted To:**

**SUHRID SHAKHAR GHOSH**

**Assistant Professor**

**Dept of CSE,RUET**

**Project Title:**

**Learn 10 different Algorithms useful for competitive programming and implement them in solving problems.**

## Algorithm -1: Depth First Search

### Problem-1:

Link: <https://codeforces.com/contest/727/problem/A>

### Solution:

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

using namespace std;

#define ll          long long
#define scl(n)      scanf("%lld", &n)
#define fr(i,n)     for (ll i=0;i<n;i++)
#define fr1(i,n)    for(ll i=1;i<=n;i++)
#define pfl(x)      printf("%lld\n",x)
#define endl        "\n"
#define pb          push_back
#define asort(a)     sort(a,a+n)
#define dsort(a)     sort(a,a+n,greater<int>())
#define vasort(v)    sort(v.begin(), v.end());
#define vdsort(v)    sort(v.begin(), v.end(),greater<ll>());
#define pn          printf("\n")
#define md          10000007
#define debug       printf("I am here\n")

#define l(s)         s.size()
#define tcas(i,t)    for(ll i=1;i<=t;i++)
#define pcas(i)      printf("Case %lld: ",i)
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
const ll maxN=1e5+10;//for graph
#define M 10000

vector<ll>g[maxN];
vector<bool>vis;
ll n,m;
vector<ll> vec;

bool dfs(ll n){

    if(n>m) return false;
    if(n==m){
```

```

        // cout<<"YES"<<endl;

        vec.push_back(n);
        return true;
    }

    bool f=0;
    f|=dfs(2*n);

    f|=dfs(10*n+1);

    if(f) vec.push_back(n);
    return f;
}

int main()
{
    fast;
    ll t;
    //ll tno=1;;
    t=1;
    //cin>>t;
    while(t--){

        cin>>n>>m;
        if(dfs(n)){
            cout<<"YES"<<endl;
            sort(vec.begin(),vec.end());
            cout<<vec.size()<<endl;
            for(auto it:vec){cout<<it<<" ";}
            cout<<endl;
        }
        else{
            cout<<"NO"<<endl;
        }
    }

    return 0;
}

```

## Problem-2:

Link: <https://codeforces.com/contest/580/problem/C>

## Solution:

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

using namespace std;

#define ll          long long
#define scl(n)      scanf("%lld", &n)
#define fr(i,n)     for (ll i=0;i<n;i++)
#define fr1(i,n)    for(ll i=1;i<=n;i++)
#define pfl(x)      printf("%lld\n",x)
#define endl        "\n"
#define pb          push_back
#define asort(a)     sort(a,a+n)
#define dsort(a)     sort(a,a+n,greater<int>())
#define vasort(v)    sort(v.begin(), v.end());
#define vdsort(v)    sort(v.begin(), v.end(),greater<ll>());
#define pn          printf("\n")
#define md          10000007
#define debug       printf("I am here\n")

#define l(s)         s.size()
#define tcas(i,t)    for(ll i=1;i<=t;i++)
#define pcas(i)      printf("Case %lld: ",i)
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
const ll maxN=1e5+10;//for graph
#define M 10000

vector<ll>g[maxN];
bool vis[maxN];
vector<ll> cats(maxN);
ll cnt=0;
ll n,m;
void dfs(ll vertex,ll cat){
    //take action on vertex after entering the vertex

    cat+=cats[vertex];
    if(cats[vertex]==0) cat=0;
    if(cat>m) return;
```

```

    ll flag=1;

    vis[vertex]=true;
    // cout<<vertex<<endl;
    for(ll child:g[vertex]){
        //cout<<"par:"<<vertex<<" "<<"child:"<<child<<endl;
        if(vis[child]) continue;
        //take action on the node before entering the child
        flag=0;
        dfs(child,cat);
        //take action on the node after exiting the child
    }
    //take action on the vertex after exiting the node
    if(flag==1 && cat<=m) {
        //cout<<endl<<vertex<<" "<<cnt<<endl;
        cnt++;}
}
void reset(){

    for(ll i=0;i<n;i++){
        g[i].clear();
        vis[i]=0;
    }
    cats.clear();
}
int main()
{
    fast;
    ll t;
    //ll tno=1;;
    t=1;
    //cin>>t;
    while(t--){
        cnt=0;
        cin>>n>>m;
        for(ll i=1;i<=n;i++){
            cin>>cats[i];
        }
        ll v1,v2;
        for(ll i=0;i<n-1;i++){
            cin>>v1>>v2;
            g[v1].push_back(v2);
            g[v2].push_back(v1);
        }
        dfs(1,0);
    }
}

```

```
    cout<<cnt<<endl;  
}  
  
    return 0;  
}
```

### Problem-3:

Link: <https://codeforces.com/contest/24/problem/A>

### Solution:

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

using namespace std;

#define ll          long long
#define scl(n)      scanf("%lld", &n)
#define fr(i,n)     for (ll i=0;i<n;i++)
#define fr1(i,n)    for(ll i=1;i<=n;i++)
#define pfl(x)      printf("%lld\n",x)
#define endl        "\n"
#define pb          push_back
#define asort(a)     sort(a,a+n)
#define dsort(a)     sort(a,a+n,greater<int>())
#define vasort(v)    sort(v.begin(), v.end());
#define vdsort(v)    sort(v.begin(), v.end(),greater<ll>());
#define pn          printf("\n")
#define md          10000007
#define debug       printf("I am here\n")

#define l(s)         s.size()
#define tcas(i,t)    for(ll i=1;i<=t;i++)
#define pcas(i)      printf("Case %lld: ",i)
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
const ll maxN=1e5+10;//for graph
#define M 10000

vector<pair<ll,ll> >g[maxN];
bool vis[maxN];

ll cost1=0,cost2=0;
void dfs(ll vertex){
    //take action on vertex after entering the vertex
    vis[vertex]=true;
    // cout<<vertex<<endl;
    for(auto child:g[vertex]){
        if(vis[child.first]) continue;
        cost1+=child.second;
    }
}
```

```

        dfs(child.first);

    }
    //take action on the vertex after exiting the node
}
void dfs2(ll vertex){
    //take action on vertex after entering the vertex
    vis[vertex]=true;
    // cout<<vertex<<endl;
    for(auto child:g[vertex]){
        if(vis[child.first]) continue;
        cost2+=child.second;

        dfs2(child.first);

    }
    //take action on the vertex after exiting the node
}
int main()
{
    //fast;
    ll t;
    //ll tno=1;;
    t=1;
    //cin>>t;
    while(t--){
        ll n,m;
        cin>>n;
        ll v1,v2,k;
        for(ll i=0;i<n;i++){
            cin>>v1>>v2>>k;
            g[v1].push_back({v2,0});
            g[v2].push_back({v1,k});
        }
        pair<ll,ll> point=g[1].back();
        g[1].pop_back();
        memset(vis,0,sizeof(vis));
        dfs(1);
        pair<ll,ll> q=g[1].back();
        g[1].pop_back();
        g[1].push_back(point);
        memset(vis,0,sizeof(vis));
        dfs2(1);
        for(auto &it:g[point.first]){
            if(it.first==1) cost1+=it.second;

```



```
}  
for(auto &it:g[q.first]){  
    if(it.first==1) cost2+=it.second;  
}  
cout<<min(cost1,cost2)<<endl;  
}  
  
return 0;  
}
```

## Algorithm -2: Breadth First Search(BFS)

### Problem-1:

Link: <https://www.spoj.com/problems/NAKANJ/>

### Solution:

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

using namespace std;

#define ll long long
#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define pfl(x) printf("%lld\n",x)
#define endl "\n"
#define pb push_back
#define asort(a) sort(a,a+n)
#define dsort(a) sort(a,a+n,greater<int>())
#define vasort(v) sort(v.begin(), v.end());
#define vdsort(v) sort(v.begin(), v.end(),greater<ll>());
#define pn printf("\n")
#define md 10000007
#define debug printf("I am here\n")

#define l(s) s.size()
#define tcas(i,t) for(ll i=1;i<=t;i++)
#define pcas(i) printf("Case %lld: ",i)
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
const ll maxN=1e5+10;//for graph
const ll INF=1e9+10;//for graph
#define M 10000

vector<ll>g[maxN];
bool vis[8][8];
ll level[8][8];

//bfs=breath first search
//according to level traversal

ll getX(string s){
    return s[0]-'a';
```

```

}
ll getY(string s){
    return s[1]-'1';
}

bool isValid(ll x,ll y){
    return x>=0 && y>=0 && x<8 && y<8;
}

vector<pair<ll,ll> >movements={
{-1,2},{1,2},
{-1,-2},{1,-2},
{2,-1},{2,1},
{-2,-1},{-2,1}
};

ll bfs(string source,string dest){
    ll sourceX=getX(source);
    ll sourceY=getY(source);
    ll destX=getX(dest);
    ll destY=getY(dest);

    queue<pair<ll,ll> >q;
    q.push({sourceX,sourceY});
    vis[sourceX][sourceY]=1;
    level[sourceX][sourceY]=0;
    while(!q.empty()){
        pair<ll,ll> v =q.front();
        ll x=v.first;
        ll y=v.second;
        q.pop();
        for(auto movement:movements){
            ll childX=movement.first+x;
            ll childY=movement.second+y;
            if(!isValid(childX,childY)) continue;
            if(!vis[childX][childY]){
                q.push({childX,childY});
                level[childX][childY]=level[x][y]+1;
                vis[childX][childY]=1;
            }
        }
        if(level[destX][destY]!=INF) break;
    }
    return level[destX][destY];
}

```

```

void reset(){
    for(ll i=0;i<8;i++){
        for(ll j=0;j<8;j++){
            vis[i][j]=0;
            level[i][j]=INF;
        }
    }
}

int main()
{
    fast;
    ll t;
    //ll tno=1;;
    //t=1;
    cin>>t;
    while(t--){
        reset();
        //ll n;
        //cin>>n;
        // ll v1,v2;
        // while(n--)
        string s1,s2;
        cin>>s1>>s2;
        cout<< bfs(s1,s2) <<endl;
    }

    return 0;
}

```

## Problem-2:

Link: <https://www.spoj.com/problems/MICEMAZE/>

## Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace __gnu_pbds;
using namespace std;
#define nn '\n'
#define ff first
#define ss second
#define fo(i,n) for(i=0;i<n;i++)
#define Fo(i,k,n) for(i=k;k<n?i<n:i>n;k<n?i+=1:i-=1)
#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << "," << #y << "=" << y << endl
#define pb push_back
#define eb emplace_back
#define mp make_pair
#define Setpre(n) cout<<fixed<<setprecision(n)
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define UNIQUE(a) (a).erase(unique(all(a)),(a).end())
#define SZ(a) (int)a.size()
#define mem(a,b) memset(a,b,sizeof(a))
#define itr(it, a) for(auto it = a.begin(); it != a.end(); it++)
#define PI 3.1415926535897932384626
#define fast_IO ios_base::sync_with_stdio(0), cin.tie(0), cout.tie(0)
#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))
#define asort(a) sort(a,a+n)
#define dsort(a) sort(a,a+n,greater<int>())
#define vasort(v) sort(v.begin(), v.end());
#define vdsort(v) sort(v.begin(), v.end(),greater<ll>());
#define pn printf("\n")
#define md 10000007
#define debug printf("I am here\n")
#define l(s) s.size()
#define tcas(i,t) for(ll i=1;i<=t;i++)
#define debug printf("I am here\n")
#define fr(i,n) for (ll i=0;i<n;i++)
```

```

#define fr1(i,n)          for(ll i=1;i<=n;i++)
typedef long long ll;
typedef unsigned long long ull;
typedef pair<int, int> pii;
typedef pair<ll, ll> pll;
typedef vector<int> vi;
typedef vector<ll> vl;
typedef vector<pii> vpii;
typedef vector<pll> vpll;
typedef vector<vi> vvi;
typedef vector<vl> vvl;
template <typename T> void display (T const& coll){typename T::const_iterator
pos;typename T::const_iterator end(coll.end());for(pos=coll.begin(); pos!=end;
++pos) cout << *pos << ' ';cout << endl;}
template <typename T> using PQ = priority_queue<T>;
template <typename T> using QP = priority_queue<T,vector<T>,greater<T>>;
template <typename T> using ordered_set = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
template <typename T> using ordered_multiset = tree<T, null_type,less_equal<T>,
rb_tree_tag,tree_order_statistics_node_update>;
template <typename T,typename R> using ordered_map = tree<T, R , less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
template <typename T,typename R> using ordered_multimap = tree<T, R ,
less_equal<T>, rb_tree_tag, tree_order_statistics_node_update>;
template <typename T> T BigMod (T b, T p, T m) {if (p == 0) return 1; if (p % 2
== 0) {T s = BigMod(b, p / 2, m); return ((s % m) * (s % m)) % m;} return ((b %
m) * (BigMod(b, p - 1, m) % m)) % m;}
template <typename T> T ModInv (T b, T m) {return BigMod(b, m - 2, m);}
inline void swap(ll &x, ll &y) {ll temp = x; x = y; y = temp;}
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/((double)log(base);}
inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x)};
int poplow(ll x){return __builtin_ctzll(x)};
int pophigh(ll x){return 63 - __builtin_clzll(x)};
mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
int my_rand(int l, int r) {
    return uniform_int_distribution<int>(l, r) (rng);
}

```

```

struct custom_hash {
    static uint64_t splitmix64(uint64_t x) {
        x += 0x9e3779b97f4a7c15;
        x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};

const double EPS = 1e-9;
const int N = 2e5+10;
const int M = 1e9+7;

// int a[N];
// vl g[N];

vpll g[N];
vl dist(N,LLONG_MAX);
vl par(N,-1);

void dijkstra(int source){
    QP<p11> pq;
    pq.push({0,source});
    dist[source]=0;
    while(pq.size()){
        ll v=pq.top().second;
        ll v_dist=pq.top().first;
        pq.pop();
        if(dist[v]<v_dist) continue;
        for(auto &child:g[v]){
            ll child_v=child.first;
            ll wt=child.second;
            if(dist[v]+wt<dist[child_v]){
                dist[child_v]=dist[v]+wt;
                //par[child_v]=v;
                pq.push({dist[child_v],child_v});
            }
        }
    }
}

void reset(ll n){

```

```

        for(ll i=0;i<N;i++){
            g[i].clear();
            dist[i]=LLONG_MAX;
        }
    }

int main() {
    fast_IO;

    // #ifndef ONLINE_JUDGE
    //     freopen("input.txt","r",stdin);
    //     freopen("output.txt","w",stdout);
    // #endif

    int t = 1;
    //cin >> t;
    while(t--){
        ll n,m,ex,tim;
        cin>>n>>ex>>tim>>m;
        reset(n);
        ll v1,v2,w;
        for(ll i=0;i<m;i++){
            cin>>v1>>v2>>w;
            g[v2].push_back({v1,w});
        }
        dijkstra(ex);
        ll ans=0;
        for(ll i=1;i<=n;i++){
            if(dist[i]<=tim && tim>=0){
                ans++;
            }
        }
        cout<<ans<<endl;

    }

    return 0;
}

```



### Problem-3:

Link: <https://cses.fi/problemset/task/1193>

### Solution:

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

using namespace std;

#define ll long long
#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define pfl(x) printf("%lld\n",x)
#define endl "\n"
#define pb push_back
#define asort(a) sort(a,a+n)
#define dsort(a) sort(a,a+n,greater<int>())
#define vasort(v) sort(v.begin(), v.end());
#define vdsort(v) sort(v.begin(), v.end(),greater<ll>());
#define pn printf("\n")
#define md 10000007
#define debug printf("I am here\n")

#define l(s) s.size()
#define tcas(i,t) for(ll i=1;i<=t;i++)
#define pcas(i) printf("Case %lld: ",i)
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);

#define M 10000
const ll INF=1e9+10;//for graph
const ll maxN=1e3+10;//for graph

bool vis[maxN][maxN];

ll n,m;
ll stx,sty,exx,eny;
ll arr[maxN][maxN];
char ch[maxN][maxN];

bool isValid(ll i,ll j){
    if(i<0 || j<0 || i>=n || j>=m || vis[i][j] || arr[i][j]=='#'){
```

```

        return false;
    }
    return true;
}

ll level[maxN][maxN];
ll prev[maxN][maxN];
//bfs=breath first search
//according to level traversal
vector<pair<ll,ll> >movements={{-1,0},{1,0},{0,-1},{0,1}};

void bfs(ll srcx,ll srcy){
    queue<pair<ll,ll> > q;
    q.push({srcx,srcy});
    vis[srcx][srcy]=1;
    level[srcx][srcy]=0;
    while(!q.empty()){
        pair<ll,ll> curr_v=q.front();
        q.pop();
        //cout<<curr_v.first<<" "<<curr_v.second<<endl;
        for(auto it:movements){
            if(isvalid(curr_v.first+it.first,curr_v.second+it.second)){
                q.push({curr_v.first+it.first,curr_v.second+it.second});
                if(it.first==-1 && it.second==0)
ch[curr_v.first+it.first][curr_v.second+it.second]='U';
                if(it.first==1 && it.second==0)
ch[curr_v.first+it.first][curr_v.second+it.second]='D';
                if(it.first==0 && it.second==-1)
ch[curr_v.first+it.first][curr_v.second+it.second]='L';
                if(it.first==0 && it.second==1)
ch[curr_v.first+it.first][curr_v.second+it.second]='R';
                vis[curr_v.first+it.first][curr_v.second+it.second]=1;
                level[curr_v.first+it.first][curr_v.second+it.second]=level[curr_v.first][curr_v.second]+1;

            }
        }
    }
}

void reset(){
    for(ll i=0;i<maxN;i++){
        for(ll j=0;j<maxN;j++){

```

```

        vis[i][j]=0;
        level[i][j]=INF;
    }
}

int main()
{
    fast;
    ll t;
    //setIO();
    //ll tno=1;;
    t=1;
    //cin>>t;

    while(t--){
        cin>>n>>m;
        string s;

        for(ll i=0;i<n;i++){
            cin>>s;
            for(ll j=0;j<m;j++){
                arr[i][j]=(ll)s[j];

                if(arr[i][j]=='A'){
                    stx=i;
                    sty=j;
                }
                if(arr[i][j]=='B'){
                    enx=i;
                    eny=j;
                }
            }
        }
        reset();
        bfs(stx,sty);
        if(level[enx][eny]==INF){
            cout<<"NO"<<endl;
        }
        else{
            cout<<"YES"<<endl;
            cout<<level[enx][eny]<<endl;
        }
    }
}

```

```

        string ans = "";

        // Starting from B to track back
        int curri = enx, currj = eny;

        // loop until we reach A(starti, startj)
        while(!(curri==stx && currj==sty)){

            // storing direction in our answer where we have come
            // it is opposite as while doing bfs
            // because we are going from B to A now
            ans += ch[curri][currj];
            switch(ch[curri][currj]){
                case 'D':
                    curri--;
                    break;
                case 'U':
                    curri++;
                    break;
                case 'R':
                    currj--;
                    break;
                case 'L':
                    currj++;
                    break;
            }
        }
        reverse(ans.begin(),ans.end());
        cout<<ans<<endl;
    }

    return 0;
}

```

## Algorithm -3: Dijkstra:

### Problem-1:

Link: <https://codeforces.com/contest/1822/problem/F>

### Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

//VVI
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define pb push_back
#define ll long long
#define ff first
#define ss second
#define SZ(a) (int)a.size()
#define UNIQUE(a) (a).erase(unique(all(a)),(a).end())
#define eb emplace_back
#define mp make_pair

///BIT MANIPULATION

#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))

//LOOPS

#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define Fo(i,k,n) for(i=k;k<n?i<n:i>n;k<n?i+=1:i-=1)

///PRINTING

#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << ", " << #y << "=" << y << endl
```

```

#define nn '\n'
#define pfl(x)          printf("%lld\n",x)
#define pcas(i)          printf("Case %lld: ",i)
#define Setpre(n) cout<<fixed<<setprecision(n)
#define itr(it, a) for(auto it = a.begin(); it != a.end(); it++)
#define debug            printf("I am here\n")

///SORTING AND FILLING

#define asort(a)          sort(a,a+n)
#define dsort(a)          sort(a,a+n,greater<int>())
#define vasort(v)         sort(v.begin(), v.end());
#define vdsort(v)         sort(v.begin(), v.end(),greater<ll>());
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define mem(a,b) memset(a,b,sizeof(a))
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))

///CONSTANTS
#define md                10000007
#define PI 3.1415926535897932384626
const double EPS = 1e-9;
const ll N = 2e5+10;
const ll M = 1e9+7;

///INLINE FUNCTIONS
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/((double)log(base));}
inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x)};
int poplow(ll x){return __builtin_ctzll(x)};
int pophigh(ll x){return 63 - __builtin_clzll(x)};

/// Data structures
typedef unsigned long long ull;
typedef pair<ll, ll>    pll;

```

```

typedef vector<ll>      vl;
typedef vector<pll>     vpll;
typedef vector<vl>      vvl;
template <typename T> using PQ = priority_queue<T>;
template <typename T> using QP = priority_queue<T,vector<T>,greater<T>>;

template <typename T> using ordered_set = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
template <typename T,typename R> using ordered_map = tree<T, R , less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
;

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ){ os << it << endl; }
return os; }
    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }
    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }

```

```

    string to_string ( const char & c ) { string s; s += c; return "\"" + s +
    "\""; }

    template<typename Type> string to_string( vector<Type> );
    template<typename First, typename Second> string to_string( pair<First,
    Second> );
    template<typename Collection> string to_string( Collection );

    template<typename First, typename Second> string to_string( pair<First,
    Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
    "}"; }

    template<typename Type> string to_string( vector<Type> v ) { bool sep =
    false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
    to_string( x ); } s += "]"; return s; }

    template<typename Collection> string to_string( Collection collection ) {
    bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
    sep = true; s += to_string( x ); } s += "}"; return s; }

    void print() { cerr << endl; sep = false; }
    template <typename First, typename... Other> void print( First first,
    Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
    ); print( other... ); }

} using namespace io;

/*=====//

SEFARDET

//=====*/

void setIO(){
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);

    freopen("output.txt", "w", stdout);
    #endif // ONLINE_JUDGE
}

struct custom_hash {
    static uint64_t splitmix64(uint64_t x) {

```



```

        x += 0x9e3779b97f4a7c15;
        x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};
vp11 g[N];
v1 dist(N,LLONG_MAX);
// v1 par(N,-1);

void dijkstra(ll source){
    QP<p11> pq;
    pq.push(mp(0,source));
    dist[source]=0;
    while(pq.size()){
        ll v=pq.top().second;
        ll v_dist=pq.top().first;
        pq.pop();
        if(dist[v]<v_dist) continue;
        for(auto &child:g[v]){
            ll child_v=child.first;
            ll wt=child.second;
            if(dist[v]+wt<dist[child_v]){
                dist[child_v]=dist[v]+wt;
                // par[child_v]=v;
                pq.push(mp(dist[child_v],child_v));
            }
        }
    }
}

vector<ll>g2[N];
bool vis[N];
ll level[N];

void bfs(ll source){
    queue<ll> q;
    q.push(source);
    vis[source]=1;
    level[source]=0;

```

```

while(!q.empty()){
    ll cur_v=q.front();
    q.pop();
    for(ll child:g2[cur_v]){
        if(!vis[child]){
            q.push(child);
            vis[child]=1;
            level[child]=1+level[cur_v];
        }
    }
}
}

```

```

void reset(ll n){
    mem(vis,0);
    for(ll i=0;i<=n;i++){
        g[i].clear();
        g2[i].clear();
        dist[i]=LLONG_MAX;
    }
}

```

```

int main()
{
    fast;
    ll t;
    //setIO();
    //ll tno=1;;
    t=1;
    cin>>t;

    while(t--){
        ll n,w,c;
        cin>>n>>w>>c;
        reset(n);
        ll u,v;
        for(ll i=0;i<n-1;i++){
            cin>>u>>v;
            g[u].push_back({v,w});
            g[v].push_back({u,w});
            g2[u].push_back(v);
            g2[v].push_back(u);
        }
        bfs(1);
        ll ans=0;
    }
}

```

```

dijkstra(1);
ll mx_node=-1;
for(ll i=1;i<=n;i++){
    if(dist[i]>ans){
        ans=dist[i];
        mx_node=i;
    }
}
for(ll i=1;i<=n;i++){
    dist[i]=LLONG_MAX;
}
dijkstra(mx_node);
// deb(mx_node);
// deb(ans);
for(ll i=1;i<=n;i++){
    // cout<<i<<" "<<dist[i]<<nn;
    ans=max(ans,dist[i]-level[i]*c);
}
cout<<ans<<nn;
}
return 0;
}

```

## Problem-2:

Link: <https://codeforces.com/problemset/problem/20/C>

## Solution:

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

using namespace std;

#define ll long long
#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define pfl(x) printf("%lld\n",x)
#define endl "\n"
#define pb push_back
#define asort(a) sort(a,a+n)
#define dsort(a) sort(a,a+n,greater<int>())
#define vasort(v) sort(v.begin(), v.end());
#define vdsort(v) sort(v.begin(), v.end(),greater<ll>());
#define pn printf("\n")
#define md 10000007
#define debug printf("I am here\n")

#define l(s) s.size()
#define tcas(i,t) for(ll i=1;i<=t;i++)
#define pcas(i) printf("Case %lld: ",i)
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
const ll maxN=1e5+10;//for graph
const ll INF=1e16+10;
#define M 10000

vector<pair<ll,ll> >g[maxN];
vector<ll> lev(maxN,INF);
vector<ll>par(maxN);
bool vis[maxN];
//when edges dont have same weight...0 and 1 weights..use 0-1 bfs
ll n,m,k;
vector<ll> ans;
void bfs(){

    priority_queue<pair<ll,ll>,vector<pair<ll,ll> >,greater<pair<ll,ll> >>st;
    st.push({1,0});
```

```

lev[1]=0;
vis[1]=1;
while(st.size()>0){
    auto node=st.top();
    ll wt=node.second;
    ll v=node.first;
    st.pop();

    vis[v]=1;
    for(auto child :g[v]){
        int child_v=child.first;
        int wt=child.second;
        if(vis[child_v] && lev[v]+wt>lev[child_v]) continue;
        if(lev[v]+wt<lev[child_v]){
            lev[child_v]=lev[v]+wt;
            par[child_v]=v;
            st.push({child_v,lev[child_v]});
        }
    }
}

//return lev[n]==INF? -1:lev[n];

void func(ll vertex){
    ans.push_back(vertex);
    if(vertex==1) return;
    func(par[vertex]);
}

int main()
{
    fast;
    ll t;
    //ll tno=1;;
    t=1;
    //cin>>t;
    while(t--){

        cin>>n>>m;
        ll v1,v2;
        for(ll i=0;i<m;i++){
            cin>>v1>>v2>>k;

```

```

        if(v1==v2) continue; //for loops
        g[v1].push_back({v2,k});
        //normal weight=0
        g[v2].push_back({v1,k});
        //reverse-> weight =1

    }
    bfs();
    if(lev[n]==INF) {cout<<-1<<endl; return 0;}
    func(n);

//    for(ll i=1;i<=n;i++){
//        cout<<lev[i]<<" ";
//    }
//    cout<<endl;
    reverse(ans.begin(),ans.end());
    // for(auto it:ans){
    //     cout<<it<<" ";
    // }
    for(auto it:ans){
        cout<<it<<" ";
    }
    cout<<endl;

}

    return 0;
}

```

### Problem-3:

Link: <https://codeforces.com/problemset/problem/449/B>

### Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace __gnu_pbds;
#define nn '\n'
#define fo(i,n) for(i=0;i<n;i++)
#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << "," << #y << "=" << y << endl
#define Setpre(n) cout<<fixed<<setprecision(n)
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define mem(a,b) memset(a,b,sizeof(a))
#define fast_IO ios_base::sync_with_stdio(0), cin.tie(0), cout.tie(0)
#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))
#define ll long long
#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define pfl(x) printf("%lld\n",x)
#define pb push_back
#define asort(a) sort(a,a+n)
#define dsort(a) sort(a,a+n,greater<int>())
#define vasort(v) sort(v.begin(), v.end());
#define vdsort(v) sort(v.begin(), v.end(),greater<ll>());
#define pn printf("\n")
#define md 10000007
#define debug printf("I am here\n")
#define l(s) s.size()
#define tcas(i,t) for(ll i=1;i<=t;i++)
```

```

#define pcas(i)                printf("Case %lld: ",i)
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << ", " << #y << "=" << y << endl
#define Setpre(n) cout<<fixed<<setprecision(n)
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/((double)log(base));}

inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x);};
int poplow(ll x){return __builtin_ctzll(x);};
int pophigh(ll x){return 63 - __builtin_clzll(x);};

typedef unsigned long long ull;
typedef pair<ll, ll>    pll;
typedef vector<ll>      vl;
typedef vector<pll>     vpll;
typedef vector<vl>      vvl;
template <typename T> using PQ = priority_queue<T>;
template <typename T> using QP = priority_queue<T,vector<T>,greater<T>>;

template <typename T> using ordered_set = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
template <typename T,typename R> using ordered_map = tree<T, R , less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
;
const double EPS = 1e-9;
//const ll N = 2e5+10;
//const ll M = 1e9+7;

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ) { os << it << endl; }
return os; }

```



```

    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }

    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }

    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }

    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }
    string to_string ( const char &c ) { string s; s += c; return "'" + s +
"\""; }

    template<typename Type> string to_string( vector<Type> );
    template<typename First, typename Second> string to_string( pair<First,
Second> );
    template<typename Collection> string to_string( Collection );

    template<typename First, typename Second> string to_string( pair<First,
Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
"}"; }

    template<typename Type> string to_string( vector<Type> v ) { bool sep =
false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
to_string( x ); } s += "]"; return s; }

    template<typename Collection> string to_string( Collection collection ) {
bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
sep = true; s += to_string( x ); } s += "}"; return s; }

    void print() { cerr << endl; sep = false; }

```

```

    template <typename First, typename... Other> void print( First first,
Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
); print( other... ); }

} using namespace io;

/*=====//

SEF04ET

//=====*/

void setIO(){
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);

    freopen("output.txt", "w", stdout);
    #endif // ONLINE_JUDGE
}

struct custom_hash {
    static uint64_t splitmix64(uint64_t x) {
        x += 0x9e3779b97f4a7c15;
        x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};

ll N,M,K;
vector<vector<pair<int,int> > > adj;
ll par[100005];
ll dist[100005];
ll edge[400005];
ll relaxed[400005];

```

```

void dijkstra(){
    priority_queue<pair<long long, int> > q;
    for(int i=1;i<=N;++i)dist[i] = (long long) 1e16;
    for(int i=1;i<=N;++i)par[i] = -1;
    dist[1] = 0;
    q.push(make_pair(dist[1],1));
    while(!q.empty()){
        long long d = -q.top().first;
        int u = q.top().second;
        q.pop();
        if(d > dist[u])continue;
        for(int i=0;i<adj[u].size();++i){
            int v = adj[u][i].first;
            int e = adj[u][i].second;
            if(dist[v] > d + edge[e]){
                dist[v] = d + edge[e];
                par[v] = e;
                q.push(make_pair(-dist[v],v));
            } else if(dist[v] == d + edge[e]){
                if(e < M){
                    par[v] = e;
                }
            }
        }
    }
}

int main(){
    ll u,v;
    long long w;
    cin>>N>>M>>K;
    adj = vector<vector<pair<int,int> > > (N+3);
    for(int i=0;i<M;++i){
        cin>>u>>v>>w;
        adj[u].push_back(make_pair(v,i));
        adj[v].push_back(make_pair(u,i));
        edge[i] = w;
    }
    for(int i=0;i<K;++i){
        cin>>v>>w;
        adj[1].push_back(make_pair(v,i+M));
        adj[v].push_back(make_pair(1,i+M));
        edge[i+M] = w;
    }
    dijkstra();
}

```

```
for(int i=1;i<=N;++i){
    if(par[i]==-1)continue;
    relaxed[par[i]] = 1;
}
int ans = 0;
for(int i=M;i<M+K;++i){
    if(!relaxed[i]) ++ans;
}
printf("%d\n",ans);
return 0;
}
```

## Algorithm -4: Bellman-Ford

### Problem-1:

Link: <https://www.eolymp.com/en/problems/1453>

### Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace __gnu_pbds;
#define nn '\n'
#define fo(i,n) for(i=0;i<n;i++)
#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << ", " << #y << "=" << y << endl
#define Setpre(n) cout<<fixed<<setprecision(n)
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define mem(a,b) memset(a,b,sizeof(a))
#define fast_IO ios_base::sync_with_stdio(0), cin.tie(0), cout.tie(0)
#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))
#define ll long long
#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define pfl(x) printf("%lld\n",x)
#define pb push_back
#define asort(a) sort(a,a+n)
#define dsort(a) sort(a,a+n,greater<int>())
#define vasort(v) sort(v.begin(), v.end());
#define vdsort(v) sort(v.begin(), v.end(),greater<ll>());
#define pn printf("\n")
#define md 10000007
#define debug printf("I am here\n")
#define l(s) s.size()
#define tcas(i,t) for(ll i=1;i<=t;i++)
```

```

#define pcas(i)                printf("Case %lld: ",i)
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << ", " << #y << "=" << y << endl
#define Setpre(n) cout<<fixed<<setprecision(n)
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/((double)log(base));}

inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x);};
int poplow(ll x){return __builtin_ctzll(x);};
int pophigh(ll x){return 63 - __builtin_clzll(x);};

typedef unsigned long long ull;
typedef pair<ll, ll>    pll;
typedef vector<ll>      vl;
typedef vector<pll>     vpll;
typedef vector<vl>      vvl;
template <typename T> using PQ = priority_queue<T>;
template <typename T> using QP = priority_queue<T,vector<T>,greater<T>>;

template <typename T> using ordered_set = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
template <typename T,typename R> using ordered_map = tree<T, R , less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
;
const double EPS = 1e-9;
const ll N = 2e5+10;
const ll INF = 30000;

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ) { os << it << endl; }
return os; }

```

```

    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }

    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }

    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }

    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }
    string to_string ( const char &c ) { string s; s += c; return "'" + s +
"\""; }

    template<typename Type> string to_string( vector<Type> );
    template<typename First, typename Second> string to_string( pair<First,
Second> );
    template<typename Collection> string to_string( Collection );

    template<typename First, typename Second> string to_string( pair<First,
Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
"}"; }

    template<typename Type> string to_string( vector<Type> v ) { bool sep =
false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
to_string( x ); } s += "]"; return s; }

    template<typename Collection> string to_string( Collection collection ) {
bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
sep = true; s += to_string( x ); } s += "}"; return s; }

    void print() { cerr << endl; sep = false; }

```

```

    template <typename First, typename... Other> void print( First first,
Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
); print( other... ); }

} using namespace io;

/*=====//

SEF04ET

//=====*/

void setIO(){
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);

    freopen("output.txt", "w", stdout);
    #endif // ONLINE_JUDGE
}

struct custom_hash {
    static uint64_t splitmix64(uint64_t x) {
        x += 0x9e3779b97f4a7c15;
        x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};

vector<ll>g[N];
vector<ll>vis(N);
vector<ll>dist(N);
ll n,m,w;

void reset(ll n){

```



```

        for(ll i=0;i<=n;i++){
            g[i].clear();
            dist[i]=INF;
        }
    }
void BellmanFord(ll src)
{
    dist[src] = 0;
    for (ll i = 0; i < n - 1; i++) {
        for (ll j = 0; j < m; j++) {
            if ( dist[g[j][0]] != INF && dist[g[j][0]] + g[j][2] < dist[g[j][1]])
                dist[g[j][1]] =dist[g[j][0]] + g[j][2];
        }
    }
}

int main()
{
    fast;
    ll t;
    //setIO();
    //ll tno=1;;
    t=1;
    //cin>>t;

    while(t--){
        cin>>n>>m;
        ll v1,v2,k;
        reset(n);
        for(ll i=0;i<m;i++){
            cin>>v1>>v2>>k;
            g[i].push_back(v1);
            g[i].push_back(v2);
            g[i].push_back(k);
        }
        // for(auto it:g){
        //     cout<<it<<endl;
        // }

        BellmanFord(1);
        for (int i = 1; i <=n; i++){
            cout << dist[i] << " ";
        }
        cout<<endl;
    }
}

```

```
}  
  
    return 0;  
}
```

## Problem-2:

Link: <https://cses.fi/problemset/task/1197>

## Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

//VVI
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define pb push_back
#define ll long long
#define ff first
#define ss second
#define SZ(a) (int)a.size()
#define UNIQUE(a) (a).erase(unique(all(a)),(a).end())
#define eb emplace_back
#define mp make_pair

///BIT MANIPULATION

#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))

//LOOPS

#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define Fo(i,k,n) for(i=k;k<n?i<n:i>n;k<n?i+=1:i-=1)

///PRINTING

#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << "," << #y << "=" << y << endl
#define nn '\n'
#define pfl(x) printf("%lld\n",x)
#define pcas(i) printf("Case %lld: ",i)
#define Setpre(n) cout<<fixed<<setprecision(n)
```

```

#define itr(it, a) for(auto it = a.begin(); it != a.end(); it++)
#define debug          printf("I am here\n")

///SORTING AND FILLING

#define asort(a)          sort(a,a+n)
#define dsort(a)          sort(a,a+n,greater<int>())
#define vasort(v)         sort(v.begin(), v.end());
#define vdsort(v)         sort(v.begin(), v.end(),greater<ll>());
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define mem(a,b) memset(a,b,sizeof(a))
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))

///CONSTANTS
#define md                10000007
#define PI 3.1415926535897932384626

///INLINE FUNCTIONS
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/(double)log(base);}
inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x)};
int poplow(ll x){return __builtin_ctzll(x)};
int pophigh(ll x){return 63 - __builtin_clzll(x)};

typedef unsigned long long ull;
typedef pair<ll, ll>    pll;
typedef vector<ll>      vl;
typedef vector<pll>     vpll;
typedef vector<vl>      vvl;
template <typename T> using PQ = priority_queue<T>;
template <typename T> using QP = priority_queue<T,vector<T>,greater<T>>;

template <typename T> using ordered_set = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;

```

```

template <typename T,typename R> using ordered_map = tree<T, R , less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
;
const double EPS = 1e-9;
const ll N = 2e5+10;
const ll M = 1e9+7;

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ) { os << it << endl; }
return os; }
    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }
    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }
    string to_string ( const char &c ) { string s; s += c; return "'" + s +
"';' };

    template<typename Type> string to_string( vector<Type> );

```

```

    template<typename First, typename Second> string to_string( pair<First,
Second> );
    template<typename Collection> string to_string( Collection );

    template<typename First, typename Second> string to_string( pair<First,
Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
"}"; }
    template<typename Type> string to_string( vector<Type> v ) { bool sep =
false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
to_string( x ); } s += "]"; return s; }
    template<typename Collection> string to_string( Collection collection ) {
bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
sep = true; s += to_string( x ); } s += "}"; return s; }

    void print() { cerr << endl; sep = false; }
    template <typename First, typename... Other> void print( First first,
Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
); print( other... ); }

} using namespace io;

/*=====//

SEFFO4ET

//=====*/

void setIO(){
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);

    freopen("output.txt", "w", stdout);
    #endif // ONLINE_JUDGE
}

struct custom_hash {
    static uint64_t splitmix64(uint64_t x) {
        x += 0x9e3779b97f4a7c15;
        x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
};

```

```

    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};

#define INF 1000000009
vp11 g[N];
ll n,m;
v1 dist(N,INF);
v1 par(N,-1);

void bellman_ford(){
    ll x=-1;
    for(ll i=1;i<=n;i++){ dist[i]=INF;par[i]=-1;}
    dist[1]=0;

    for(ll i=0; i<n; i++){
        x=-1;
        for(ll node=1; node<=n; node++){
            //if(dist[node]==INF) continue;
            for(pair<ll,ll> a : g[node]){
                if(dist[a.first]>dist[node]+a.second){
                    dist[a.first]=dist[node]+a.second;
                    par[a.first]=node;
                    x=a.first;
                }
            }
        }
        //if(!x) break;
    }
    if(x==-1){
        cout<<"NO"<<endl;
    }
    else{
        //x can be on any cycle or reachable from some cycle
        v1 path;
        for (ll i=0; i<n; i++) x = par[x];

        for(ll cur=x; ; cur=par[cur]) {
            //cout<<cur<<" ";
            path.push_back (cur);
            if (cur == x && path.size() > 1) break;
        }
    }
}

```

```

        }
        //cout<<endl;
        reverse(path.begin(), path.end());
        cout << "YES"<<endl;
        cout<<path<<endl;
    }
}

void reset(){
    for(ll i=1;i<=n+1;i++){
        g[i].clear();
    }
}

int main()
{
    fast;
    ll t;
    //setIO();
    //ll tno=1;;
    t=1;
    //cin>>t;

    while(t--){

        cin>>n>>m;
        ll u,v,w;
        //cout<<n<<m<<endl;
        reset();
        for(ll i=0;i<m;i++){
            cin>>u>>v>>w;
            g[u].push_back({v,w});
        }
        // for(ll i=1;i<=n;i++){
        //     cout<<i<<" "<<g[i]<<endl;
        // }
        bellman_ford();

    }

    return 0;
}

```





### Problem-3:

Link: <https://cses.fi/problemset/task/1673>

### Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

//VVI
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define pb push_back
#define ll long long
#define ff first
#define ss second
#define SZ(a) (int)a.size()
#define UNIQUE(a) (a).erase(unique(all(a)),(a).end())
#define eb emplace_back
#define mp make_pair

///BIT MANIPULATION

#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))

//LOOPS

#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define Fo(i,k,n) for(i=k;k<n?i<n:i>n;k<n?i+=1:i-=1)

///PRINTING

#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << "," << #y << "=" << y << endl
#define nn '\n'
#define pfl(x) printf("%lld\n",x)
#define pcas(i) printf("Case %lld: ",i)
#define Setpre(n) cout<<fixed<<setprecision(n)
```

```

#define itr(it, a) for(auto it = a.begin(); it != a.end(); it++)
#define debug          printf("I am here\n")

///SORTING AND FILLING

#define asort(a)          sort(a,a+n)
#define dsort(a)          sort(a,a+n,greater<int>())
#define vasort(v)         sort(v.begin(), v.end());
#define vdsort(v)         sort(v.begin(), v.end(),greater<ll>());
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define mem(a,b) memset(a,b,sizeof(a))
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))

///CONSTANTS
#define md                10000007
#define PI 3.1415926535897932384626
const double EPS = 1e-9;
const ll N = 2e5+10;
const ll M = 1e9+7;

///INLINE FUNCTIONS
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/((double)log(base));}
inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x)};
int poplow(ll x){return __builtin_ctzll(x)};
int pophigh(ll x){return 63 - __builtin_clzll(x)};

/// Data structures
typedef unsigned long long ull;
typedef pair<ll, ll>      pll;
typedef vector<ll>        vl;
typedef vector<pll>        vpll;
typedef vector<vl>         vvl;
template <typename T> using PQ = priority_queue<T>;

```

```

template <typename T> using QP = priority_queue<T,vector<T>,greater<T>>;

template <typename T> using ordered_set = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
template <typename T,typename R> using ordered_map = tree<T, R , less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
;

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ) { os << it << endl; }
return os; }
    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }
    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }
    string to_string ( const char &c ) { string s; s += c; return "'" + s +
"';' }; }

    template<typename Type> string to_string( vector<Type> );

```

```

    template<typename First, typename Second> string to_string( pair<First,
Second> );
    template<typename Collection> string to_string( Collection );

    template<typename First, typename Second> string to_string( pair<First,
Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
"}"; }
    template<typename Type> string to_string( vector<Type> v ) { bool sep =
false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
to_string( x ); } s += "]"; return s; }
    template<typename Collection> string to_string( Collection collection ) {
bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
sep = true; s += to_string( x ); } s += "}"; return s; }

    void print() { cerr << endl; sep = false; }
    template <typename First, typename... Other> void print( First first,
Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
); print( other... ); }

} using namespace io;

/*=====//

SEFFO4ET

//=====*/

void setIO(){
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);

    freopen("output.txt", "w", stdout);
    #endif // ONLINE_JUDGE
}

struct custom_hash {
    static uint64_t splitmix64(uint64_t x) {
        x += 0x9e3779b97f4a7c15;
        x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
};

```

```

    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};

///single source longest path problem
#define INF 1e18
const ll NINF = INF*(-1);
vpll g[N];
ll n,m;
vl dist(N,INF);
// vl par(N,-1);
vector<pair<pair<ll,ll>,ll>>edges;

void bellman_ford(){
    ll x=-1;

    dist[1]=0;
    ll u,v,w;
    for(ll i=0; i<n-1; i++){
        // x=-1;
        for(auto it:edges){
            u=it.first.first;
            v=it.first.second;
            w=it.second;
            if(dist[u]==INF) continue;
            dist[v] = min(dist[v], w+dist[u]);
            dist[v] = max(dist[v], NINF);
        }
        //if(!x) break;
    }

    for(ll i=0; i<n-1; i++){
        // x=-1;
        for(auto it:edges){
            u=it.first.first;
            v=it.first.second;
            w=it.second;
            if(dist[u] == INF) continue;
            dist[v] = max(dist[v], NINF);
            if(dist[u]+w < dist[v]) dist[v]=NINF;
            //if(!x) break;
        }
    }
}

```

```

    }
}

int main()
{
    fast;
    ll t;
    //setIO();
    //ll tno=1;;
    t=1;
    //cin>>t;

    while(t--){
        cin>>n>>m;
        ll u,v,w;
        for(ll i=0;i<m;i++){
            cin>>u>>v>>w;

            edges.push_back({{u,v},-w});
        }
        bellman_ford();
        if(dist[n]==NINF) cout<<-1<<nn;
        else cout<<dist[n]*(-1)<<nn;
    }

    return 0;
}

```

## Algorithm -5: Floyd-Warshall Algorithm

### Problem-1:

Link: <https://codeforces.com/contest/295/problem/B>

### Solution:

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

using namespace std;

#define ll          long long
#define scl(n)      scanf("%lld", &n)
#define fr(i,n)     for (ll i=0;i<n;i++)
#define fr1(i,n)    for(ll i=1;i<=n;i++)
#define pfl(x)      printf("%lld\n",x)
#define endl        "\n"
#define pb          push_back
#define asort(a)     sort(a,a+n)
#define dsort(a)     sort(a,a+n,greater<int>())
#define vasort(v)    sort(v.begin(), v.end());
#define vdsort(v)    sort(v.begin(), v.end(),greater<ll>());
#define pn          printf("\n")
#define md          10000007
#define debug       printf("I am here\n")

#define l(s)         s.size()
#define tcas(i,t)    for(ll i=1;i<=t;i++)
#define pcas(i)      printf("Case %lld: ",i)
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
const ll N=510;//for graph
const ll INF=1e9+10;
#define M 10000

ll dist[N][N];

int main()
{
    fast;
    ll t;
```



```

        //ll tno=1;;
        t=1;
        //cin>>t;
        while(t--){
            ll n;
            cin>>n;
            for(ll i=1;i<=n;i++){
                for(ll j=1;j<=n;j++){
                    cin>>dist[i][j];
                }
            }
            vector<ll>del_oder(n);
            for(ll i=0;i<n;i++){
                cin>>del_oder[i];
            }
            reverse(del_oder.begin(),del_oder.end());
            vector<ll>ans;
            for(ll k=0;k<n;k++){
                ll k_v=del_oder[k];
                for(ll i=1;i<=n;i++){
                    for(ll j=1;j<=n;j++){
                        dist[i][j]=min(dist[i][j],dist[i][k_v]+dist[k_v][j]);
                    }
                }
                ll sum=0;
                for(ll i=0;i<=k;i++){
                    for(ll j=0;j<=k;j++){
                        sum+=dist[del_oder[i]][del_oder[j]];
                    }
                }
                ans.push_back(sum);
            }
            reverse(ans.begin(),ans.end());
            for(auto it:ans){
                cout<<it<<" ";
            }
            cout<<endl;
        }
        return 0;
    }
}

```

## Problem-2:

Link: <https://cses.fi/problemset/task/1672>

## Solution:

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

using namespace std;

#define ll          long long
#define scl(n)      scanf("%lld", &n)
#define fr(i,n)     for (ll i=0;i<n;i++)
#define fr1(i,n)    for(ll i=1;i<=n;i++)
#define pfl(x)      printf("%lld\n",x)
#define endl        "\n"
#define pb          push_back
#define asort(a)     sort(a,a+n)
#define dsort(a)     sort(a,a+n,greater<int>())
#define vasort(v)    sort(v.begin(), v.end());
#define vdsort(v)    sort(v.begin(), v.end(),greater<ll>());
#define pn          printf("\n")
#define md          10000007
#define debug       printf("I am here\n")

#define l(s)         s.size()
#define tcas(i,t)    for(ll i=1;i<=t;i++)
#define pcas(i)      printf("Case %lld: ",i)
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
const ll maxN=1e3+10;
const ll INF=1000000000000;
#define M 10000

void setIO(){
    #ifndef ONLINE_JUDGE
        freopen("input.txt", "r", stdin);

        freopen("output.txt", "w", stdout);
    #endif // ONLINE_JUDGE
}
```

```

ll dist[maxN][maxN];
int main()
{
    fast;
    ll t;
    //setIO();
    //ll tno=1;;
    t=1;
    //cin>>t;

    while(t--){
        ll n,m,q;
        cin>>n>>m>>q;
        for(ll i=1;i<=n;i++){
            for(ll j=1;j<=n;j++){
                dist[i][j]=INF;
            }
        }
        ll f,w,e;
        while(m--){
            cin>>f>>w>>e;
            dist[w][w]=dist[f][f]=0;
            dist[f][w]=min(dist[f][w],e);
            dist[w][f]=min(dist[w][f],e);
        }
        for(ll k=1;k<=n;k++){
            for(ll i=1;i<=n;i++){
                for(ll j=1;j<=n;j++){
                    dist[i][j]=min(dist[i][j],dist[i][k]+dist[k][j]);
                }
            }
        }
        ll a,b;
        while(q--){
            cin>>a>>b;
            ll mindist=dist[a][b];

            if(mindist>=INF){
                cout<<-1<<endl;
            }
            else cout<<mindist<<endl;
        }
    }
}

```

```
    return 0;  
}
```

### Problem-3:

Link: <https://codeforces.com/contest/25/problem/C>

### Solution:

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

using namespace std;

#define ll          long long
#define scl(n)      scanf("%lld", &n)
#define fr(i,n)     for (ll i=0;i<n;i++)
#define fr1(i,n)    for(ll i=1;i<=n;i++)
#define pfl(x)      printf("%lld\n",x)
#define endl        "\n"
#define pb          push_back
#define asort(a)     sort(a,a+n)
#define dsort(a)     sort(a,a+n,greater<int>())
#define vasort(v)    sort(v.begin(), v.end());
#define vdsort(v)    sort(v.begin(), v.end(),greater<ll>());
#define pn          printf("\n")
#define md          10000007
#define debug       printf("I am here\n")

#define l(s)         s.size()
#define tcas(i,t)    for(ll i=1;i<=t;i++)
#define pcas(i)      printf("Case %lld: ",i)
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
const ll maxN=1e3+10;
const ll INF=1000000000000;
#define M 10000

void setIO(){
    #ifndef ONLINE_JUDGE
        freopen("input.txt", "r", stdin);

        freopen("output.txt", "w", stdout);
    #endif // ONLINE_JUDGE
}

ll dist[maxN][maxN];
int main()
{
```

```

fast;
ll t;
//setIO();
//ll tno=1;;
t=1;
//cin>>t;

while(t--){
    ll n;
    cin>>n;
    //    for(ll i=1;i<=n;i++){
    //        for(ll j=1;j<=n;j++){
    //            if(i!=j) dist[i][j]=INF;
    //        }
    //    }
    ll len;
    for(ll i=1;i<=n;i++){
        for(ll j=1;j<=n;j++){
            cin>>len;
            dist[i][j]=len;
        }
    }

    ll q;
    cin>>q;
    ll a,b,cost;
    while(q--){
        cin>>a>>b>>cost;
        dist[a][b]=dist[b][a]=min(dist[a][b],cost);

        for(ll i=1;i<=n;i++){
            for(ll j=1;j<=n;j++){
                dist[i][j]=min(dist[i][j],dist[i][a]+dist[a][j]);
            }
        }

        for(ll i=1;i<=n;i++){
            for(ll j=1;j<=n;j++){
                dist[i][j]=min(dist[i][j],dist[i][b]+dist[b][j]);
            }
        }
    }
}

```

```
    }  
    ll sum=0;  
    for(ll i=1;i<=n;i++){  
        for(ll j=i+1;j<=n;j++){  
            sum+=dist[i][j];  
        }  
    }  
  
    cout<<sum<<" ";  
}  
  
}  
  
return 0;  
}
```

## Algorithm -6: Disjoint Set Union(DSU)

### Problem- 1:

Link: <https://codeforces.com/edu/course/2/lesson/7/1/practice/contest/289390/problem/B>

### Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

//VVI
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define pb push_back
#define ll long long
#define ff first
#define ss second
#define SZ(a) (int)a.size()
#define UNIQUE(a) (a).erase(unique(all(a)),(a).end())
#define eb emplace_back
#define mp make_pair

///BIT MANIPULATION

#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))

//LOOPS

#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define Fo(i,k,n) for(i=k;k<n?i<n:i>n;k<n?i+=1:i-=1)

///PRINTING

#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << "," << #y << "=" << y << endl
#define nn '\n'
#define pfl(x) printf("%lld\n",x)
```



```

#define pcas(i)                printf("Case %lld: ",i)
#define Setpre(n) cout<<fixed<<setprecision(n)
#define itr(it, a) for(auto it = a.begin(); it != a.end(); it++)
#define debug                  printf("I am here\n")

///SORTING AND FILLING

#define asort(a)                sort(a,a+n)
#define dsort(a)                sort(a,a+n,greater<int>())
#define vasort(v)               sort(v.begin(), v.end());
#define vdsort(v)               sort(v.begin(), v.end(),greater<ll>());
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define mem(a,b) memset(a,b,sizeof(a))
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))

///CONSTANTS
#define md                      10000007
#define PI 3.1415926535897932384626
const double EPS = 1e-9;
const ll N = 3e5+10;
const ll M = 1e9+7;

///INLINE FUNCTIONS
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/(double)log(base);}
inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x)};
int poplow(ll x){return __builtin_ctzll(x)};
int pophigh(ll x){return 63 - __builtin_clzll(x)};

/// Data structures
typedef unsigned long long ull;
typedef pair<ll, ll>      pll;
typedef vector<ll>        vl;
typedef vector<pll>        vpll;

```

```

typedef vector<vl>      vl;
template <typename T> using PQ = priority_queue<T>;
template <typename T> using QP = priority_queue<T,vector<T>,greater<T>>;

template <typename T> using ordered_set = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
template <typename T,typename R> using ordered_map = tree<T, R , less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
;

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ) { os << it << endl; }
return os; }
    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }
    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }
    string to_string ( const char &c ) { string s; s += c; return "'" + s +
"'; }

```

```

template<typename Type> string to_string( vector<Type> );
template<typename First, typename Second> string to_string( pair<First,
Second> );
template<typename Collection> string to_string( Collection );

template<typename First, typename Second> string to_string( pair<First,
Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
"}"; }

template<typename Type> string to_string( vector<Type> v ) { bool sep =
false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
to_string( x ); } s += "]"; return s; }

template<typename Collection> string to_string( Collection collection ) {
bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
sep = true; s += to_string( x ); } s += "}"; return s; }

void print() { cerr << endl; sep = false; }
template <typename First, typename... Other> void print( First first,
Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
); print( other... ); }

} using namespace io;

/*=====//

SEF04ET

//=====*/

void setIO(){
#ifdef ONLINE_JUDGE
freopen("input.txt", "r", stdin);

freopen("output.txt", "w", stdout);
#endif // ONLINE_JUDGE
}

struct custom_hash {
static uint64_t splitmix64(uint64_t x) {
x += 0x9e3779b97f4a7c15;
x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;

```

```

        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};
ll par[N];
ll sz[N];
ll maxm[N];
ll minm[N];
// multiset<int> sizes;

void make(ll v){
    par[v]=v;
    sz[v]=1;
    maxm[v]=v;
    minm[v]=v;
    // sizes.insert(1);
}

int find(ll v){
    if(v==par[v]) return v;
    return par[v]=find(par[v]);
}

// void merge(int a,int b){
//     sizes.erase(sizes.find(sz[a]));
//     sizes.erase(sizes.find(sz[b]));
//     sizes.insert(sz[a]+sz[b]);
// }

void Union(ll a,ll b){
    a=find(a);
    b=find(b);
    if(a!=b){
        if(sz[a]<sz[b]) swap(a,b);
        par[b]=a;
        // merge(a,b);
        sz[a]+=sz[b];
        maxm[a]=max(maxm[a],maxm[b]);
        minm[a]=min(minm[a],minm[b]);
    }
}

```

```

}
int main()
{
    fast;
    ll t;
    //setIO();
    //ll tno=1;;
    t=1;
    //cin>>t;

    while(t--){
        ll n,m;
        cin>>n>>m;
        string s;
        for(ll i=1;i<=n;i++){
            make(i);
        }
        ll u,v;
        while(m--){
            cin>>s;;
            if(s=="union"){
                cin>>u>>v;
                Union(u,v);
            }
            else{
                cin>>u;
                ll k=find(u);
                cout<<minm[k]<<" "<<maxm[k]<<" "<<sz[k]<<endl;
            }
        }

    }

    return 0;
}

```

## Problem-2:

Link: <https://codeforces.com/edu/course/2/lesson/7/1/practice/contest/289390/problem/C>

## Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

//VVI
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);

#define ll long long
#define ff first
#define ss second
#define SZ(a) (int)a.size()
#define UNIQUE(a) (a).erase(unique(all(a)),(a).end())
#define eb emplace_back
#define mp make_pair

///BIT MANIPULATION

#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))

//LOOPS

#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define Fo(i,k,n) for(i=k;k<n?i<n:i>n;k<n?i+=1:i-=1)

///PRINTING

#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << "," << #y << "=" << y << endl
#define nn '\n'
#define pfl(x) printf("%lld\n",x)
#define pcas(i) printf("Case %lld: ",i)
#define Setpre(n) cout<<fixed<<setprecision(n)
```

```

#define itr(it, a) for(auto it = a.begin(); it != a.end(); it++)
#define debug          printf("I am here\n")

///SORTING AND FILLING

#define asort(a)        sort(a,a+n)
#define dsort(a)        sort(a,a+n,greater<int>())
#define vasort(v)       sort(v.begin(), v.end());
#define vdsort(v)       sort(v.begin(), v.end(),greater<ll>());
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define mem(a,b) memset(a,b,sizeof(a))
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))

///CONSTANTS
#define md              10000007
#define PI 3.1415926535897932384626
const double EPS = 1e-9;
const ll N = 2e5+10;
const ll M = 1e9+7;

///INLINE FUNCTIONS
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/((double)log(base));}
inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x)};
int poplow(ll x){return __builtin_ctzll(x)};
int pophigh(ll x){return 63 - __builtin_clzll(x)};

/// Data structures
typedef unsigned long long ull;
typedef pair<ll, ll>      pll;
typedef vector<ll>        vl;
typedef vector<pll>        vpll;
typedef vector<vl>         vvl;
template <typename T> using PQ = priority_queue<T>;

```

```

template <typename T> using QP = priority_queue<T,vector<T>,greater<T>>;

template <typename T> using ordered_set = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
template <typename T,typename R> using ordered_map = tree<T, R , less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
;

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ) { os << it << endl; }
return os; }
    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }
    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }
    string to_string ( const char &c ) { string s; s += c; return "'" + s +
"';' }; }

    template<typename Type> string to_string( vector<Type> );

```



```

    template<typename First, typename Second> string to_string( pair<First,
Second> );
    template<typename Collection> string to_string( Collection );

    template<typename First, typename Second> string to_string( pair<First,
Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
"}"; }
    template<typename Type> string to_string( vector<Type> v ) { bool sep =
false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
to_string( x ); } s += "]"; return s; }
    template<typename Collection> string to_string( Collection collection ) {
bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
sep = true; s += to_string( x ); } s += "}"; return s; }

    void print() { cerr << endl; sep = false; }
    template <typename First, typename... Other> void print( First first,
Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
); print( other... ); }

} using namespace io;

/*=====//

SEFFO4ET

//=====*/

void setIO(){
#ifdef ONLINE_JUDGE
freopen("input.txt", "r", stdin);

freopen("output.txt", "w", stdout);
#endif // ONLINE_JUDGE
}

struct custom_hash {
    static uint64_t splitmix64(uint64_t x) {
        x += 0x9e3779b97f4a7c15;
        x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
};

```

```

    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};

ll par[N];
ll sz[N];
ll points[N];
ll extra[N];
// multiset<int> sizes;

void make(ll v){
    par[v]=v;
    sz[v]=1;

    // sizes.insert(1);
}

ll find(ll v){
    if(v==par[v]) return v;
    return find(par[v]);
}

// void merge(int a,int b){
//     sizes.erase(sizes.find(sz[a]));
//     sizes.erase(sizes.find(sz[b]));
//     sizes.insert(sz[a]+sz[b]);
// }

void Union(ll a,ll b){
    ll pa=find(a);
    ll pb=find(b);
    if(pa!=pb){
        if(sz[pa]<sz[pb]) swap(pa,pb);
        par[pb]=pa;
        // merge(a,b);
        sz[pa]+=sz[pb];

        extra[pb]=points[pb]-points[pa];
    }
}

ll getpoints(ll x){

```

```

        if(par[x]==x){
            return points[x];
        }
        return (extra[x])+getpoints(par[x]);
    }

int main()
{
    fast;
    ll t;
    //setIO();
    //ll tno=1;;
    t=1;
    //cin>>t;

    while(t--){
        ll n,m;
        cin>>n>>m;
        string s;
        for(ll i=0;i<=N;i++){
            make(i);
        }
        ll u,v;
        while(m--){
            cin>>s;
            if(s=="join"){
                cin>>u>>v;
                Union(u,v);
            }
            else if(s=="add"){
                cin>>u>>v;
                // cout<<k<<endl;
                points[find(u)]+=v;
                // cout<<points[k]<<endl;
            }
            else{
                cin>>u;
                // cout<<k<<endl;
                cout<<getpoints(u)<<endl;
            }
        }
    }
}

```

```
    return 0;  
}
```

### Problem-3:

Link: <https://codeforces.com/edu/course/2/lesson/7/1/practice/contest/289390/problem/D>

### Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

//VVI
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define pb push_back
#define ll long long
#define ff first
#define ss second
#define SZ(a) (int)a.size()
#define UNIQUE(a) (a).erase(unique(all(a)),(a).end())
#define eb emplace_back
#define mp make_pair

///BIT MANIPULATION

#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))

//LOOPS

#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define Fo(i,k,n) for(i=k;k<n?i<n:i>n;k<n?i+=1:i-=1)

///PRINTING

#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << "," << #y << "=" << y << endl
#define nn '\n'
#define pfl(x) printf("%lld\n",x)
#define pcas(i) printf("Case %lld: ",i)
#define Setpre(n) cout<<fixed<<setprecision(n)
```

```

#define itr(it, a) for(auto it = a.begin(); it != a.end(); it++)
#define debug          printf("I am here\n")

///SORTING AND FILLING

#define asort(a)          sort(a,a+n)
#define dsort(a)          sort(a,a+n,greater<int>())
#define vasort(v)         sort(v.begin(), v.end());
#define vdsort(v)         sort(v.begin(), v.end(),greater<ll>());
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define mem(a,b) memset(a,b,sizeof(a))
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))

///CONSTANTS
#define md                10000007
#define PI 3.1415926535897932384626
const double EPS = 1e-9;
const ll N = 2e5+10;
const ll M = 1e9+7;

///INLINE FUNCTIONS
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/((double)log(base));}
inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x)};
int poplow(ll x){return __builtin_ctzll(x)};
int pophigh(ll x){return 63 - __builtin_clzll(x)};

/// Data structures
typedef unsigned long long ull;
typedef pair<ll, ll>      pll;
typedef vector<ll>        vl;
typedef vector<pll>        vpll;
typedef vector<vl>         vvl;
template <typename T> using PQ = priority_queue<T>;

```

```

template <typename T> using QP = priority_queue<T,vector<T>,greater<T>>;

template <typename T> using ordered_set = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
template <typename T,typename R> using ordered_map = tree<T, R , less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
;

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ) { os << it << endl; }
return os; }
    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }
    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }
    string to_string ( const char &c ) { string s; s += c; return "'" + s +
"';' }

    template<typename Type> string to_string( vector<Type> );

```

```

    template<typename First, typename Second> string to_string( pair<First,
Second> );
    template<typename Collection> string to_string( Collection );

    template<typename First, typename Second> string to_string( pair<First,
Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
"}"; }
    template<typename Type> string to_string( vector<Type> v ) { bool sep =
false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
to_string( x ); } s += "]"; return s; }
    template<typename Collection> string to_string( Collection collection ) {
bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
sep = true; s += to_string( x ); } s += "}"; return s; }

    void print() { cerr << endl; sep = false; }
    template <typename First, typename... Other> void print( First first,
Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
); print( other... ); }

} using namespace io;

/*=====//

SEFFO4ET

//=====*/

void setIO(){
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);

    freopen("output.txt", "w", stdout);
    #endif // ONLINE_JUDGE
}

struct custom_hash {
    static uint64_t splitmix64(uint64_t x) {
        x += 0x9e3779b97f4a7c15;
        x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
};

```



```

    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};
ll par[N];
ll sz[N];
// multiset<int> sizes;

void make(int v){
    par[v]=v;
    sz[v]=1;
    // sizes.insert(1);
}

int find(int v){
    if(v==par[v]) return v;
    return par[v]=find(par[v]);
}

// void merge(int a,int b){
//     sizes.erase(sizes.find(sz[a]));
//     sizes.erase(sizes.find(sz[b]));
//     sizes.insert(sz[a]+sz[b]);
// }

void Union(int a,int b){
    a=find(a);
    b=find(b);
    if(a!=b){
        if(sz[a]<sz[b]) swap(a,b);
        par[b]=a;
        // merge(a,b);
        sz[a]+=sz[b];
    }
}

int main()
{
    fast;
    ll t;
    //setIO();
    //ll tno=1;;
    t=1;

```

```

//cin>>t;

while(t--){
    ll n,m,k;
    cin>>n>>m>>k;
    for(ll i=0;i<=n;i++){
        make(i);
    }
    ll u,v;
    for(ll i=0;i<m;i++){
        cin>>u>>v;
        // Union(u,v);
    }
    string s;
    vector<pair<ll,pair<ll,ll>>>asks;
    vector<string>ans;
    while(k--){
        cin>>s;
        if(s=="ask"){
            cin>>u>>v;
            // if(find(u)==find(v)){
            //     cout<<"YES"<<endl;

            // }
            // else cout<<"NO"<<endl;
            asks.push_back({1,{u,v}});
        }
        else{
            cin>>u>>v;
            asks.push_back({0,{u,v}});
        }
    }
    reverse(all(asks));
    for(auto it:asks){
        // cout<<it<<endl;
        if(it.first==1){
            u=it.second.first;
            v=it.second.second;
            if(find(u)==find(v)){
                ans.push_back("YES");
            }
            else ans.push_back("NO");
        }
        else{

```

```
        u=it.second.first;
        v=it.second.second;
        Union(u,v);
    }

}

reverse(all(ans));
for(auto it:ans){
    cout<<it<<endl;
}

}

return 0;
}
```

## Algorithm -7: Krushkal's Algorithm(Minimum Spanning Tree)

### Problem-1:

Link: <https://codeforces.com/edu/course/2/lesson/7/2/practice/contest/289391/problem/E>

### Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

//VVI
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define pb push_back
#define ll long long
#define ff first
#define ss second
#define SZ(a) (int)a.size()
#define UNIQUE(a) (a).erase(unique(all(a)),(a).end())
#define eb emplace_back
#define mp make_pair

///BIT MANIPULATION

#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))

//LOOPS

#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define Fo(i,k,n) for(i=k;k<n?i<n:i>n;k<n?i+=1:i-=1)

///PRINTING

#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << ", " << #y << "=" << y << endl
#define nn '\n'
#define pfl(x) printf("%lld\n",x)
```

```

#define pcas(i)                printf("Case %lld: ",i)
#define Setpre(n) cout<<fixed<<setprecision(n)
#define itr(it, a) for(auto it = a.begin(); it != a.end(); it++)
#define debug                  printf("I am here\n")

///SORTING AND FILLING

#define asort(a)                sort(a,a+n)
#define dsort(a)                sort(a,a+n,greater<int>())
#define vasort(v)               sort(v.begin(), v.end());
#define vdsort(v)               sort(v.begin(), v.end(),greater<ll>());
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define mem(a,b) memset(a,b,sizeof(a))
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))

///CONSTANTS
#define md                      10000007
#define PI 3.1415926535897932384626
const double EPS = 1e-9;
const ll N = 2e5+10;
const ll M = 1e9+7;

///INLINE FUNCTIONS
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/(double)log(base);}
inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x)};
int poplow(ll x){return __builtin_ctzll(x)};
int pophigh(ll x){return 63 - __builtin_clzll(x)};

/// Data structures
typedef unsigned long long ull;
typedef pair<ll, ll>      pll;
typedef vector<ll>        vl;
typedef vector<pll>        vpll;

```

```

typedef vector<vl>          vl;
template <typename T> using PQ = priority_queue<T>;
template <typename T> using QP = priority_queue<T,vector<T>,greater<T>>;

template <typename T> using ordered_set = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
template <typename T,typename R> using ordered_map = tree<T, R , less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
;

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ) { os << it << endl; }
return os; }
    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }
    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }
    string to_string ( const char &c ) { string s; s += c; return "'" + s +
"'; }

```

```

template<typename Type> string to_string( vector<Type> );
template<typename First, typename Second> string to_string( pair<First,
Second> );
template<typename Collection> string to_string( Collection );

template<typename First, typename Second> string to_string( pair<First,
Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
"}"; }

template<typename Type> string to_string( vector<Type> v ) { bool sep =
false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
to_string( x ); } s += "]"; return s; }

template<typename Collection> string to_string( Collection collection ) {
bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
sep = true; s += to_string( x ); } s += "}"; return s; }

void print() { cerr << endl; sep = false; }
template <typename First, typename... Other> void print( First first,
Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
); print( other... ); }

} using namespace io;

/*=====//

SEF04ET

//=====*/

void setIO(){
#ifdef ONLINE_JUDGE
freopen("input.txt", "r", stdin);

freopen("output.txt", "w", stdout);
#endif // ONLINE_JUDGE
}

struct custom_hash {
static uint64_t splitmix64(uint64_t x) {
x += 0x9e3779b97f4a7c15;
x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;

```

```

        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};
int par[N];
int sz[N];
// multiset<int> sizes;

void make(int v){
    par[v]=v;
    sz[v]=1;
    // sizes.insert(1);
}

int find(int v){
    if(v==par[v]) return v;
    return par[v]=find(par[v]);
}

// void merge(int a,int b){
//     sizes.erase(sizes.find(sz[a]));
//     sizes.erase(sizes.find(sz[b]));
//     sizes.insert(sz[a]+sz[b]);
// }

void Union(int a,int b){
    a=find(a);
    b=find(b);
    if(a!=b){
        if(sz[a]<sz[b]) swap(a,b);
        par[b]=a;
        // merge(a,b);
        sz[a]+=sz[b];
    }
}

int main()
{
    fast;
    ll t;
    //setIO();

```



```

    //ll tno=1;;
    t=1;
    //cin>>t;

    while(t--){
        vector<pair<ll,pair<ll,ll>> >edges;
        ll n,m;
        cin>>n>>m;
        for(ll i=1;i<=n;i++){
            make(i);
        }
        ll u,v,w;
        while(m--){
            cin>>u>>v>>w;
            // Union(u,v);
            edges.push_back({w,{u,v}});
        }
        sort(all(edges));
        ll ans=0;
        for(auto it:edges){
            u=it.second.first;
            v=it.second.second;
            if(find(u)!=find(v)){
                Union(u,v);
                ans+=it.first;
            }
        }
        cout<<ans<<endl;
    }

    return 0;
}

```

## Problem-2:

Link:

<https://codeforces.com/edu/course/2/lesson/7/2/practice/contest/289391/problem/G>

Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

//VVI
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define pb push_back
#define ll long long
#define ff first
#define ss second
#define SZ(a) (int)a.size()
#define UNIQUE(a) (a).erase(unique(all(a)),(a).end())
#define eb emplace_back
#define mp make_pair

//BIT MANIPULATION

#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))

//LOOPS

#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define Fo(i,k,n) for(i=k;k<n?i<n:i>n;k<n?i+=1:i-=1)

//PRINTING

#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << "," << #y << "=" << y << endl
#define nn '\n'
```

```

#define pfl(x)                printf("%lld\n",x)
#define pcas(i)               printf("Case %lld: ",i)
#define Setpre(n) cout<<fixed<<setprecision(n)
#define itr(it, a) for(auto it = a.begin(); it != a.end(); it++)
#define debug                 printf("I am here\n")

//SORTING AND FILLING

#define asort(a)              sort(a,a+n)
#define dsort(a)              sort(a,a+n,greater<int>())
#define vasort(v)             sort(v.begin(), v.end());
#define vdsort(v)             sort(v.begin(), v.end(),greater<ll>());
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define mem(a,b) memset(a,b,sizeof(a))
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))

//CONSTANTS
#define md                    10000007
#define PI 3.1415926535897932384626
const double EPS = 1e-9;
const ll N = 2e5+10;
const ll M = 1e9+7;

//INLINE FUNCTIONS
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/((double)log(base));}
inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x);};
int poplow(ll x){return __builtin_ctzll(x);};
int pophigh(ll x){return 63 - __builtin_clzll(x);};

// Data structures
typedef unsigned long long ull;
typedef pair<ll, ll>    pll;
typedef vector<ll>      vl;

```

```

typedef vector<pll>      vpll;
typedef vector<vl>      vvl;
template <typename T> using PQ = priority_queue<T>;
template <typename T> using QP = priority_queue<T,vector<T>,greater<T>>;

template <typename T> using ordered_set = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
template <typename T,typename R> using ordered_map = tree<T, R , less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
;

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ) { os << it << endl; }
return os; }
    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }
    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }

```

```

    string to_string ( const char & c ) { string s; s += c; return "\"" + s +
    "\""; }

    template<typename Type> string to_string( vector<Type> );
    template<typename First, typename Second> string to_string( pair<First,
    Second> );
    template<typename Collection> string to_string( Collection );

    template<typename First, typename Second> string to_string( pair<First,
    Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
    "}"; }

    template<typename Type> string to_string( vector<Type> v ) { bool sep =
    false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
    to_string( x ); } s += "]"; return s; }

    template<typename Collection> string to_string( Collection collection ) {
    bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
    sep = true; s += to_string( x ); } s += "}"; return s; }

    void print() { cerr << endl; sep = false; }
    template <typename First, typename... Other> void print( First first,
    Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
    ); print( other... ); }

} using namespace io;

/*=====//

SEFARUET

//=====*/

void setIO(){
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);

    freopen("output.txt", "w", stdout);
    #endif // ONLINE_JUDGE
}

struct custom_hash {
    static uint64_t splitmix64(uint64_t x) {

```

```

        x += 0x9e3779b97f4a7c15;
        x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};
int par[N];
int sz[N];
// multiset<int> sizes;

void make(int v){
    par[v]=v;
    sz[v]=1;
    // sizes.insert(1);
}

int find(int v){
    if(v==par[v]) return v;
    return par[v]=find(par[v]);
}

// void merge(int a,int b){
//     sizes.erase(sizes.find(sz[a]));
//     sizes.erase(sizes.find(sz[b]));
//     sizes.insert(sz[a]+sz[b]);
// }

void Union(int a,int b){
    a=find(a);
    b=find(b);
    if(a!=b){
        if(sz[a]<sz[b]) swap(a,b);
        par[b]=a;
        // merge(a,b);
        sz[a]+=sz[b];
    }
}

int main()
{
    fast;

```

```

    ll t;
    //setIO();
    //ll tno=1;;
    t=1;
    //cin>>t;

    while(t--){
        vector<pair<ll,pair<ll,ll>> >edges;
        ll n,m;
        cin>>n>>m;
        for(ll i=1;i<=n;i++){
            make(i);
        }
        ll u,v,w;
        while(m--){
            cin>>u>>v>>w;
            // Union(u,v);
            edges.push_back({w,{u,v}});
        }
        sort(all(edges));
        ll ans=0;
        for(auto it:edges){
            u=it.second.first;
            v=it.second.second;
            if(find(u)!=find(v)){
                Union(u,v);
                ans=it.first;
            }
        }
        cout<<ans<<endl;
    }

    return 0;
}

```

### Problem-3:

Link: <https://codeforces.com/edu/course/2/lesson/7/2/practice/contest/289391/problem/H>

### Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

//VVI
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define pb push_back
#define ll long long
#define ff first
#define ss second
#define SZ(a) (int)a.size()
#define UNIQUE(a) (a).erase(unique(all(a)),(a).end())
#define eb emplace_back
#define mp make_pair

///BIT MANIPULATION

#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))

//LOOPS

#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define Fo(i,k,n) for(i=k;k<n?i<n:i>n;k<n?i+=1:i-=1)

///PRINTING

#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << "," << #y << "=" << y << endl
#define nn '\n'
#define pfl(x) printf("%lld\n",x)
#define pcas(i) printf("Case %lld: ",i)
```



```

#define Setpre(n) cout<<fixed<<setprecision(n)
#define itr(it, a) for(auto it = a.begin(); it != a.end(); it++)
#define debug          printf("I am here\n")

///SORTING AND FILLING

#define asort(a)          sort(a,a+n)
#define dsort(a)          sort(a,a+n,greater<int>())
#define vasort(v)         sort(v.begin(), v.end());
#define vdsort(v)         sort(v.begin(), v.end(),greater<ll>());
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define mem(a,b) memset(a,b,sizeof(a))
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))

///CONSTANTS
#define md                10000007
#define PI 3.1415926535897932384626
const double EPS = 1e-9;
const ll N = 2e5+10;
const ll M = 1e9+7;

///INLINE FUNCTIONS
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/(double)log(base);}
inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x)};
int poplow(ll x){return __builtin_ctzll(x)};
int pophigh(ll x){return 63 - __builtin_clzll(x)};

/// Data structures
typedef unsigned long long ull;
typedef pair<ll, ll>      pll;
typedef vector<ll>        vl;
typedef vector<pll>       vpll;
typedef vector<vl>        vvl;

```

```

template <typename T> using PQ = priority_queue<T>;
template <typename T> using QP = priority_queue<T,vector<T>,greater<T>>;

template <typename T> using ordered_set = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
template <typename T,typename R> using ordered_map = tree<T, R , less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
;

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ) { os << it << endl; }
return os; }
    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }
    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }
    string to_string ( const char &c ) { string s; s += c; return "'" + s +
"'"; }

```

```

template<typename Type> string to_string( vector<Type> );
template<typename First, typename Second> string to_string( pair<First,
Second> );
template<typename Collection> string to_string( Collection );

template<typename First, typename Second> string to_string( pair<First,
Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
"}"; }

template<typename Type> string to_string( vector<Type> v ) { bool sep =
false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
to_string( x ); } s += "]"; return s; }

template<typename Collection> string to_string( Collection collection ) {
bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
sep = true; s += to_string( x ); } s += "}"; return s; }

void print() { cerr << endl; sep = false; }
template <typename First, typename... Other> void print( First first,
Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
); print( other... ); }

} using namespace io;

/*=====//

SEFAR4ET

//=====*/

void setIO(){
#ifdef ONLINE_JUDGE
freopen("input.txt", "r", stdin);

freopen("output.txt", "w", stdout);
#endif // ONLINE_JUDGE
}

struct custom_hash {
static uint64_t splitmix64(uint64_t x) {
x += 0x9e3779b97f4a7c15;
x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
x = (x ^ (x >> 27)) * 0x94d049bb133111eb;

```

```

        return x ^ (x >> 31);
    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};
int par[N];
int sz[N];
// multiset<int> sizes;

void make(int v){
    par[v]=v;
    sz[v]=1;
    // sizes.insert(1);
}

int find(int v){
    if(v==par[v]) return v;
    return par[v]=find(par[v]);
}

// void merge(int a,int b){
//     sizes.erase(sizes.find(sz[a]));
//     sizes.erase(sizes.find(sz[b]));
//     sizes.insert(sz[a]+sz[b]);
// }

void Union(int a,int b){
    a=find(a);
    b=find(b);
    if(a!=b){
        if(sz[a]<sz[b]) swap(a,b);
        par[b]=a;
        // merge(a,b);
        sz[a]+=sz[b];
    }
}

bool cmp(vector<ll>a,vector<ll>b){
    return a[0]>b[0];
}

int main()
{
    fast;

```

```

    ll t;
    //setIO();
    //ll tno=1;;
    t=1;
    //cin>>t;

    while(t--){
        vector<vector<ll>>>edges;
        ll n,m,s;
        cin>>n>>m>>s;
        for(ll i=1;i<=n;i++){
            make(i);
        }
        ll u,v,w;
        for(ll i=0;i<m;i++){
            cin>>u>>v>>w;
            // Union(u,v);
            vector<ll>a={w,u,v,i+1};
            edges.push_back(a);
        }
        sort(all(edges),cmp);

        ll ans=0;
        vector<ll>answer;
        ll curr=0;
        vector<ll>vis(m+1,0);
        for(ll i=0;i<m;i++){
            u=edges[i][1];
            v=edges[i][2];
            if(find(u)!=find(v)){
                Union(u,v);
                vis[i]=1;
            }
            // else{
            //     if(curr+edges[i].first<=s)
            {answer.push_back(i+1);curr+=edges[i].first;}
            // }
        }
        for(ll i=m-1;i>=0;i--){
            if(!vis[i]){
                if (s >= edges[i][0])
                {
                    s -= edges[i][0];
                    ans++;
                    answer.push_back(edges[i][3]);
                }
            }
        }
    }
}

```

```
        }  
    }  
}  
  
cout<<ans<<nn<<answer<<nn;  
}  
  
return 0;  
}
```

## Algorithm -8: Bipartite Graph

### Problem-1:

Link: <https://codeforces.com/edu/course/2/lesson/7/2/practice/contest/289391/problem/1>

### Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

//VVI
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define pb push_back
#define ll long long
#define ff first
#define ss second
#define SZ(a) (int)a.size()
#define UNIQUE(a) (a).erase(unique(all(a)),(a).end())
#define eb emplace_back
#define mp make_pair

///BIT MANIPULATION

#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))

//LOOPS

#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define Fo(i,k,n) for(i=k;k<n?i<n:i>n;k<n?i+=1:i-=1)

///PRINTING

#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << "," << #y << "=" << y << endl
#define nn '\n'
#define pfl(x) printf("%lld\n",x)
```

```

#define pcas(i)                printf("Case %lld: ",i)
#define Setpre(n) cout<<fixed<<setprecision(n)
#define itr(it, a) for(auto it = a.begin(); it != a.end(); it++)
#define debug                  printf("I am here\n")

///SORTING AND FILLING

#define asort(a)                sort(a,a+n)
#define dsort(a)                sort(a,a+n,greater<int>())
#define vasort(v)               sort(v.begin(), v.end());
#define vdsort(v)               sort(v.begin(), v.end(),greater<ll>());
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define mem(a,b) memset(a,b,sizeof(a))
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))

///CONSTANTS
#define md                      10000007
#define PI 3.1415926535897932384626
const double EPS = 1e-9;
const ll N = 2e5+10;
const ll M = 1e9+7;

///INLINE FUNCTIONS
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/(double)log(base);}
inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x)};
int poplow(ll x){return __builtin_ctzll(x)};
int pophigh(ll x){return 63 - __builtin_clzll(x)};

/// Data structures
typedef unsigned long long ull;
typedef pair<ll, ll>          pll;
typedef vector<ll>            vl;
typedef vector<pll>           vpll;

```



```

typedef vector<vl>      vl;
template <typename T> using PQ = priority_queue<T>;
template <typename T> using QP = priority_queue<T,vector<T>,greater<T>>;

template <typename T> using ordered_set = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
template <typename T,typename R> using ordered_map = tree<T, R , less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
;

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ) { os << it << endl; }
return os; }
    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }
    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }
    string to_string ( const char &c ) { string s; s += c; return "'" + s +
"'; }

```

```

template<typename Type> string to_string( vector<Type> );
template<typename First, typename Second> string to_string( pair<First,
Second> );
template<typename Collection> string to_string( Collection );

template<typename First, typename Second> string to_string( pair<First,
Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
"}"; }

template<typename Type> string to_string( vector<Type> v ) { bool sep =
false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
to_string( x ); } s += "]"; return s; }

template<typename Collection> string to_string( Collection collection ) {
bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
sep = true; s += to_string( x ); } s += "}"; return s; }

void print() { cerr << endl; sep = false; }
template <typename First, typename... Other> void print( First first,
Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
); print( other... ); }

} using namespace io;

/*=====//

SEF04ET

//=====*/

void setIO(){
#ifdef ONLINE_JUDGE
freopen("input.txt", "r", stdin);

freopen("output.txt", "w", stdout);
#endif // ONLINE_JUDGE
}

struct custom_hash {
static uint64_t splitmix64(uint64_t x) {
x += 0x9e3779b97f4a7c15;
x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;

```

```

        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};
ll par[N];
ll sz[N];
ll len[N];
// multiset<int> sizes;

void make(ll v){
    par[v]=v;
    sz[v]=1;
    len[v]=0;
    // sizes.insert(1);
}

pair<ll,ll> find(ll v){
    if(v==par[v]) return {v,0};
    auto val=find(par[v]);
    par[v] = val.first;
    len[v] = (len[v] + val.second) % 2;
    return {par[v], len[v]};
}

// void merge(int a,int b){
//     sizes.erase(sizes.find(sz[a]));
//     sizes.erase(sizes.find(sz[b]));
//     sizes.insert(sz[a]+sz[b]);
// }

void Union(ll A,ll B){
    auto a=find(A);
    auto b=find(B);
    if(a!=b){
        if(sz[a.first]>=sz[b.first]) swap(a,b);
        par[a.first]=b.first;
        // merge(a,b);
        sz[b.first]+=sz[a.first];
        len[a.first] = (1 +a.second + b.second) %2;
    }
}

```

```

}
int main()
{
    fast;
    ll t;
    //setIO();
    //ll tno=1;;
    t=1;
    //cin>>t;

    while(t--){
        ll n,q,shift=0;
        cin>>n>>q;
        for(ll i=1;i<=n;i++){
            make(i);
        }
        ll u,x,y;
        while(q--){
            cin>>u>>x>>y;
            x=(x+shift)%n;
            y=(y+shift)%n;
            if(u==0){
                //union of x and y node
                Union(x,y);
                // cout<<len[x]<<" "<<len[y]<<endl;
            }
            else{
                // check if color of these two nodes are same
                // cout<<len[x]<<" "<<len[y]<<endl;
                auto px=find(x);
                auto py=find(y);
                // cout<<len[px.first]<<" "<<len[py.first]<<endl;
                // cout<<px.second<<" "<<py.second<<endl;
                if(px.second==py.second){
                    cout<<"YES"<<nn;
                    shift=(shift+1)%n;
                }
                else{
                    cout<<"NO"<<nn;
                }
            }
        }
    }
}

```

```
    return 0;  
}
```

## Problem-2:

Link: <https://codeforces.com/edu/course/2/lesson/7/2/practice/contest/289391/problem/J>

## Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

//VVI
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define pb push_back
#define ll long long
#define ff first
#define ss second
#define SZ(a) (int)a.size()
#define UNIQUE(a) (a).erase(unique(all(a)),(a).end())
#define eb emplace_back
#define mp make_pair

///BIT MANIPULATION

#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))

//LOOPS

#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define Fo(i,k,n) for(i=k;k<n?i<n:i>n;k<n?i+=1:i-=1)

///PRINTING

#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << "," << #y << "=" << y << endl
#define nn '\n'
#define pfl(x) printf("%lld\n",x)
#define pcas(i) printf("Case %lld: ",i)
```

```

#define Setpre(n) cout<<fixed<<setprecision(n)
#define itr(it, a) for(auto it = a.begin(); it != a.end(); it++)
#define debug          printf("I am here\n")

///SORTING AND FILLING

#define asort(a)          sort(a,a+n)
#define dsort(a)          sort(a,a+n,greater<int>())
#define vasort(v)         sort(v.begin(), v.end());
#define vdsort(v)         sort(v.begin(), v.end(),greater<ll>());
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define mem(a,b) memset(a,b,sizeof(a))
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))

///CONSTANTS
#define md                10000007
#define PI 3.1415926535897932384626
const double EPS = 1e-9;
const ll N = 3e5+10;
const ll M = 1e9+7;

///INLINE FUNCTIONS
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/(double)log(base);}
inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x)};
int poplow(ll x){return __builtin_ctzll(x)};
int pophigh(ll x){return 63 - __builtin_clzll(x)};

/// Data structures
typedef unsigned long long ull;
typedef pair<ll, ll>      pll;
typedef vector<ll>        vl;
typedef vector<pll>       vpll;
typedef vector<vl>        vvl;

```

```

template <typename T> using PQ = priority_queue<T>;
template <typename T> using QP = priority_queue<T,vector<T>,greater<T>>;

template <typename T> using ordered_set = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
template <typename T,typename R> using ordered_map = tree<T, R , less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
;

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ) { os << it << endl; }
return os; }
    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }
    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }
    string to_string ( const char &c ) { string s; s += c; return "'" + s +
"'"; }

```



```

template<typename Type> string to_string( vector<Type> );
template<typename First, typename Second> string to_string( pair<First,
Second> );
template<typename Collection> string to_string( Collection );

template<typename First, typename Second> string to_string( pair<First,
Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
"}"; }

template<typename Type> string to_string( vector<Type> v ) { bool sep =
false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
to_string( x ); } s += "]"; return s; }

template<typename Collection> string to_string( Collection collection ) {
bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
sep = true; s += to_string( x ); } s += "}"; return s; }

void print() { cerr << endl; sep = false; }
template <typename First, typename... Other> void print( First first,
Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
); print( other... ); }

} using namespace io;

/*=====//

SEF04ET

//=====*/

void setIO(){
#ifdef ONLINE_JUDGE
freopen("input.txt", "r", stdin);

freopen("output.txt", "w", stdout);
#endif // ONLINE_JUDGE
}

struct custom_hash {
static uint64_t splitmix64(uint64_t x) {
x += 0x9e3779b97f4a7c15;
x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
x = (x ^ (x >> 27)) * 0x94d049bb133111eb;

```

```

        return x ^ (x >> 31);
    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};
ll par[N];
ll sz[N];
ll len[N];
// multiset<int> sizes;

void make(ll v){
    par[v]=v;
    sz[v]=1;
    len[v]=0;
    // sizes.insert(1);
}

pair<ll,ll> find(ll v){
    if(v==par[v]) return {v,0};
    auto val=find(par[v]);
    par[v] = val.first;
    len[v] = (len[v] + val.second) % 2;
    return {par[v], len[v]};
}

// void merge(int a,int b){
//     sizes.erase(sizes.find(sz[a]));
//     sizes.erase(sizes.find(sz[b]));
//     sizes.insert(sz[a]+sz[b]);
// }

bool Union(ll A,ll B){
    auto a=find(A);
    auto b=find(B);
    if(a.first==b.first) return a.second==b.second;
    if(a!=b){
        if(sz[a.first]>=sz[b.first]) swap(a,b);
        par[a.first]=b.first;
        // merge(a,b);
        sz[b.first]+=sz[a.first];
        len[a.first] = (1 +a.second + b.second) %2;
    }
}

```

```

        return false;
    }
}
int main()
{
    fast;
    ll t;
    //setIO();
    //ll tno=1;;
    t=1;
    //cin>>t;

    while(t--){
        ll n,q,shift=0;
        cin>>n>>q;
        for(ll i=1;i<=n;i++){
            make(i);
        }
        ll u,x,y;
        bool f=0;
        for(ll i=1;i<=q;i++){
            cin>>x>>y;
            if(Union(x,y)){
                cout<<i<<nn;
                f=1;
                break;
            }

        }
        if(!f) cout<<-1<<nn;
    }

    return 0;
}

```

### Problem-3:

Link: <https://cses.fi/problemset/result/4932262/>

### Solution:

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

using namespace std;

#define ll          long long
#define scl(n)      scanf("%lld", &n)
#define fr(i,n)     for (ll i=0;i<n;i++)
#define fr1(i,n)    for(ll i=1;i<=n;i++)
#define pfl(x)      printf("%lld\n",x)
#define endl        "\n"
#define pb          push_back
#define asort(a)     sort(a,a+n)
#define dsort(a)     sort(a,a+n,greater<int>())
#define vasort(v)    sort(v.begin(), v.end());
#define vdsort(v)    sort(v.begin(), v.end(),greater<ll>());
#define pn          printf("\n")
#define md          10000007
#define debug       printf("I am here\n")

#define l(s)         s.size()
#define tcas(i,t)    for(ll i=1;i<=t;i++)
#define pcas(i)      printf("Case %lld: ",i)
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);

const ll maxN=2e5+10;//for graph
#define M 10000

vector<ll>g[maxN];
ll vis[maxN];
ll col[maxN];
ll par[maxN];
bool f=0;
void dfs(ll vertex){

    vis[vertex]=1;
```

```

for(ll child:g[vertex]){
    if(!vis[child]){
        if(col[vertex]==1) col[child]=2;
        else if(col[vertex]==2) col[child]=1;

        par[child]=vertex;
        dfs(child);
    }
    else{
        if(col[child]==col[vertex]) f=1;
    }
}

}
}

```

```

int main()
{
    fast;
    ll t;
    //setIO();
    //ll tno=1;;
    t=1;
    //cin>>t;

    while(t--){
        ll n,e;
        cin>>n>>e;
        ll v1,v2;

        for(ll i=0;i<e;i++){
            cin>>v1>>v2;
            g[v1].push_back(v2);
            g[v2].push_back(v1);
        }
        ll cnt=0;

        for(ll i=1;i<=n;i++){
            if(!vis[i]){
                col[i]=1;
                dfs(i);
                cnt++;
            }
        }
    }
}

```

```
    }  
    // cout<<cnt<<endl;  
    if(f) cout<<"IMPOSSIBLE"<<endl;  
    else{  
        for(ll i=1;i<=n;i++){  
            cout<<col[i]<<" ";  
        }  
        cout<<endl;  
    }  
}  
  
return 0;  
}
```

## Algorithm -9: Binary Indexed Tree

### Problem- 1:

Link: <https://codeforces.com/problemset/problem/459/D>

### Solution:

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

// VVI
#define fast \
    ios_base::sync_with_stdio(0); \
    cin.tie(0); \
    cout.tie(0);
#define pb push_back
#define ll long long
#define ff first
#define ss second
#define SZ(a) (int)a.size()
#define UNIQUE(a) (a).erase(unique(all(a)), (a).end())
#define eb emplace_back
#define mp make_pair

/// BIT MANIPULATION

#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))

// LOOPS

#define scl(n) scanf("%lld", &n)
#define fr(i, n) for (ll i = 0; i < n; i++)
#define fr1(i, n) for (ll i = 1; i <= n; i++)
#define Fo(i, k, n) for (i = k; k < n ? i < n : i > n; k < n ? i += 1 : i -= 1)

/// PRINTING

#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << ", " << #y << "=" << y << endl
```

```

#define nn '\n'
#define pfl(x) printf("%lld\n", x)
#define pcas(i) printf("Case %lld: ", i)
#define Setpre(n) cout << fixed << setprecision(n)
#define itr(it, a) for (auto it = a.begin(); it != a.end(); it++)
#define debug printf("I am here\n")

/// SORTING AND FILLING

#define asort(a) sort(a, a + n)
#define dsort(a) sort(a, a + n, greater<int>())
#define vasort(v) sort(v.begin(), v.end());
#define vdsort(v) sort(v.begin(), v.end(), greater<ll>());
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define mem(a, b) memset(a, b, sizeof(a))
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))

// CONSTANTS
#define md 10000007
#define PI 3.1415926535897932384626
const double EPS = 1e-9;
const ll N = 1e6 + 3;
const ll M = 1e9 + 7;

/// INLINE FUNCTIONS
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) { return p < 0 ? p / q : p / q + !(p % q); }
inline ll Floor(ll p, ll q) { return p > 0 ? p / q : p / q - !(p % q); }
inline double logb(ll base, ll num) { return (double)log(num) /
(double)log(base); }
inline bool isPerfectSquare(long double x)
{
    if (x >= 0)
    {
        long long sr = sqrt(x);
        return (sr * sr == x);
    }
    return false;
}
double euclidean_distance(ll x1, ll y1, ll x2, ll y2)
{
    double a = (x2 - x1) * (x2 - x1);

```



```

    double b = (y2 - y1) * (y2 - y1);
    double c = (double)sqrt(a + b);
    return c;
}
int popcount(ll x) { return __builtin_popcountll(x); };
int poplow(ll x) { return __builtin_ctzll(x); };
int pophigh(ll x) { return 63 - __builtin_clzll(x); };

/// Data structures
typedef unsigned long long ull;
typedef pair<ll, ll> pll;
typedef vector<ll> vl;
typedef vector<pll> vpll;
typedef vector<vl> vvl;
template <typename T>
using PQ = priority_queue<T>;
template <typename T>
using QP = priority_queue<T, vector<T>, greater<T>>;

template <typename T>
using ordered_set = tree<T, null_type, less<T>, rb_tree_tag,
tree_order_statistics_node_update>;
template <typename T, typename R>
using ordered_map = tree<T, R, less<T>, rb_tree_tag,
tree_order_statistics_node_update>;
;

namespace io
{
    template <typename First, typename Second>
    ostream &operator<<(ostream &os, const pair<First, Second> &p) { return os <<
p.first << " " << p.second; }
    template <typename First, typename Second>
    ostream &operator<<(ostream &os, const map<First, Second> &mp)
    {
        for (auto it : mp)
        {
            os << it << endl;
        }
        return os;
    }
    template <typename First>
    ostream &operator<<(ostream &os, const vector<First> &v)
    {
        bool space = false;

```

```

        for (First x : v)
        {
            if (space)
                os << " ";
            space = true;
            os << x;
        }
        return os;
    }

template <typename First>
ostream &operator<<(ostream &os, const set<First> &st)
{
    bool space = false;
    for (First x : st)
    {
        if (space)
            os << " ";
        space = true;
        os << x;
    }
    return os;
}

template <typename First>
ostream &operator<<(ostream &os, const multiset<First> &st)
{
    bool space = false;
    for (First x : st)
    {
        if (space)
            os << " ";
        space = true;
        os << x;
    }
    return os;
}

template <typename First, typename Second>
istream &operator>>(istream &is, pair<First, Second> &p) { return is >>
p.first >> p.second; }

template <typename First>
istream &operator>>(istream &is, vector<First> &v)
{
    for (First &x : v)
    {
        is >> x;
    }
}

```

```

    }
    return is;
}

long long fastread()
{
    char c;
    long long d = 1, x = 0;
    do
        c = getchar();
    while (c == ' ' || c == '\n');
    if (c == '-')
        c = getchar(), d = -1;
    while (isdigit(c))
    {
        x = x * 10 + c - '0';
        c = getchar();
    }
    return d * x;
}

static bool sep = false;

using std::to_string;

string to_string(bool x) { return (x ? "true" : "false"); }
string to_string(const string &s) { return "\"" + s + "\""; }
string to_string(const char *s) { return "\"" + string(s) + "\""; }
string to_string(const char &c)
{
    string s;
    s += c;
    return "'" + s + "'";
}

template <typename Type>
string to_string(vector<Type>);
template <typename First, typename Second>
string to_string(pair<First, Second>);
template <typename Collection>
string to_string(Collection);

template <typename First, typename Second>
string to_string(pair<First, Second> p) { return "{" + to_string(p.first) +
", " + to_string(p.second) + "}"; }

```

```

template <typename Type>
string to_string(vector<Type> v)
{
    bool sep = false;
    string s = "[";
    for (Type x : v)
    {
        if (sep)
            s += ", ";
        sep = true;
        s += to_string(x);
    }
    s += "]";
    return s;
}

template <typename Collection>
string to_string(Collection collection)
{
    bool sep = false;
    string s = "{";
    for (auto x : collection)
    {
        if (sep)
            s += ", ";
        sep = true;
        s += to_string(x);
    }
    s += "}";
    return s;
}

void print()
{
    cerr << endl;
    sep = false;
}

template <typename First, typename... Other>
void print(First first, Other... other)
{
    if (sep)
        cerr << " | ";
    sep = true;
    cerr << to_string(first);
    print(other...);
}

```

```

}
using namespace io;

/*=====//

SEFFA4ET

//=====*/

void setIO()
{
#ifdef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);

    freopen("output.txt", "w", stdout);
#endif // ONLINE_JUDGE
}

struct custom_hash
{
    static uint64_t splitmix64(uint64_t x)
    {
        x += 0x9e3779b97f4a7c15;
        x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
    size_t operator()(uint64_t x) const
    {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};

ll BITree[4*N];

//do this for range: getSum(r) - getSum(l - 1)
ll getSum(ll index){
    ll sum = 0; // Iniiialize result
    // Traverse ancestors of BITree[index]
    while (index>0){
        sum += BITree[index]; // Add current element of BITree to sum
    }
}

```

```

        index -= index & (-index); // Move index to parent node in getSum View
    }
    return sum;
}

void updateBIT(ll n, ll index, ll val){
    // Traverse all ancestors and add 'val'
    while (index <= n){
        // Add 'val' to current node of BI Tree
        BITree[index] += val;
        // Update index to that of parent in update View
        index += index & (-index);
    }
}

ll ans;
ll k,m,g,f;
int main()
{
    fast;
    ll t;
    // setIO();
    // ll tno=1;;
    t = 1;
    // cin>>t;

    while (t--)
    {
        ll n,i;
        cin >> n;

        vector<ll> vec(n),x(n), y(n), z(n + 1);
        map<ll, ll> freq, freq2, mpp;

        cin >> vec;

        mem(BITree,0);

        for ( i = 0; i < n; i++)
        {
            freq[vec[i]]++;
            x[i] = freq[vec[i]];
        }

        // buildSegTree(x,0,0,n-1);

```

```

for ( i = n - 1; i >= 0; i--)
{
    freq2[vec[i]]++;
    y[i] = freq2[vec[i]];
}
//    cout<<x<<endl<<y<<endl;

for ( i = 0; i < n; i++)
{
    mpp[y[i]]++;
}
for ( i = 0; i <= n; i++)
{
    updateBIT(n+1,i+1,mpp[i]);
}

ans=0;
for (i = 0; i < n ; i++)
{
    k = x[i];
    m = y[i];
    //  g = queryLazySegTree(0, 0, n, m, m);
    g=getSum(m+1);
    // ll g=querySegTree(0,0,n,m,m);

    // updateValSegTree(0,0,n,m,g-1);
    // updateLazySegTree(0, 0, n, m, m, -1);
    updateBIT(n+1,m+1,-1);

    // g=querySegTree(0,0,n,m,m);
    // cout<<g<<endl;
    // ll f = querySegTree(0, 0, n, 0, k - 1)
    f= getSum(k)-getSum(0);
    // queryLazySegTree(0,0,n,0,k-1);
    // cout<<k<<" "<<f<<endl;
    ans += f;
}
//    cout<<endl;

cout << ans << endl;
}

return 0;
}

```

## Problem- 2:

Link: <https://www.spoj.com/problems/INVCNT/>

## Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

//WVI
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define pb push_back
#define ll long long
#define ff first
#define ss second
#define SZ(a) (int)a.size()
#define UNIQUE(a) (a).erase(unique(all(a)),(a).end())
#define eb emplace_back
#define mp make_pair

///BIT MANIPULATION

#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))

//LOOPS

#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define Fo(i,k,n) for(i=k;k<n?i<n:i>n;k<n?i+=1:i-=1)

///PRINTING

#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << "," << #y << "=" << y << endl
#define nn '\n'
#define pfl(x) printf("%lld\n",x)
#define pcas(i) printf("Case %lld: ",i)
#define Setpre(n) cout<<fixed<<setprecision(n)
```



```

#define itr(it, a) for(auto it = a.begin(); it != a.end(); it++)
#define debug          printf("I am here\n")

///SORTING AND FILLING

#define asort(a)          sort(a,a+n)
#define dsort(a)          sort(a,a+n,greater<int>())
#define vasort(v)         sort(v.begin(), v.end());
#define vdsort(v)         sort(v.begin(), v.end(),greater<ll>());
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define mem(a,b) memset(a,b,sizeof(a))
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))

///CONSTANTS
#define md                10000007
#define PI 3.1415926535897932384626
const double EPS = 1e-9;
const ll N = 1e7+10;
const ll M = 1e9+7;

///INLINE FUNCTIONS
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/((double)log(base));}
inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x)};
int poplow(ll x){return __builtin_ctzll(x)};
int pophigh(ll x){return 63 - __builtin_clzll(x)};

/// Data structures
typedef unsigned long long ull;
typedef pair<ll, ll>      pll;
typedef vector<ll>        vl;
typedef vector<pll>        vpll;
typedef vector<vl>         vvl;
template <typename T> using PQ = priority_queue<T>;

```

```

template <typename T> using QP = priority_queue<T,vector<T>,greater<T>>;

template <typename T> using ordered_set = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
template <typename T,typename R> using ordered_map = tree<T, R , less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
;

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ) { os << it << endl; }
return os; }
    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }
    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }
    string to_string ( const char &c ) { string s; s += c; return "'" + s +
"';' }; }

    template<typename Type> string to_string( vector<Type> );

```

```

    template<typename First, typename Second> string to_string( pair<First,
Second> );
    template<typename Collection> string to_string( Collection );

    template<typename First, typename Second> string to_string( pair<First,
Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
"}"; }
    template<typename Type> string to_string( vector<Type> v ) { bool sep =
false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
to_string( x ); } s += "]"; return s; }
    template<typename Collection> string to_string( Collection collection ) {
bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
sep = true; s += to_string( x ); } s += "}"; return s; }

    void print() { cerr << endl; sep = false; }
    template <typename First, typename... Other> void print( First first,
Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
); print( other... ); }

} using namespace io;

/*=====//

SEFFO4ET

//=====*/

void setIO(){
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);

    freopen("output.txt", "w", stdout);
    #endif // ONLINE_JUDGE
}

struct custom_hash {
    static uint64_t splitmix64(uint64_t x) {
        x += 0x9e3779b97f4a7c15;
        x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
};

```

```

    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};
ll BITree[4*N];

//do this for range: getSum(r) - getSum(l - 1)
ll getSum(ll index){
    ll sum = 0; // Initialize result
    // Traverse ancestors of BITree[index]
    while (index>0){
        sum += BITree[index]; // Add current element of BITree to sum
        index -= index & (-index); // Move index to parent node in getSum View
    }
    return sum;
}

void updateBIT(ll n, ll index, ll val){
    // Traverse all ancestors and add 'val'
    while (index <= n){
        // Add 'val' to current node of BI Tree
        BITree[index] += val;
        // Update index to that of parent in update View
        index += index & (-index);
    }
}

int main()
{
    fast;
    ll t;
    //setIO();
    //ll tno=1;;
    t=1;
    cin>>t;

    while(t--){
        ll n;
        cin>>n;
        vector<ll>vec(n);
        cin>>vec;
        mem(BITree,0);
        ll maxm=*max_element(all(vec));
    }
}

```

```
    ll cnt=0;
    for(ll i=1;i<=n;i++){
        cnt+=getSum(maxm)-getSum(vec[i-1]);
        updateBIT(N,vec[i-1],1);
    }
    cout<<cnt<<nn;

}

return 0;
}
```

### Problem- 3:

Link: <https://www.spoj.com/problems/NICEDAY/>

### Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

//VVI
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define pb push_back
#define ll long long
#define ff first
#define ss second
#define SZ(a) (int)a.size()
#define UNIQUE(a) (a).erase(unique(all(a)),(a).end())
#define eb emplace_back
#define mp make_pair

///BIT MANIPULATION

#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))

//LOOPS

#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define Fo(i,k,n) for(i=k;k<n?i<n:i>n;k<n?i+=1:i-=1)

///PRINTING

#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << "," << #y << "=" << y << endl
#define nn '\n'
#define pfl(x) printf("%lld\n",x)
#define pcas(i) printf("Case %lld: ",i)
#define Setpre(n) cout<<fixed<<setprecision(n)
```

```

#define itr(it, a) for(auto it = a.begin(); it != a.end(); it++)
#define debug          printf("I am here\n")

///SORTING AND FILLING

#define asort(a)          sort(a,a+n)
#define dsort(a)          sort(a,a+n,greater<int>())
#define vasort(v)         sort(v.begin(), v.end());
#define vdsort(v)         sort(v.begin(), v.end(),greater<ll>());
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define mem(a,b) memset(a,b,sizeof(a))
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))

///CONSTANTS
#define md                10000007
#define PI 3.1415926535897932384626
const double EPS = 1e-9;
const ll N = 2e5+10;
const ll M = 1e9+7;

///INLINE FUNCTIONS
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/((double)log(base));}
inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x)};
int poplow(ll x){return __builtin_ctzll(x)};
int pophigh(ll x){return 63 - __builtin_clzll(x)};

/// Data structures
typedef unsigned long long ull;
typedef pair<ll, ll>      pll;
typedef vector<ll>        vl;
typedef vector<pll>        vpll;
typedef vector<vl>         vvl;
template <typename T> using PQ = priority_queue<T>;

```

```

template <typename T> using QP = priority_queue<T,vector<T>,greater<T>>;

template <typename T> using ordered_set = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
template <typename T,typename R> using ordered_map = tree<T, R , less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
;

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ) { os << it << endl; }
return os; }
    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }
    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }
    string to_string ( const char &c ) { string s; s += c; return "'" + s +
"';' }; }

    template<typename Type> string to_string( vector<Type> );

```



```

    template<typename First, typename Second> string to_string( pair<First,
Second> );
    template<typename Collection> string to_string( Collection );

    template<typename First, typename Second> string to_string( pair<First,
Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
"}"; }
    template<typename Type> string to_string( vector<Type> v ) { bool sep =
false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
to_string( x ); } s += "]"; return s; }
    template<typename Collection> string to_string( Collection collection ) {
bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
sep = true; s += to_string( x ); } s += "}"; return s; }

    void print() { cerr << endl; sep = false; }
    template <typename First, typename... Other> void print( First first,
Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
); print( other... ); }

} using namespace io;

/*=====//

SEFFO4ET

//=====*/

void setIO(){
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);

    freopen("output.txt", "w", stdout);
    #endif // ONLINE_JUDGE
}

struct custom_hash {
    static uint64_t splitmix64(uint64_t x) {
        x += 0x9e3779b97f4a7c15;
        x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
};

```

```

    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};
ll BITree[4*N];

//do this for range: getSum(r) - getSum(l - 1)
ll getMin(ll index){
    ll ret = INT_MAX; // Initialize result
    // Traverse ancestors of BITree[index]
    while (index>0){
        ret = min(ret,BITree[index]); // Add current element of BITree to sum
        index -= index & (-index); // Move index to parent node in getSum View
    }
    return ret;
}

void updateBIT(ll n, ll index, ll val){
    // Traverse all ancestors and add 'val'
    while (index <= n){
        // Add 'val' to current node of BI Tree
        BITree[index] = min(val,BITree[index]);
        // Update index to that of parent in update View
        index += index & (-index);
    }
}

bool cmp(pair<ll,pair<ll,ll>>a,pair<ll,pair<ll,ll>>b){
    return
(a.first==b.first?(a.second.first==b.second.first?a.second.second<b.second.second
:a.second.first<b.second.first):a.first<b.first);
}

int main()
{
    fast;
    ll t;
    //setIO();
    //ll tno=1;;
    t=1;
    cin>>t;

    while(t--){

```

```

    ll n;
    cin>>n;
    vector<pair<ll,pair<ll,ll>> >vec(n);
    for(ll i=0;i<n;i++){
        cin>>vec[i].first;
        cin>>vec[i].second.first;
        cin>>vec[i].second.second;
    }
    // for(ll i=0;i<n;i++){
    //     cout<<vec[i]<<nn;
    // }
    sort(all(vec),cmp);
    fill(BITree,BITree+n+1,INT_MAX);
    ll ans=0;
    for(ll i=0;i<n;i++){
        ll k=getMin(vec[i].second.first);
        if(k>vec[i].second.second){
            ans++;
        }
        updateBIT(n+1,vec[i].second.first,vec[i].second.second);
    }
    cout<<ans<<nn;
}

return 0;
}

```

## Algorithm -10: Segment Tree

**Problem-1:** (Range minimum query algorithm)

Link: <https://www.spoj.com/problems/RMQSQ/en/>

**Solution:**

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

//VVI
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define pb push_back
#define ll long long
#define ff first
#define ss second
#define SZ(a) (int)a.size()
#define UNIQUE(a) (a).erase(unique(all(a)),(a).end())
#define eb emplace_back
#define mp make_pair

///BIT MANIPULATION

#define Set(x, k) (x |= (1LL << k))
#define Unset(x, k) (x &= ~(1LL << k))
#define Check(x, k) (x & (1LL << k))
#define Toggle(x, k) (x ^ (1LL << k))

//LOOPS

#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define Fo(i,k,n) for(i=k;k<n?i<n:i>n;k<n?i+=1:i-=1)

///PRINTING

#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << "," << #y << "=" << y << endl
#define nn '\n'
#define pfl(x) printf("%lld\n",x)
```

```

#define pcas(i)                printf("Case %lld: ",i)
#define Setpre(n) cout<<fixed<<setprecision(n)
#define itr(it, a) for(auto it = a.begin(); it != a.end(); it++)
#define debug                  printf("I am here\n")

///SORTING AND FILLING

#define asort(a)                sort(a,a+n)
#define dsort(a)                sort(a,a+n,greater<int>())
#define vasort(v)               sort(v.begin(), v.end());
#define vdsort(v)               sort(v.begin(), v.end(),greater<ll>());
#define rev(x) reverse(all(x))
#define sortall(x) sort(all(x))
#define mem(a,b) memset(a,b,sizeof(a))
#define all(x) x.begin(), x.end()
#define rev(x) reverse(all(x))

///CONSTANTS
#define md                      10000007
#define PI 3.1415926535897932384626
const double EPS = 1e-9;
const ll N = 2e5+10;
const ll M = 1e9+7;

///INLINE FUNCTIONS
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/(double)log(base);}
inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x)};
int poplow(ll x){return __builtin_ctzll(x)};
int pophigh(ll x){return 63 - __builtin_clzll(x)};

/// Data structures
typedef unsigned long long ull;
typedef pair<ll, ll>      pll;
typedef vector<ll>        vl;
typedef vector<pll>        vpll;

```

```

typedef vector<vl>      vl;
template <typename T> using PQ = priority_queue<T>;
template <typename T> using QP = priority_queue<T,vector<T>,greater<T>>;

template <typename T> using ordered_set = tree<T, null_type, less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
template <typename T,typename R> using ordered_map = tree<T, R , less<T>,
rb_tree_tag, tree_order_statistics_node_update>;
;

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ) { os << it << endl; }
return os; }
    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }
    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }
    string to_string ( const char &c ) { string s; s += c; return "'" + s +
"'; }

```

```

template<typename Type> string to_string( vector<Type> );
template<typename First, typename Second> string to_string( pair<First,
Second> );
template<typename Collection> string to_string( Collection );

template<typename First, typename Second> string to_string( pair<First,
Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
"}"; }
template<typename Type> string to_string( vector<Type> v ) { bool sep =
false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
to_string( x ); } s += "]"; return s; }
template<typename Collection> string to_string( Collection collection ) {
bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
sep = true; s += to_string( x ); } s += "}"; return s; }

void print() { cerr << endl; sep = false; }
template <typename First, typename... Other> void print( First first,
Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
); print( other... ); }

} using namespace io;

/*=====//

SEF04ET

//=====*/

void setIO(){
#ifdef ONLINE_JUDGE
freopen("input.txt", "r", stdin);

freopen("output.txt", "w", stdout);
#endif // ONLINE_JUDGE
}

ll T=0;

ll tre[3*N];

```

```

ll lazy[3*N];

ll merge(ll x, ll y){
    return min(x,y);
}

void buildSegTree(vector<ll>& arr, ll treeIndex, ll lo, ll hi){

    if (lo == hi) {                // leaf node. store value in node.
        tre[treeIndex] = arr[lo];
        return;
    }

    ll mid = lo + (hi - lo) / 2;    // recurse deeper for children.
    buildSegTree(arr, 2 * treeIndex + 1, lo, mid);
    buildSegTree(arr, 2 * treeIndex + 2, mid + 1, hi);

    // merge build results
    tre[treeIndex] = merge(tre[2 * treeIndex + 1], tre[2 * treeIndex + 2]);
}

// call this method as buildSegTree(arr, 0, 0, n-1);
// Here arr[] is input array and n is its size.

ll querySegTree(ll treeIndex, ll lo, ll hi, ll i, ll j){
    // query for arr[i..j]

    if (lo > j || hi < i)           // segment completely outside range
        return 0;                  // represents a null node

    if (i <= lo && j >= hi)         // segment completely inside range
        return tre[treeIndex];

    ll mid = lo + (hi - lo) / 2;    // partial overlap of current segment and
    queried range. Recurse deeper.

    if (i > mid)
        return querySegTree(2 * treeIndex + 2, mid + 1, hi, i, j);
    else if (j <= mid)
        return querySegTree(2 * treeIndex + 1, lo, mid, i, j);

    ll leftQuery = querySegTree(2 * treeIndex + 1, lo, mid, i, mid);
    ll rightQuery = querySegTree(2 * treeIndex + 2, mid + 1, hi, mid + 1, j);

    // merge query results

```



```

    return merge(leftQuery, rightQuery);
}

// call this method as querySegTree(0, 0, n-1, i, j);
// Here [i,j] is the range/interval you are querying.
// This method relies on "null" nodes being equivalent to storing zero.

void updateValSegTree(ll treeIndex, ll lo, ll hi, ll arrIndex, ll val)
{
    if (lo == hi) { // leaf node. update element.
        tre[treeIndex] = val;
        return;
    }

    ll mid = lo + (hi - lo) / 2; // recurse deeper for appropriate child

    if (arrIndex > mid)
        updateValSegTree(2 * treeIndex + 2, mid + 1, hi, arrIndex, val);
    else if (arrIndex <= mid)
        updateValSegTree(2 * treeIndex + 1, lo, mid, arrIndex, val);

    // merge updates
    tre[treeIndex] = merge(tre[2 * treeIndex + 1], tre[2 * treeIndex + 2]);
}

// call this method as updateValSegTree(0, 0, n-1, i, val);
// Here you want to update the value at index i with value val.

void updateLazySegTree(ll treeIndex, ll lo, ll hi, ll i, ll j, ll val){
    if (lazy[treeIndex] != 0) { // this node is lazy
        tre[treeIndex] += (hi - lo + 1) * lazy[treeIndex]; // normalize current
        node by removing laziness

        if (lo != hi) { // update lazy[] for
        children nodes
            lazy[2 * treeIndex + 1] += lazy[treeIndex];
            lazy[2 * treeIndex + 2] += lazy[treeIndex];
        }

        lazy[treeIndex] = 0; // current node
        processed. No longer lazy
    }

    if (lo > hi || lo > j || hi < i)

```

```

        return; // out of range.
    escape.

    if (i <= lo && hi <= j) { // segment is fully
within update range
        tre[treeIndex] += (hi - lo + 1) * val; // update segment

        if (lo != hi) { // update lazy[] for
children
            lazy[2 * treeIndex + 1] += val;
            lazy[2 * treeIndex + 2] += val;
        }

        return;
    }

    ll mid = lo + (hi - lo) / 2; // recurse deeper
for appropriate child

    updateLazySegTree(2 * treeIndex + 1, lo, mid, i, j, val);
    updateLazySegTree(2 * treeIndex + 2, mid + 1, hi, i, j, val);

    // merge updates
    tre[treeIndex] = tre[2 * treeIndex + 1] + tre[2 * treeIndex + 2];
}
// call this method as updateLazySegTree(0, 0, n-1, i, j, val);
// Here you want to update the range [i, j] with value val.

ll queryLazySegTree(ll treeIndex, ll lo, ll hi, ll i, ll j){
    // query for arr[i..j]

    if (lo > j || hi < i) // segment completely
outside range
        return 0; // represents a null
node

    if (lazy[treeIndex] != 0) { // this node is lazy
        tre[treeIndex] += (hi - lo + 1) * lazy[treeIndex]; // normalize current
node by removing laziness

        if (lo != hi) { // update lazy[] for
children nodes
            lazy[2 * treeIndex + 1] += lazy[treeIndex];
            lazy[2 * treeIndex + 2] += lazy[treeIndex];
        }
    }
}

```

```

        lazy[treeIndex] = 0;                                // current node
processed. No longer lazy
    }

    if (i <= lo && j >= hi)                                // segment completely
inside range
        return tre[treeIndex];

    ll mid = lo + (hi - lo) / 2;                            // partial overlap of
current segment and queried range. Recurse deeper.

    if (i > mid)
        return queryLazySegTree(2 * treeIndex + 2, mid + 1, hi, i, j);
    else if (j <= mid)
        return queryLazySegTree(2 * treeIndex + 1, lo, mid, i, j);

    ll leftQuery = queryLazySegTree(2 * treeIndex + 1, lo, mid, i, mid);
    ll rightQuery = queryLazySegTree(2 * treeIndex + 2, mid + 1, hi, mid + 1, j);

    // merge query results
    return leftQuery + rightQuery;
}
// call this method as queryLazySegTree(0, 0, n-1, i, j);
// Here [i,j] is the range/interval you are querying.
// This method relies on "null" nodes being equivalent to storing zero.

struct custom_hash {
    static uint64_t splitmix64(uint64_t x) {
        x += 0x9e3779b97f4a7c15;
        x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};

int main()
{
    fast;
    ll t;

```

```
//setIO();
//ll tno=1;;
t=1;
//cin>>t;

while(t--){
    ll n;
    cin>>n;
    vector<ll>vec(n);
    cin>>vec;
    buildSegTree(vec,0,0,n-1);
    ll q;
    cin>>q;

    ll l,r;
    while(q--){
        cin>>l>>r;
        ll ans=querySegTree(0,0,n-1,l,r);
        cout<<ans<<endl;
    }

}

return 0;
}
```

## Problem-2: (Range sum queries:Lazy propagation)

Link: <https://cses.fi/problemset/task/1648>

Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

typedef tree <
    pair<int, int>,
    null_type,
    less<pair<int, int>>,
    rb_tree_tag,
    tree_order_statistics_node_update> ordered_set;

#define ll long long
#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define pfl(x) printf("%lld\n",x)
#define pb push_back
#define asort(a) sort(a,a+n)
#define dsort(a) sort(a,a+n,greater<int>())
#define vasort(v) sort(v.begin(), v.end());
#define vdsort(v) sort(v.begin(), v.end(),greater<ll>());
#define pn printf("\n")
#define md 10000007
#define debug printf("I am here\n")

#define l(s) s.size()
#define tcas(i,t) for(ll i=1;i<=t;i++)
#define pcas(i) printf("Case %lld: ",i)
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << ", " << #y << "=" << y << endl
#define Setpre(n) cout<<fixed<<setprecision(n)
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/(double)log(base);}
#define M 10000
```

```

inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x);};
int poplow(ll x){return __builtin_ctzll(x);};
int pophigh(ll x){return 63 - __builtin_clzll(x);};

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ) { os << it << endl; }
return os; }
    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }
    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }
    string to_string ( const char &c ) { string s; s += c; return "'" + s +
"'"; }

```

```

template<typename Type> string to_string( vector<Type> );
template<typename First, typename Second> string to_string( pair<First,
Second> );
template<typename Collection> string to_string( Collection );

template<typename First, typename Second> string to_string( pair<First,
Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
"}"; }

template<typename Type> string to_string( vector<Type> v ) { bool sep =
false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
to_string( x ); } s += "]"; return s; }

template<typename Collection> string to_string( Collection collection ) {
bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
sep = true; s += to_string( x ); } s += "}"; return s; }

void print() { cerr << endl; sep = false; }
template <typename First, typename... Other> void print( First first,
Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
); print( other... ); }

} using namespace io;

/*=====//

SEF04ET

//=====*/

void setIO(){
#ifdef ONLINE_JUDGE
freopen("input.txt", "r", stdin);

freopen("output.txt", "w", stdout);
#endif // ONLINE_JUDGE
}

struct custom_hash {
static uint64_t splitmix64(uint64_t x) {
x += 0x9e3779b97f4a7c15;
x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;

```

```

        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};
const double EPS = 1e-9;
const int N = 2e5+10;
ll T=0;

ll tre[3*N];
ll lazy[3*N];

ll merge(ll x,ll y){
    return x+y;
}

void buildSegTree(vector<ll>& arr, ll treeIndex, ll lo, ll hi){

    if (lo == hi) {                // leaf node. store value in node.
        tre[treeIndex] = arr[lo];
        return;
    }

    ll mid = lo + (hi - lo) / 2;    // recurse deeper for children.
    buildSegTree(arr, 2 * treeIndex + 1, lo, mid);
    buildSegTree(arr, 2 * treeIndex + 2, mid + 1, hi);

    // merge build results
    tre[treeIndex] = merge(tre[2 * treeIndex + 1], tre[2 * treeIndex + 2]);
}

// call this method as buildSegTree(arr, 0, 0, n-1);
// Here arr[] is input array and n is its size.

ll querySegTree(ll treeIndex, ll lo, ll hi, ll i, ll j){
    // query for arr[i..j]

    if (lo > j || hi < i)           // segment completely outside range
        return 0;                  // represents a null node

    if (i <= lo && j >= hi)         // segment completely inside range

```



```

        return tre[treeIndex];

    ll mid = lo + (hi - lo) / 2;    // partial overlap of current segment and
queried range. Recurse deeper.

    if (i > mid)
        return querySegTree(2 * treeIndex + 2, mid + 1, hi, i, j);
    else if (j <= mid)
        return querySegTree(2 * treeIndex + 1, lo, mid, i, j);

    ll leftQuery = querySegTree(2 * treeIndex + 1, lo, mid, i, mid);
    ll rightQuery = querySegTree(2 * treeIndex + 2, mid + 1, hi, mid + 1, j);

    // merge query results
    return merge(leftQuery, rightQuery);
}

// call this method as querySegTree(0, 0, n-1, i, j);
// Here [i,j] is the range/interval you are querying.
// This method relies on "null" nodes being equivalent to storing zero.

void updateValSegTree(ll treeIndex, ll lo, ll hi, ll arrIndex, ll val)
{
    if (lo == hi) {                // leaf node. update element.
        tre[treeIndex] = val;
        return;
    }

    ll mid = lo + (hi - lo) / 2;    // recurse deeper for appropriate child

    if (arrIndex > mid)
        updateValSegTree(2 * treeIndex + 2, mid + 1, hi, arrIndex, val);
    else if (arrIndex <= mid)
        updateValSegTree(2 * treeIndex + 1, lo, mid, arrIndex, val);

    // merge updates
    tre[treeIndex] = merge(tre[2 * treeIndex + 1], tre[2 * treeIndex + 2]);
}

// call this method as updateValSegTree(0, 0, n-1, i, val);
// Here you want to update the value at index i with value val.

void updateLazySegTree(ll treeIndex, ll lo, ll hi, ll i, ll j, ll val){
    if (lazy[treeIndex] != 0) {    // this node is lazy

```

```

        tre[treeIndex] += (hi - lo + 1) * lazy[treeIndex]; // normalize current
node by removing laziness

        if (lo != hi) {                                // update lazy[] for
children nodes
            lazy[2 * treeIndex + 1] += lazy[treeIndex];
            lazy[2 * treeIndex + 2] += lazy[treeIndex];
        }

        lazy[treeIndex] = 0;                            // current node
processed. No longer lazy
    }

    if (lo > hi || lo > j || hi < i)                    // out of range.
        return;
    escape.

    if (i <= lo && hi <= j) {                            // segment is fully
within update range
        tre[treeIndex] += (hi - lo + 1) * val;            // update segment

        if (lo != hi) {                                // update lazy[] for
children
            lazy[2 * treeIndex + 1] += val;
            lazy[2 * treeIndex + 2] += val;
        }

        return;
    }

    ll mid = lo + (hi - lo) / 2;                        // recurse deeper
for appropriate child

    updateLazySegTree(2 * treeIndex + 1, lo, mid, i, j, val);
    updateLazySegTree(2 * treeIndex + 2, mid + 1, hi, i, j, val);

    // merge updates
    tre[treeIndex] = tre[2 * treeIndex + 1] + tre[2 * treeIndex + 2];
}
// call this method as updateLazySegTree(0, 0, n-1, i, j, val);
// Here you want to update the range [i, j] with value val.

ll queryLazySegTree(ll treeIndex, ll lo, ll hi, ll i, ll j){
    // query for arr[i..j]

```

```

    if (lo > j || hi < i) // segment completely
outside range
    return 0; // represents a null
node

    if (lazy[treeIndex] != 0) { // this node is lazy
        tre[treeIndex] += (hi - lo + 1) * lazy[treeIndex]; // normalize current
node by removing laziness

        if (lo != hi) { // update lazy[] for
children nodes
            lazy[2 * treeIndex + 1] += lazy[treeIndex];
            lazy[2 * treeIndex + 2] += lazy[treeIndex];
        }

        lazy[treeIndex] = 0; // current node
processed. No longer lazy
    }

    if (i <= lo && j >= hi) // segment completely
inside range
        return tre[treeIndex];

    ll mid = lo + (hi - lo) / 2; // partial overlap of
current segment and queried range. Recurse deeper.

    if (i > mid)
        return queryLazySegTree(2 * treeIndex + 2, mid + 1, hi, i, j);
    else if (j <= mid)
        return queryLazySegTree(2 * treeIndex + 1, lo, mid, i, j);

    ll leftQuery = queryLazySegTree(2 * treeIndex + 1, lo, mid, i, mid);
    ll rightQuery = queryLazySegTree(2 * treeIndex + 2, mid + 1, hi, mid + 1, j);

    // merge query results
    return leftQuery + rightQuery;
}
// call this method as queryLazySegTree(0, 0, n-1, i, j);
// Here [i,j] is the range/interval you are querying.
// This method relies on "null" nodes being equivalent to storing zero.

int main()
{
    fast;

```

```

    ll t;
    //setIO();
    //ll tno=1;;
    t=1;
    //cin>>t;

    while(t--){
        ll n,q;
        cin>>n>>q;
        vector<ll>vec(n);
        cin>>vec;
        //cout<<vec<<endl;
        buildSegTree(vec,0,0,n-1);
        ll l,r;
        ll x,u,k;
        while(q--){
            cin>>x;
            if(x==1){
                cin>>k>>u;
                k--;
                ll temp=querySegTree(0,0,n-1,k,k);
                updateLazySegTree(0,0,n-1,k,k,-temp);
                updateLazySegTree(0,0,n-1,k,k,u);

            }
            else if(x==2){
                cin>>l>>r;

                ll ans= queryLazySegTree(0,0,n-1,l-1,r-1);
                cout<<ans<<endl;
            }
        }
    }

    return 0;
}

```

### Problem-3: (Range update queries)

Link: <https://cses.fi/problemset/task/1651>

Solution:

```
#include<bits/stdc++.h>
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;

typedef tree <
    pair<int, int>,
    null_type,
    less<pair<int, int>>,
    rb_tree_tag,
    tree_order_statistics_node_update> ordered_set;

#define ll long long
#define scl(n) scanf("%lld", &n)
#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for(ll i=1;i<=n;i++)
#define pfl(x) printf("%lld\n",x)
#define pb push_back
#define asort(a) sort(a,a+n)
#define dsort(a) sort(a,a+n,greater<int>())
#define vasort(v) sort(v.begin(), v.end());
#define vdsort(v) sort(v.begin(), v.end(),greater<ll>());
#define pn printf("\n")
#define md 10000007
#define debug printf("I am here\n")

#define l(s) s.size()
#define tcas(i,t) for(ll i=1;i<=t;i++)
#define pcas(i) printf("Case %lld: ",i)
#define fast ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define deb(x) cout << #x << "=" << x << endl
#define deb2(x, y) cout << #x << "=" << x << ", " << #y << "=" << y << endl
#define Setpre(n) cout<<fixed<<setprecision(n)
inline ll GCD(ll a, ll b) { return b == 0 ? a : GCD(b, a % b); }
inline ll LCM(ll a, ll b) { return a * b / GCD(a, b); }
inline ll Ceil(ll p, ll q) {return p < 0 ? p / q : p / q + !(p % q);}
inline ll Floor(ll p, ll q) {return p > 0 ? p / q : p / q - !(p % q);}
inline double logb(ll base,ll num){ return (double)log(num)/((double)log(base);}
#define M 10000
```

```

inline bool isPerfectSquare(long double x){ if (x >= 0) { long long sr =
sqrt(x);return (sr * sr == x); }return false; }
double euclidean_distance(ll x1,ll y1,ll x2,ll y2){double a=(x2-x1)*(x2-
x1);double b=(y2-y1)*(y2-y1);double c=(double)sqrt(a+b);return c;}
int popcount(ll x){return __builtin_popcountll(x);};
int poplow(ll x){return __builtin_ctzll(x);};
int pophigh(ll x){return 63 - __builtin_clzll(x);};

namespace io{
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const pair<First, Second> &p ) { return os << p.first << " " << p.second; }
    template<typename First, typename Second> ostream& operator << ( ostream &os,
const map<First, Second> &mp ) { for( auto it : mp ) { os << it << endl; }
return os; }
    template<typename First> ostream& operator << ( ostream &os, const
vector<First> &v ) { bool space = false; for( First x : v ) { if( space ) os << "
"; space = true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const set<First>
&st ) { bool space = false; for( First x : st ) { if( space ) os << " "; space =
true; os << x; } return os; }
    template<typename First> ostream& operator << ( ostream &os, const
multiset<First> &st ) { bool space = false; for( First x : st ) { if( space ) os
<< " "; space = true; os << x; } return os; }

    template<typename First, typename Second> istream& operator >> ( istream &is,
pair<First, Second> &p ) { return is >> p.first >> p.second; }
    template<typename First> istream& operator >> ( istream &is, vector<First> &v
) { for( First &x : v ) { is >> x; } return is; }

    long long fastread(){ char c; long long d = 1, x = 0; do c = getchar();
while( c == ' ' || c == '\n' ); if( c == '-' ) c = getchar(), d = -1; while(
isdigit( c ) ){ x = x * 10 + c - '0'; c = getchar(); } return d * x; }

    static bool sep = false;

    using std::to_string;

    string to_string( bool x ){ return ( x ? "true" : "false" ); }
    string to_string( const string &s ){ return "\"" + s + "\""; }
    string to_string( const char * s ){ return "\"" + string( s ) + "\""; }
    string to_string ( const char &c ) { string s; s += c; return "'" + s +
"'" ; }

```

```

template<typename Type> string to_string( vector<Type> );
template<typename First, typename Second> string to_string( pair<First,
Second> );
template<typename Collection> string to_string( Collection );

template<typename First, typename Second> string to_string( pair<First,
Second> p ){ return "{" + to_string( p.first ) + ", " + to_string( p.second ) +
"}"; }

template<typename Type> string to_string( vector<Type> v ) { bool sep =
false; string s = "["; for( Type x: v ){ if( sep ) s += ", "; sep = true; s +=
to_string( x ); } s += "]"; return s; }

template<typename Collection> string to_string( Collection collection ) {
bool sep = false; string s = "{"; for( auto x: collection ){ if( sep ) s += ", ";
sep = true; s += to_string( x ); } s += "}"; return s; }

void print() { cerr << endl; sep = false; }
template <typename First, typename... Other> void print( First first,
Other... other ) { if( sep ) cerr << " | "; sep = true; cerr << to_string( first
); print( other... ); }

} using namespace io;

/*=====//

SEF04ET

//=====*/

void setIO(){
#ifdef ONLINE_JUDGE
freopen("input.txt", "r", stdin);

freopen("output.txt", "w", stdout);
#endif // ONLINE_JUDGE
}

struct custom_hash {
static uint64_t splitmix64(uint64_t x) {
x += 0x9e3779b97f4a7c15;
x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;

```

```

        x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
        return x ^ (x >> 31);
    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM =
chrono::steady_clock::now().time_since_epoch().count();
        return splitmix64(x + FIXED_RANDOM);
    }
};
const double EPS = 1e-9;
const int N = 2e5+10;
ll T=0;

ll tre[3*N];
ll lazy[3*N];

ll merge(ll x,ll y){
    return x+y;
}

void buildSegTree(vector<ll>& arr, ll treeIndex, ll lo, ll hi){

    if (lo == hi) {                // leaf node. store value in node.
        tre[treeIndex] = arr[lo];
        return;
    }

    ll mid = lo + (hi - lo) / 2;    // recurse deeper for children.
    buildSegTree(arr, 2 * treeIndex + 1, lo, mid);
    buildSegTree(arr, 2 * treeIndex + 2, mid + 1, hi);

    // merge build results
    tre[treeIndex] = merge(tre[2 * treeIndex + 1], tre[2 * treeIndex + 2]);
}

// call this method as buildSegTree(arr, 0, 0, n-1);
// Here arr[] is input array and n is its size.

ll querySegTree(ll treeIndex, ll lo, ll hi, ll i, ll j){
    // query for arr[i..j]

    if (lo > j || hi < i)           // segment completely outside range
        return 0;                  // represents a null node

    if (i <= lo && j >= hi)         // segment completely inside range

```



```

        return tre[treeIndex];

    ll mid = lo + (hi - lo) / 2;    // partial overlap of current segment and
queried range. Recurse deeper.

    if (i > mid)
        return querySegTree(2 * treeIndex + 2, mid + 1, hi, i, j);
    else if (j <= mid)
        return querySegTree(2 * treeIndex + 1, lo, mid, i, j);

    ll leftQuery = querySegTree(2 * treeIndex + 1, lo, mid, i, mid);
    ll rightQuery = querySegTree(2 * treeIndex + 2, mid + 1, hi, mid + 1, j);

    // merge query results
    return merge(leftQuery, rightQuery);
}

// call this method as querySegTree(0, 0, n-1, i, j);
// Here [i,j] is the range/interval you are querying.
// This method relies on "null" nodes being equivalent to storing zero.

void updateValSegTree(ll treeIndex, ll lo, ll hi, ll arrIndex, ll val)
{
    if (lo == hi) {                // leaf node. update element.
        tre[treeIndex] = val;
        return;
    }

    ll mid = lo + (hi - lo) / 2;    // recurse deeper for appropriate child

    if (arrIndex > mid)
        updateValSegTree(2 * treeIndex + 2, mid + 1, hi, arrIndex, val);
    else if (arrIndex <= mid)
        updateValSegTree(2 * treeIndex + 1, lo, mid, arrIndex, val);

    // merge updates
    tre[treeIndex] = merge(tre[2 * treeIndex + 1], tre[2 * treeIndex + 2]);
}

// call this method as updateValSegTree(0, 0, n-1, i, val);
// Here you want to update the value at index i with value val.

void updateLazySegTree(ll treeIndex, ll lo, ll hi, ll i, ll j, ll val){
    if (lazy[treeIndex] != 0) {    // this node is lazy

```

```

        tre[treeIndex] += (hi - lo + 1) * lazy[treeIndex]; // normalize current
node by removing laziness

        if (lo != hi) { // update lazy[] for
children nodes
            lazy[2 * treeIndex + 1] += lazy[treeIndex];
            lazy[2 * treeIndex + 2] += lazy[treeIndex];
        }

        lazy[treeIndex] = 0; // current node
processed. No longer lazy
    }

    if (lo > hi || lo > j || hi < i) // out of range.
        return;
escape.

    if (i <= lo && hi <= j) { // segment is fully
within update range
        tre[treeIndex] += (hi - lo + 1) * val; // update segment

        if (lo != hi) { // update lazy[] for
children
            lazy[2 * treeIndex + 1] += val;
            lazy[2 * treeIndex + 2] += val;
        }

        return;
    }

    ll mid = lo + (hi - lo) / 2; // recurse deeper
for appropriate child

    updateLazySegTree(2 * treeIndex + 1, lo, mid, i, j, val);
    updateLazySegTree(2 * treeIndex + 2, mid + 1, hi, i, j, val);

    // merge updates
    tre[treeIndex] = tre[2 * treeIndex + 1] + tre[2 * treeIndex + 2];
}
// call this method as updateLazySegTree(0, 0, n-1, i, j, val);
// Here you want to update the range [i, j] with value val.

ll queryLazySegTree(ll treeIndex, ll lo, ll hi, ll i, ll j){
    // query for arr[i..j]

```

```

    if (lo > j || hi < i)                                // segment completely
outside range
    return 0;                                            // represents a null
node

    if (lazy[treeIndex] != 0) {                          // this node is lazy
        tre[treeIndex] += (hi - lo + 1) * lazy[treeIndex]; // normalize current
node by removing laziness

        if (lo != hi) {                                // update lazy[] for
children nodes
            lazy[2 * treeIndex + 1] += lazy[treeIndex];
            lazy[2 * treeIndex + 2] += lazy[treeIndex];
        }

        lazy[treeIndex] = 0;                            // current node
processed. No longer lazy
    }

    if (i <= lo && j >= hi)                              // segment completely
inside range
        return tre[treeIndex];

    ll mid = lo + (hi - lo) / 2;                        // partial overlap of
current segment and queried range. Recurse deeper.

    if (i > mid)
        return queryLazySegTree(2 * treeIndex + 2, mid + 1, hi, i, j);
    else if (j <= mid)
        return queryLazySegTree(2 * treeIndex + 1, lo, mid, i, j);

    ll leftQuery = queryLazySegTree(2 * treeIndex + 1, lo, mid, i, mid);
    ll rightQuery = queryLazySegTree(2 * treeIndex + 2, mid + 1, hi, mid + 1, j);

    // merge query results
    return leftQuery + rightQuery;
}
// call this method as queryLazySegTree(0, 0, n-1, i, j);
// Here [i,j] is the range/interval you are querying.
// This method relies on "null" nodes being equivalent to storing zero.

int main()
{
    fast;

```

```

    ll t;
    //setIO();
    //ll tno=1;;
    t=1;
    //cin>>t;

    while(t--){
        ll n,q;
        cin>>n>>q;
        vector<ll>vec(n);
        cin>>vec;
        buildSegTree(vec,0,0,n-1);
        ll x,k,u;
        while(q--){
            cin>>x;
            if(x==1){
                ll l,r;
                cin>>l>>r>>u;
                updateLazySegTree(0,0,n-1,l-1,r-1,u);
            }
            else{
                cin>>k;
                ll ans=queryLazySegTree(0,0,n-1,k-1,k-1);
                cout<<ans<<endl;
            }
        }
    }

    return 0;
}

```

