

Worksheet 7

Student Name:

Ravindra Singh Mehta

UID:20BCS7060

Branch: CSE

Section/Group:DWWC-43

Semester:5

Date of

Performance:11-01-
2023

Subject Name:Data
Structure

Subject Code:

1. Aim/Overview of the practical:

ii) Sometimes mysteries happen. Chef found a directed graph with N vertices and M edges in his kitchen!

The evening was boring and chef has nothing else to do, so to entertain himself, Chef thought about a question "What is the minimum number of edges he needs to reverse in order to have at least one path from vertex 1 to vertex N , where the vertices are numbered from 1 to N .

iii) In a market of mangoes, there are NN sellers numbered from 11 to NN . The ii -th seller initially charges a price of A_i coins for each of their mangoes. However, the sellers are very competitive, and they will change prices based on other sellers nearby.

You have a simple graph (unweighted, undirected graph containing no self-loops or multiple edges) GG with NN vertices and MM edges. There is an edge between two sellers if and only if they are neighbours. When you buy a mango from a seller numbered XX , the following occurs:

- Seller XX keeps his price the same.
- Every seller YY who is a neighbour of XX increases their price by 11, that is, $A_Y = A_Y + 1$ for every YY who is a neighbour of XX .
- Every other seller ZZ who is not a neighbour of XX decreases their price by 11; that is, $A_Z = A_Z - 1$ for every ZZ who is not a neighbour of XX .

Prices can become zero or negative during this process.

Now you should process QQ queries of the following three types:

- $\texttt{+ } u\ v$ — Add an edge between nodes u and v in the graph. It's guaranteed there is no edge between u and v currently.
- $\texttt{- } u\ v$ — Remove an edge between nodes u and v in the graph. It's guaranteed there is an edge between u and v currently.
- $\texttt{? } ?$ — Find the **minimum** amount of money you need to pay so that you can buy **exactly** one mango from **every** seller.

After queries of this type, **the prices of all sellers are reset to their initial value.**

- iv) A late night party is going on in hotel Taj. There are N persons numbered from **1, 2, 3 N** (assume one based indexing) and a celebrity is present in the party. Some goons are planning to assassinate the celebrity, meanwhile cops come to know about the assassination, so they arrive immediately at the party to protect the celebrity.

A person can be termed as a Celebrity if he/she fulfills following conditions :

- If a person is known to each and every person present in the party, but he doesn't know anyone.

You have to find out whether the celebrity is alive or not.

- v) Chef's college is starting next week. There are SS subjects in total, and he needs to choose KK of them to attend each day, to fulfill the required number of credits to pass the semester. There are $N + 1$ buildings. His hostel is in building number 00 . Subject i is taught in building A_i . After each subject, there is a break, during which he goes back to his hostel. There are MM bidirectional paths of length 1 which connects building u to building v . Find the minimum possible total distance Chef needs to travel each day if he chooses his subjects wisely.

- vi) **This is an interactive problem.**

There is a hidden undirected graph with n vertices. There are no self-loops or multiple edges. Each vertex is colored black or white, and the colors are also hidden from you.

To make a query, you choose a vertex v . The interactor will respond with the current color of v , and then it will flip the colors of all vertices adjacent to v . A vertex is not considered adjacent to itself, so the color of v doesn't change.

After at most 6000 queries, find the adjacency matrix of the graph.

Note: The interactor is not adaptive. In other words, the graph and colors are fixed in the beginning.

Interaction

Begin the interaction by reading a single integer n - the number of vertices in the graph.

To ask a query, output $?\ v$ for a vertex v ($1 \leq v \leq n$). Then read a single character describing the current color of v , where "B" denotes black and "W" denotes white. After making this query, the colors of all vertices adjacent to v will flip.

To print the answer, output $!$ then on the next n lines print n strings of length n , consisting of symbols "0" and "1". The j -th character of the i -th string should be "1" if and only if there is an edge between vertices i and j . Since there are no self-loops, the main diagonal should contain only "0".

If at any time you make an invalid query or exceed the query limit, the interaction is terminated and you will receive a Wrong Answer verdict.

Remember to flush the output after printing each line!

vii) Given an array AA of length NN such that $1 \leq A_i \leq N$ and $A_i \neq i, \forall i \in [1, N]$.

Count the number of arrays BB of length NN such that $\forall i \in [1, N] \forall i \in [1, N]$:

- $B_i \neq B_{A_i}$
- $1 \leq B_i \leq M$

Since the answer may be large, print it modulo 10^9+7 .

Input Format

- The first line contains a single integer TT – the number of test cases. The description of TT test cases follows.
- Each test case contains 22 lines of input:
 - The first line of each test case contains two space separated integers NN, MM .
 - The second line of each test case contains NN space separated integers A_1, A_2, \dots, A_N .

ix) Chef Ada constructed a simple undirected graph GG with NN vertices (numbered 11 through NN) as follows:

- Start with a [complete graph](#) with KK vertices and $K \cdot (K-1)/2$ edges.
- Then, add $N-K$ vertices one by one. After adding each vertex, find some [clique](#) of $K-1$ vertices in the graph and connect the new vertex to all the vertices of that clique.

Consider NN boolean variables X_1, X_2, \dots, X_N and their negations $\neg X_1, \neg X_2, \dots, \neg X_N$, corresponding to vertices of the graph. You are given the initial values of all variables.

A *CNF boolean expression* is defined as a [conjunction](#) of clauses, where each clause is a [disjunction](#) of two or more elements of the set of variables and their negations.

Using the graph GG , Ada wrote a CNF boolean expression BB with MM clauses, where each clause corresponds to a clique in the graph with some (maybe none) of the variables (corresponding to vertices) negated. More formally, for each ii from 11 to MM :

- Ada chose some clique in the graph, consisting of L_i vertices Z_1, Z_2, \dots, Z_{L_i}
- then for each valid jj , Ada chose a variable $P_{i,j}$ as either X_{Z_j} or $\neg X_{Z_j}$
- the ii -th clause in the expression BB had L_i terms: $P_{i,1} \vee P_{i,2} \vee \dots \vee P_{i,L_i}$

Ada wants the expression to evaluate to "true". In order to achieve that, she may change the values of variables; for each valid ii , the cost of changing the value of X_i (from "true" to "false" or vice versa) is C_i . Find the minimum total cost needed to make the expression BB evaluate to "true" or determine that it is impossible.

x) Chef likes tournament graphs, which are directed graphs where each unordered pair of vertices is directly connected by exactly one edge. A directed graph is *strongly connected* if for each pair of vertices (a, b) , there is a path from the vertex a to the vertex b .

Consider a tournament graph GG with NN vertices (numbered 11 through NN). Chef takes a sequence of pairs of vertices $(a_1, b_1), (a_2, b_2), \dots, (a_M, b_M)$ and does the following for each ii from 11 to MM in this order:

- Flip the direction of the edge between vertices a_i and b_i .
- If the graph is strongly connected either before or after flipping this edge, declare GG a *bad tournament*.

If GG is never declared a bad tournament, Chef calls it a *good tournament*. Given the sequence of MM edge flips, find a good tournament.

xi) Uneori îmi spun: am să mor

atât de singuratecă-n mijlocul lor;

limba simplă a bucuriilor n-am învățat;

am să mor ca o pasăre care prea mult a zburat,

dar n-a făcut cuib nicăieri.

— Oamenii ma uimesc, Magda Isafost

As of writing the informal statement, the winter just came and the statement was in season. Now, it is not winter anymore, so here is the formal statement instead:

You are given a connected graph with N nodes and M edges. You are given Q queries of the following types:

- $\texttt{1 u}$ 1 u : Given node u ($1 \leq u \leq N$), set the state of the node u to frozen.
- $\texttt{2 t}$ 2 t : Given t , let t units of time pass by.
- $\texttt{3 v}$ 3 v : Given node v ($1 \leq v \leq N$), answer if node v is currently frozen.

Initially, **no** node is frozen.

The graph has the following property:

- If, at time T , a node u is frozen, then, at time $(T+1)$, **all neighbours** of u become frozen.

For each query of type 3, answer whether the node is currently frozen.

Note:

- If a node is already frozen, it remains unaffected after a type 1 query.
- If, for a frozen node u , a neighbour v of node u is frozen at time T , node v remains unaffected at time $(T+1)$.

xii) Leha is playing a very interesting game. The game will be played on a rectangular grid consisting of N rows and M columns. Initially all the cells of the grid are uncolored.

Leha's initial score is zero. At each turn, he chooses some cell that is yet not colored, and colors that cell. The score obtained in this step will be number of neighboring colored cells of the cell that Leha colored in this step. Two cells are neighbors of each other if they share a side between them. The game will end when all the cells are colored. Finally, total score obtained at the end of the game will sum of score obtained in each turn.

Leha wants to know what maximum score he can get? Can you please help him in finding this out?

ii) Steps for experiment/practical/Code:

ii)

```
#include <bits/stdc++.h>
using namespace std;
const int N = 1e5+10;
const int infi=1e9+10;
vector<pair<int,int>>g[N];
vector<int>level(N,infi);
int n,m;
void bfs(){
    level[1]=0;
    deque<int> dq;
    dq.push_back(1);
    while(!dq.empty()){
        int cur_v= dq.front();
        dq.pop_front();
        for(auto childs:g[cur_v]){
            int child = childs.first;
            int wt = childs.second;
            if(level[cur_v]+wt < level[child]){
                level[child] = level[cur_v] + wt;
                if(wt==1) dq.push_back(child);
                else dq.push_front(child);
            }
        }
    }
    if(level[n]==infi) cout<<-1 ;
    else cout<<level[n];
}
int main() {

    cin>>n>>m;
    for(int i=0;i<m;i++){
        int x,y;
        cin>>x>>y; return 0;}
```

iii)

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

```
using namespace std;
```

```
int main() {  
    ios::sync_with_stdio(false);  
    cin.tie(nullptr);  
    int n, m;  
    cin >> n >> m;  
    long long sum = 0;  
    for (int i = 1; i <= n; i++) {  
        long long x;  
        cin >> x;  
        sum += x;  
    }  
    long long edges = (long long)m, unused = ((long long)n * (n - 1)) / 2LL - edges;  
    for (int i = 0; i < m; i++) {  
        int u, v;  
        cin >> u >> v;  
    }  
    int b=edges-unused;  
    int q;  
    cin >> q;  
    for (int i = 0; i < q; i++) {  
        char x;  
        cin >> x;  
        if (x == '?') {  
            cout << sum + edges-unused << "\n";  
            continue;  
        }  
        int u, v;  
        cin >> u >> v;  
        if (x == '+') {  
            edges++;unused--;  
        }  
    }  
}
```

```

    }
    else if (x == '-') {
        edges--;
        unused++;
    }

}
return 0;
}

```

iv)

```

#include <iostream>

using namespace std;

int main() {

    int t, n, m, r1, r2;
    scanf("%d", &t);
    for(int i=0; i<t; i++){
        scanf("%d %d", &n, &m);
        int arr[n+1]={0};
        for(int j=0; j<m; j++){
            scanf("%d %d", &r1, &r2);
            arr[r2]++;
            arr[r1]=-1;
        }
        int flag=1;
        for(int j=1; j<=n; j++){
            if(arr[j]==n-1){
                printf("alive ");
                printf("%d\n", j);
                flag=0;
                break;
            }
        }
        if(flag==1){ printf("dead\n");}
    }

    return 0;}

```

v)

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

#define llint long long int
using namespace std;

void run()
{
    // Insert code here
    int n, m, s, k;
    cin >> n >> m >> s >> k;

    vector<vector<int>>> graph(n+1);

    for(int i = 0; i < m; ++i){
        int u, v;
        cin >> u >> v;
        graph[u].push_back(v);
        graph[v].push_back(u);
    }

    std::vector<int> count(n+1);
    for (int i = 0; i < s; ++i){
        int val;
        cin >> val;
        count[val]++;
    }
    vector<bool> vis(n+1);
    queue<int> q;

    q.push(0);
    vis[0] = true;

    llint res = 0, curr = 0;

    while(!q.empty() && k > 0){
```



```

int size = q.size();
for(int i = 0; i < size; ++i){
    int node = q.front();
    q.pop();
    for(auto adj : graph[node]){
        if(!vis[adj]){
            vis[adj] = true;
            q.push(adj);
        }
    }
    int val = min(k, count[node]);
    res += 2*curr*val;
    k -= val;
}
curr++;
}
cout << res << "\n";
}

int main()
{
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(NULL);

    int t = 1;
    std::cin >> t;
    while (t--)
        run();

    return 0;
}

```

vi)

```
#include <bits/stdc++.h>
using namespace std;
bool query(int v) {
    cout << "? " << v << endl;
    char c;
    cin >> c;
    return c == 'B';
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    int n;
    cin >> n;
    vector<vector<bool>> q(n + 1);
    vector<int> ve;
    ve.push_back(1);
    for(int i = 1; i <= n; i++) {
        ve.push_back(i);
    }
    for(int k : ve) {
        for(int i = k; i <= n; i++) {
            q[i].push_back(query(i));
        }
    }
    vector<vector<bool>> adj(n + 1, vector<bool>(n + 1));
    for(int i = 2; i <= n; i++) {
        for(int j = 1; j < i; j++) {
            adj[i][j] = adj[j][i] = (q[i][j - 1] ^ q[i][j + 1]);
        }
    }
    cout << "!\\n";
    for(int i = 1; i <= n; i++) {
        for(int j = 1; j <= n; j++) {
            cout << adj[i][j];
        }
        cout << '\\n';
    }
    cout << flush;
}
```

vii)

```
# include<bits/stdc++.h>
# define int long long
using namespace std;

const int mod =1e9+7;

int power(int a ,int b){
    int res =1;

    while(b){
        if(b&1){
            res =( res*a)%mod ;
        }

        b/=2;
        a= (a*a)%mod ;
    }

    return res;
}

void preprocess (vector<vector<int>>& dp ,int m){
    int n =size(dp) ;
    dp[1][0] = 0 , dp[1][1] = m ;

    for(int len =2 ; len<n; len++){
        dp[len][0] =( dp[len-1][0] * (m-2) + dp[len-1][1]*(m-1) ) %mod ;
        dp[len][1]= dp[len-1][0] * 1 ;
    }
}

int solve (vector<int>a , int n ,int m ){
    vector<bool> vis(n+1,false) ;
    vector<int>indegree(n+1,0) ;

    queue<int>processNext ;

    for(int i=1;i<=n;i++){
        indegree[a[i]]++ ;        // i->Ai
    }

    for(int i=1;i<=n;i++){
        if(indegree[i]==0) processNext.push(i) ;
    }
}
```

```

}

int count_nodes_in_chain = 0;

while( !processNext.empty() ){
    auto rem = processNext.front() ;
    processNext.pop() ;

    count_nodes_in_chain++ ;
    vis[rem]=true;

    int next = a[rem] ;
    if(--indegree[next] == 0) processNext.push(next) ;
}

int ans = 1;

ans= (ans * power(m-1,count_nodes_in_chain ) )%mod ;

vector<vector<int>> dp (n+1 , vector<int>(2)) ;

preprocess(dp,m) ;

for(int i=1;i<=n;i++){
    if( !vis[i] ){
        int len =0 , cur =i ;
        while(!vis[cur] ){
            len++;
            vis[cur]=true;
            cur = a[cur ] ;
        }

        ans = (ans * dp[len][0] ) % mod ;
    }
}

return ans ;
}

int32_t main() {
    int t;
    cin>>t;

    while(t--){
        int n,m;

```

```

        cin>>n>>m ;
        vector<int>arr(n+1) ;

        for(int i=1;i<=n;i++){
            cin>>arr[i] ;
        }

    cout<<solve(arr,n,m) <<endl;
}
}

```

ix)

```

#include<bits/stdc++.h>
namespace my_std{
    using namespace std;
    #define pii pair<int,int>
    #define fir first
    #define sec second
    #define MP make_pair
    #define rep(i,x,y) for (int i=(x);i<=(y);i++)
    #define drep(i,x,y) for (int i=(x);i>=(y);i--)
    #define go(x) for (int i=head[x];i=edge[i].nxt)
    #define templ template<typename T>
    #define sz 233
    typedef long long ll;
    typedef double db;
    mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
    templ inline T rnd(T l,T r) {return uniform_int_distribution<T>(l,r)(rng);}
    templ inline bool chkmax(T &x,T y){return x<y?x=y,1:0;}
    templ inline bool chkmin(T &x,T y){return x>y?x=y,1:0;}
    templ inline void read(T& t)
    {
        t=0;char f=0,ch=getchar();double d=0.1;
        while(ch>'9'||ch<'0') f=(ch=='-'),ch=getchar();
        while(ch<='9'&&ch>='0') t=t*10+ch-48,ch=getchar();
        if(ch=='.') {ch=getchar();while(ch<='9'&&ch>='0') t+=d*(ch^48),d*=0.1,ch=getchar();}
        t=(f?-t:t);
    }
    template<typename T,typename... Args>inline void read(T& t,Args&... args){read(t);
    read(args...);}
    char __sr[1<<21],__z[20];int __C=-1,__zz=0;
    inline void Ot(){fwrite(__sr,1,__C+1,stdout),__C=-1;}
    inline void print(int x)

```

```

{
    if(__C>1<<20)Ot();if(x<0)__sr[++__C]='-',x=-x;
    while(__z[++__zz]=x%10+48,x/=10);
    while(__sr[++__C]=__z[__zz],--__zz);__sr[++__C]='\n';
}
void file()
{
    #ifdef NTFOrz
    freopen("a.in","r",stdin);
    #endif
}
inline void chktime()
{
    #ifdef NTFOrz
    cerr<<clock()/1000.0<<"\n";
    #endif
}
#ifdef mod
ll ksm(ll x,int y){ll ret=1;for (;y>=>1,x=x*x%mod) if (y&1) ret=ret*x%mod;return ret;}
ll inv(ll x){return ksm(x,mod-2);}
#else
ll ksm(ll x,int y){ll ret=1;for (;y>=>1,x=x*x) if (y&1) ret=ret*x;return ret;}
#endif
// inline ll mul(ll a,ll b){ll d=(ll)(a*(double)b/mod+0.5);ll ret=a*b-d*mod;if (ret<0)
ret+=mod;return ret;}
}
using namespace my_std;

int n,m,K;
int fa[sz];
vector<int>V[sz],VV[sz];
char s[sz]; ll w[sz];
vector<int>Z[sz]; int e[sz][sz];
vector<pii>q[sz];
ll dp[sz][1<<10];

void dfs(int x)
{
    for (auto v:VV[x]) dfs(v);
    int k=V[x].size(),S=1<<k;
    rep(ss,0,S-1)
    {
        int flg=1; for (auto p:q[x]) flg&=!((ss&p.fir)^p.sec);
        if (!flg) continue;
        ll W=((ss>>(k-1)&1)!=s[x]-'0')*w[x];
        for (auto v:VV[x])

```

```

    {
        int t=0;
        rep(i,0,k-1) rep(j,0,(int)V[v].size()-2) if (V[x][i]==V[v][j]) t|=(ss>>i&1)<<j;
        W+=dp[v][t]; chkmin(W,ll(1e18));
    }
    chkmin(dp[x][ss&((S>>1)-1)],W);
}

}

void CLR()
{
    rep(i,1,n) V[i].clear(),VV[i].clear(),s[i]='\0',w[i]=0,fa[i]=0,q[i].clear();
    rep(i,1,m) Z[i].clear();
    rep(i,1,n) rep(j,1,n) e[i][j]=0;
}

void work()
{
    read(n,m,K);
    cin>>(s+1);
    rep(i,1,n) read(w[i]);
    rep(k,1,m)
    {
        int L,x; read(L);
        while (L--) read(x),Z[k].push_back(x);
        sort(Z[k].begin(),Z[k].end(),[](int x,int y){return abs(x)<abs(y);});
        for (auto _x:Z[k]) for (auto _y:Z[k]) { int x=abs(_x),y=abs(_y); if (x>y) e[x][y]=1; }
    }
    rep(i,1,K) rep(j,1,i-1) e[i][j]=1;
    drep(i,n,1) rep(j,1,i-1) rep(k,1,j-1) if (e[i][j]&&e[i][k]) e[j][k]=1;
    rep(i,1,n) rep(j,1,i-1) if (e[i][j]) V[i].push_back(j); rep(i,2,n) if (!V[i].size()) V[i].push_back(i-1); rep(i,2,n) fa[i]=V[i].back();
    rep(i,2,n) VV[fa[i]].push_back(i);
    rep(i,1,n) V[i].push_back(i);
    rep(k,1,m)
    {
        int x=abs(Z[k].back()),t1=0,t2=0;
        for (auto y:Z[k]) rep(i,0,(int)V[x].size()-1) if (abs(y)==V[x][i]) t1|=1<<i,t2|=(y<0)<<i;
        q[x].push_back(MP(t1,t2));
    }
    memset(dp,0x3f,sizeof(dp));
    dfs(1);
    printf("%lld\n",(dp[1][0]>1e10?-1ll:dp[1][0]));
    CLR();
}

```

```
int main()
{
    file();
    int T; read(T);
    while (T--) work();
    return 0;
}
```

x)

```
#include <bits/stdc++.h>
using namespace std;
```

```
#define N 1010
```

```
int n, m, a[N], b[N];
bool col[N], s[N][N];
```

```
bool check(int u) {
    for (int i = 1; i <= n; i++) col[i] = 0; col[u] = 1;
    for (int i = 1; i <= n; i++) if (i != u) s[i][u] = 1, s[u][i] = 0;
    for (int i = 0; i < m; i++) {
        int x = a[i], y = b[i];
        if ((col[x] ^ col[y]) == 1) {
            if (col[x]) swap(x, y);
            s[x][y] ^= 1, s[y][x] ^= 1;
            col[x] = 1;
            for (int j = 1; j <= n; j++) if (!col[j]) s[j][x] = 1, s[x][j] = 0;
        }
        else if ((col[x] & col[y]) == 1) s[x][y] ^= 1, s[y][x] ^= 1;
    }
    bool fg = false;
    for (int i = 1; i <= n; i++) if (!col[i]) fg = true;
    if (!fg) return 0;
    for (int i = 0; i < m; i++) {
        int x = a[i], y = b[i];
        s[x][y] ^= 1, s[y][x] ^= 1;
    }
    for (int i = 1; i <= n; puts(""), i++) for (int j = i + 1; j <= n; j++) printf("%d ", s[i][j]);
    return 1;
}
```

```
int main() {
    int T;
    scanf("%d", &T);
    while (T --) {
```



```

scanf("%d %d", &n, &m);
for (int i = 0; i < m; i++) scanf("%d %d", &a[i], &b[i]);
for (int i = 1; i <= n; i++) if (check(i)) break;
}

return 0;
}

```

xi)

```

#include<bits/stdc++.h>
using namespace std;

int32_t main()
{
    int n,m,q1;
    cin>>n>>m>>q1;
    vector<vector<int>>>vec(n+1);
    for(int i=0;i<m;i++)
    {
        int x,y;
        cin>>x>>y;
        vec[x].push_back(y);
        vec[y].push_back(x);
    }

    vector<bool>visited(n+1,false);
    queue<int>q;
    vector<bool>frozen(n+1,false);

    while(q1--)
    {
        int query,type;
        cin>>type>>query;
        if(type==1)
        {
            if(frozen[query])continue;
            frozen[query]=true;
            // if(visited[query]==false)
            // {
            //     visited[query]=true;
            //     q.push(query);
            // }
        }
        else if(type==2)

```

```
{
```



**DEPARTMENT OF
ACADEMIC AFFAIRS**

Discover. Learn. Empower.

**NAAC
GRADE A+**
ACCREDITED UNIVERSITY

```
visited[tp]=true,
q.pop();

for(auto nbr:vec[tp])
{
    if(!visited[nbr]){
        if(frozen[nbr]) continue;
        frozen[nbr]=true;
        q.push(nbr);
    }
}
}query--;
}
}else
{
    if(frozen[query])
    {
        cout<<"Yes"<<endl;
    }else
    {
        cout<<"No"<<endl;
    }
}
}
}
return 0;
}
```

xii)

```
#include <iostream>
using namespace std;

int main() {
    // your code goes here
    int t,i;
    cin>>t;
    for(i=0;i<t;++i)
    {
        int n,m;
        cin>>n>>m;

        cout<<(n-1)+(m-1)+(2)*(n-1)*(m-1)<<endl;
    }
    return 0;
}
```

3. Observations/Discussions/ Complexity Analysis:

ii)

The screenshot shows the CodeChef website interface for the 'Chef and Reversing' problem. The page is divided into several sections: 'Problem', 'Input', 'Output', and 'Constraints'. The 'Problem' section contains the problem statement and a 'Go Fullscreen' button. The 'Input' section describes the input format. The 'Output' section describes the output format. The 'Constraints' section lists the constraints. The 'Test against Custom Input' section shows a sample input and output. The 'Status' section shows 'Successfully executed' with a time of 0.008808 secs and memory of 5.796 Mb. The 'Compile & Run' button is visible.

Problem

Read problems statements in [Mandarin Chinese](#) and [Russian](#).

Sometimes mysteries happen. Chef found a directed graph with N vertices and M edges in his kitchen!

The evening was boring and chef has nothing else to do, so to entertain himself, Chef thought about a question "What is the minimum number of edges he needs to reverse in order to have at least one path from vertex 1 to vertex N , where the vertices are numbered from 1 to N .

Input

Each test file contains only one test case.

The first line of the input contains two space separated integers N and M , denoting the number of vertices and the number of edges in the graph respectively. The i^{th} line of the next M lines contains two space separated integers X_i and Y_i , denoting that the i^{th} edge connects vertices from X_i to Y_i .

Output

In a single line, print the minimum number of edges we need to revert. If there is no way of having at least one path from 1 to N , print -1.

Constraints

Test against Custom Input

7 7
1 2
3 2
3 4

Pro WA Failed Test Cases is available for this problem

Status : Successfully executed

Time: 0.008808 secs Memory: 5.796 Mb

Input

7 7
1 2
3 2

Activate Windows
Go to Settings to activate Windows.

Upload code as file

Compile & Run Submit Code

iii)

WhatsApp

Learn | CodeChef

Mango Market - Problems | CodeChef

Mango Market Submissions | CodeChef

Solution: 85052795 | CodeChef

codechef.com/problems/MANGOMKT?tab=statement

GmailYouTubeMaps

Mango Market

Difficulty Rating: 2393

Expand

Prev Problem

Next Problem

Statement

Submissions

Solution

C++14

Test against Custom Input

4 3
8 9 6 10
2 1
1 3

Problem Solver Badge 113 / 250

Solve 137 more problems to get Silver Badge

Next Problem

Status: Correct Answer

Submission ID: 85053501

Time: 0.04s

Congratulations on solving the problem. Visit our practice section to solve more interesting problems

View another problem

Activate Windows

Go to Settings to activate Windows.

Upload code as file

Compile & Run

Submit Code

34°C Partly sunny

ENG IN

2:36 PM

1/11/2023

iv)

WhatsApp

Learn | CodeChef

Celebrity In Trouble - Problems

Celebrity In Trouble Submissions

Solution: 85052979 | CodeChef

codechef.com/problems/SCC5?tab=statement

GmailYouTubeMaps

Celebrity In Trouble

Difficulty Rating: NA

Expand

Statement

Submissions

Solution

Example

Input:
2
10 17
1 8
1 9
2 8
2 6
3 8
3 10
4 7
4 8
10 8
4 10
5 8
6 8
7 8
7 9
5 7
9 8
10 3
10 8
1 5
5 1
5 3
6 1
6 4

C++14

Problem Solver Badge

114 / 250

Solve 136 more problems to get Silver Badge

Status: ✓ Correct Answer

Submission ID: 85053823

Time:
0.27s

Sub-Task	Task #	Result (time)
1	0	AC (0.003737)
Subtask Score: 20.00%		Result - AC
2	1	AC (0.006282)
2	3	AC (0.265689)
2	5	AC (0.074926)

Upload code as file

Compile & Run

Submit Code

Windows

34°C Partly sunny

ENG IN

2:39 PM

1/11/2023

v)

WhatsApp

Minimal Travel Time - Problems

Minimal Travel Time Submission

Solution: 85053315 | CodeChef

codechef.com/problems/CLASSES?tab=statement

GmailYouTubeMaps

Minimal Travel Time

Difficulty Rating: 2379

Expand

Prev Problem

Next Problem

Statement

Submissions

Solution

C++14

0 1
1 2

Problem Solver Badge 115 / 250

Solve 135 more problems to get Silver Badge

Next Problem

Status: Correct Answer

Submission ID: 850533971

Time: 0.15s

Sub-Task	Task #	Result (time)
1	0	AC (0.154403)
1	1	AC (0.039991)
1	2	AC (0.093469)

Activate Windows

Go to Settings to activate Windows.

Upload code as file

Compile & Run

Submit Code

34°C Partly sunny

ENG IN

2:41 PM

1/11/2023

vi)

WhatsApp | Hidden Colored Graph - Problem | codechef.com/problems/COLGRPH?tab=statement

Gmail YouTube Maps

Hidden Colored Graph

Difficulty Rating: 2626 Expand

Prev Problem Next Problem

Statement Submissions Solution C++14

between vertices i and j . Since there are no self-loops, the main diagonal should contain only "0".

If at any time you make an invalid query or exceed the query limit, the interaction is terminated and you will receive a Wrong Answer verdict.

Remember to flush the output after printing each line!

Constraints

- $1 \leq n \leq 100$

Sample 1:

Input	Output
3	7 1
W	
B	7 2
B	7 3
W	7 2
	1
	001
	001

Test against Custom Input

3
W

Problem Solver Badge 116 / 250

Solve 134 more problems to get Silver Badge

Next Problem

Status: ✓ Correct Answer Submission ID: 85054209

Time: 0.02s

Congratulations on solving the problem. Visit our practice section to solve more interesting problems

View another problem

Activate Windows
Go to Settings to activate Windows

Upload code as file

Compile & Run Submit Code

34°C Partly sunny 2:45 PM 1/11/2023

vii)

WhatsApp | Count Arrays - Problems | CodeChef | codechef.com/problems/CNTAR?tab=statement

Gmail YouTube Maps

Count Arrays

Difficulty Rating: 2982 Expand

Prev Problem Next Problem

Statement Submissions Solution C++14

Problem

Given an array A of length N such that $1 \leq A_i \leq N$ and $A_i \neq i, \forall i \in [1, N]$.

Count the number of arrays B of length N such that $\forall i \in [1, N]$:

- $B_i \neq B_{A_i}$
- $1 \leq B_i \leq M$

Since the answer may be large, print it modulo $10^9 + 7$.

Input Format

- The first line contains a single integer T – the number of test cases. The description of T test cases follows.
- Each test case contains 2 lines of input:
 - The first line of each test case contains two space separated integers N, M .
 - The second line of each test case contains N space separated integers A_1, A_2, \dots, A_N .

Output Format

For each test case, output a single integer on a newline - answer modulo $10^9 + 7$.

Constraints

- $1 \leq T \leq 10^5$

Problem Solver Badge 117 / 250

Solve 133 more problems to get Silver Badge

Next Problem

Status: ✓ Correct Answer Submission ID: 85054388

Time: 0.26s

Sub-Task	Task #	Result (time)
1	1	AC (0.261283)
1	2	AC (0.121563)
1	3	AC (0.210513)
1	4	AC (0.243361)
1	5	AC

Activate Windows
Go to Settings to activate Windows

Upload code as file

Compile & Run Submit Code

34°C Partly sunny 2:47 PM 1/11/2023

ix)

WhatsApp

CC Clique Boolean Expression - Probl

CC Clique Boolean Expression Subm

Solution: 46855967 | CodeChef

+

codechef.com/problems/CCNF?tab=statement

Gmail YouTube Maps

Clique Boolean Expression

Difficulty Rating: 3160

Expand

Prev Problem

Next Problem

Statement

Submissions

Solution

C++14

1 4 2 3 7

Consider N boolean variables X_1, X_2, \dots, X_N and their negations $\neg X_1, \neg X_2, \dots, \neg X_N$, corresponding to vertices of the graph. You are given the initial values of all variables.

A CNF boolean expression is defined as a [conjunction](#) of clauses, where each clause is a [disjunction](#) of two or more elements of the set of variables and their negations.

Using the graph G , Ada wrote a CNF boolean expression B with M clauses, where each clause corresponds to a clique in the graph with some (maybe none) of the variables (corresponding to vertices) negated. More formally, for each i from 1 to M :

- Ada chose some clique in the graph, consisting of L_i vertices Z_1, Z_2, \dots, Z_{L_i}
- then for each valid j , Ada chose a variable $P_{i,j}$ as either X_{Z_j} or $\neg X_{Z_j}$
- the i -th clause in the expression B had L_i terms: $P_{i,1} \vee P_{i,2} \vee \dots \vee P_{i,L_i}$

Ada wants the expression to evaluate to "true". In order to achieve that, she may change the values of variables; for each valid i , the cost of changing the value of X_i (from "true" to "false" or vice versa) is C_i . Find the minimum total cost needed to make the expression B evaluate to "true" or determine that it is impossible.

Input

- The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
- The first line of each test case contains three space-separated integers N, M and K .
- The second line contains a single binary string S with length N describing the initial values of the variables. For each valid i , the i -th character of S is '1' if X_i is true or '0' if X_i is false.
- The third line contains N space-separated integers C_1, C_2, \dots, C_N .

Problem Solver Badge 118 / 250

Solve 132 more problems to get Silver Badge

Next Problem

Status: ✓ Correct Answer

Submission ID: [85054950](#)

Time: 0.04s

Sub-Task	Task #	Result (time)
1	0	AC (0.006999)
1	1	AC (0.010547)
1	2	AC (0.006183)

Subtask Score: 25.00%

Result - AC

Activate Windows
Go to Settings to activate Windows

Upload code as file

Compile & Run

Submit Code

WhatsApp

Chef and Edge Flipping - Problem

Chef and Edge Flipping Submiss

Solution: 85054524 | CodeChef

+

codechef.com/problems/EFLIP?tab=statement

Gmail YouTube Maps

Chef and Edge Flipping

Difficulty Rating: 3468

Expand

Prev Problem

Next Problem

Statement

Submissions

Solution

Problem

Read problem statements in [Hindi](#), [Bengali](#), [Mandarin Chinese](#), [Russian](#), and [Vietnamese](#) as well.

Chef likes tournament graphs, which are directed graphs where each unordered pair of vertices is directly connected by exactly one edge. A directed graph is *strongly connected* if for each pair of vertices (a, b) , there is a path from the vertex a to the vertex b .

Consider a tournament graph G with N vertices (numbered 1 through N). Chef takes a sequence of pairs of vertices $(a_1, b_1), (a_2, b_2), \dots, (a_M, b_M)$ and does the following for each i from 1 to M in this order:

- Flip the direction of the edge between vertices a_i and b_i .
- If the graph is strongly connected either before or after flipping this edge, declare G a *bad tournament*.

If G is never declared a bad tournament, Chef calls it a *good tournament*. Given the sequence of M edge flips, find a good tournament.

Input

- The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.

C++14

35 scanf("%d", &T);
36 while (T --) {
37 scanf("%d %d", &n, &m);
38 for (int i = 0; i < m; i++) scanf("%d %d", &a[i], &b[i]);
39 for (int i = 1; i <= n; i++) if (check(i)) break;
40 }
41
42 return 0;
43 }

42:1

Test against Custom Input

Your program will run with no input

Problem Solver Badge 119 / 250

Solve 131 more problems to get Silver Badge

Next Problem

Status: Correct Answer

Submission ID: 85055214

Activate Windows

Go to Settings to activate Windows.

Upload code as file

Compile & Run

Submit Code

Windows Taskbar

34°C Partly sunny

ENG IN

2:58 PM

1/11/2023

WhatsApp

Winter - Problems | CodeChef

Winter Submissions | CodeChef

Solution: 85055388 | CodeChef

codechef.com/problems/ELWINTER?tab=statement

Gmail YouTube Maps

Winter

Difficulty Rating: 2007

Expand

Prev Problem Next Problem

Statement Submissions Solution

C++14

Problem Solver Badge 120 / 250

Solve 130 more problems to get Silver Badge

Next Problem

Problem

Uneori îmi spun: am să mor
atât de singurecă-n mijlocul lor;
limba simplă a bucuriilor n-am învățat;
am să mor ca o pasăre care prea mult a zburat,
dar n-a făcut cuib nicăieri.

— Oamenii mă uimesc, Magda Isafost

As of writing the informal statement, the winter just came and the statement was in season. Now, it is not winter anymore, so here is the formal statement instead:

You are given a connected graph with N nodes and M edges. You are given Q queries of the following types:

- 1 u : Given node u ($1 \leq u \leq N$), set the state of the node u to frozen.
- 2 t : Given t , let t units of time pass by.
- 3 v : Given node v ($1 \leq v \leq N$), answer if node v is currently frozen.

Initially, **no** node is frozen.
The graph has the following property:

- If, at time T , a node u is frozen, then, at time $(T + 1)$, **all neighbours** of u become frozen.

Status: ✓ Correct Answer

Submission ID: [85055668](#)

Time: 0.26s

Sub-Task	Task #	Result (time)
1	0	AC (0.007616)
1	1	AC (0.007643)
1	2	AC (0.007620)
1	3	AC (0.006745)
1	4	AC (0.006289)

Upload code as file

Compile & Run

Submit Code

34°C Partly sunny

ENG IN

3:05 PM 1/11/2023

The screenshot shows a web browser window with multiple tabs. The active tab is 'codechef.com/problems/OMWG?tab=statement'. The page title is 'One more weird game' with a difficulty rating of 1536. The 'Statement' tab is selected, displaying the problem description. The problem involves a grid of N rows and M columns, where Leha's score is the number of colored neighbors of a chosen cell. The goal is to find the maximum possible score.

The 'Input' section states: The first line contains a single integer T denoting the number of test cases. T test cases follow. Each of the following T lines contains two space-separated integers N, M denoting the dimensions of the grid.

The 'Output' section states: For each test case, output a single line containing an integer corresponding to the maximal possible score Leha can obtain.

The 'Constraints' section lists:

- $1 \leq T \leq 100$
- $1 \leq N, M \leq 1\,000$

On the right, the 'Problem Solver Badge' shows 121 / 250 solved problems. A 'Next Problem' button is visible. Below this, the submission status is 'Status: ✓ Correct Answer' with Submission ID: 85055943. The execution time is 0.00s.

Sub-Task	Task #	Result (time)
1	1	AC (0.003896)
Subtask Score: 30.00%		Result - AC
2	2	AC (0.003866)
Subtask Score: 70.00%		Result - AC
Total Score = 100.00%		

At the bottom, there are buttons for 'Upload code as file', 'Compile & Run', and 'Submit Code'. The Windows taskbar at the bottom shows the date as 1/11/2023 and the time as 3:08 PM.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			