```
/*******   Create SCC_to_DAG by Faisal ********/
#include <bits/stdc++.h>
using namespace std;
#define F first
#define S second
#define pb push_back

#define _ ios_base::sync_with_stdio(0);
#define fastt ios_base::sync_with_stdio(false); cin.tie(0);
#define prec  cout<<fixed<<setprecision(16);
//# define sess cout<<endl;

#define file freopen("000.txt","r",stdin);
#define balsal cout<<"dhukse"<<endl;

#define pi acos(-1.0)
#define PI 3.141592653589793
#define INF 1e9

#define si(n)                          scanf("%d",&n)
#define sii(a,b)                       scanf("%d%d",&a,&b)
#define siii(a,b,c)                    scanf("%d%d%d",&a,&b,&c)
#define sl(a)                          scanf("%lld",&a)
#define sll(a,b)                       scanf("%lld%lld",&a,&b)
#define slll(a,b,c)                    scanf("%lld%lld%lld",&a,&b,&c)
#define sf(n)                          scanf("%lf",&n)
#define ss(n)                          scanf("%s",n)

#define it_multiset  std::multiset<int>::iterator it;
#define it_mii std::map<int, int>::iterator it;

#define MOD 1000000007
#define mod  1000000007
//#define con continue;
//#define ret return
#define mx_value 1e19


long long int sett(long long int N,int pos)
{
    return N=N | (1<<pos);
}
int reset(int N,int pos)
{
    return N= N & ~(1<<pos);
}
bool check(int N,int pos)
{
    return (bool)(N & (1<<pos));
}

void update(long long &x, long long val)
{
    x = x+val;
    if(x>mod)x-=mod;
}

void fast()
```

```
{
    ios_base::sync_with_stdio(0);
    cin.tie(NULL), cout.tie(NULL);
}

int moves[4][2] = { {1,0} , {0,1} , {-1,0}, {0,-1} };

typedef long long int ll;
typedef  float flt;
typedef double dbl;
typedef vector<int >  vi;
typedef pair < int , int > pii;
typedef pair < ll , ll > pll;
typedef map<int , int> mii;


#define N 10100

vi front_graph[N];
vi reverse_graph[N];
vi dag[N];

int vis[N]= {};

int bal[N];
int bal_indx=0;
int bal_cnt=0;
int sal[N];
int sal_indx=0;
int sal_cnt=0;

void reset(int n)
{
    for(int i=0; i<=n; i++)
    {
        dag[i].clear();
        vis[i]=0;
        front_graph[i].clear();
        reverse_graph[i].clear();
        bal[i]=0;
        sal[i]=0;
    }
}

void dfs(int node, int type)
{
    vis[node] = 1;
    int loop;
    if(type==1)loop = front_graph[node].size();
    else loop = reverse_graph[node].size();
    for(int i=0; i<loop; i++)
    {
        int next;
        if(type==1) next = front_graph[node][i];
        else next = reverse_graph[node][i];
        if(!vis[next])
        {
            dfs(next,type);
```

```
        }
    }
    if(type==1)
    {
        bal[bal_indx++] = node;
        bal_cnt++;
    }
    else
    {
        sal[sal_indx++] = node;
        sal_cnt++;
    }
}

int main()
{

#ifndef ONLINE_JUDGE
    file;
#endif // ONLINE_JUDGE
    int t;
    si(t);
    int cs = 1;
    while(t--)
    {
        int n,m;
        sii(n,m);
        reset(n);
        for(int i=0; i<m; i++)
        {
            int p,q;
            sii(p,q);
            front_graph[p].pb(q);
            reverse_graph[q].pb(p);
        }


        bal_cnt=0;
        bal_indx=0;

        for(int i=1; i<=n; i++)
        {
            if(!vis[i])
            {
                dfs(i,1);
            }
        }
        map < int , int > map_vertex;
        memset(vis,0,sizeof(vis));
        int cnt = 0;
        for(int i=n-1; i>-1; i--)
        {
            if(!vis[bal[i]])
            {
                sal_indx=0;
                sal_cnt=0;
                cnt++;
                dfs(bal[i],2);
```

```
                for(int j=0; j<sal_cnt; j++)
                {
                    map_vertex[sal[j]] = cnt;  //asol graph r kon inedx dag er kon vertex bujhay
                }
            }
        }
        for(int i=1;i<=n;i++)
        {
            int loop = front_graph[i].size();
            for(int j=0;j<loop;j++)
            {
                int next = front_graph[i][j];
                int a = map_vertex[i];
                int b = map_vertex[next];
               // if(cs==79)cout<<a<<" "<<b<<endl;  //dag er edge gula
                dag[a].push_back(b);
            }
        }
        //ei code e sob vertex 1 theke hisab kora hoise
        //ekhn amra dag paisi
        //ekhn dag er upore ja khushi kora jabe

    }


    return 0;
}
```