https://codeforces.com/contest/950/problem/E 2-sat
https://ac.nowcoder.com/acm/contest/5029/B?keyword=%5B%5B%5E1%5D%5B%5D%2B%2B%5B%5E2%5D%2B-1179 ATM 有向图缩点
https://ac.nowcoder.com/acm/contest/1,&keO1-10001 荒芜的世外 有向图缩点
https://ac.nowcoder.com/acm/contest/5029/A%5B%5D%5B%5D-10360 出差路径 树子节点数

Tarjan

1 dfn=1 low=1
2 dfn=9 low=9
3 dfn=8 low=5
4 dfn=2 low=1
5 dfn=6 low=1
6 dfn=3 low=1
7 dfn=7 low=1
8 dfn=4 low=1
9 dfn=5 low=1

图的dfs

DFS图形 经过一个DFS之后生成各个连通分
量 每个连通分量 是一个 dfs 树
通图的一个连通分量(最小一条单向回路)至少 所知小的点
最向的点是最先经过的点子孙的点并且不是单向的
情况比此点还向回的的情况只可到的此点(本分量有子孙点系,可以使用反证法证明)
都没有子孙关系的话, 那么他其实是孤的, 那么他是孤岛
tips:无向图只有单向图或是孤岛

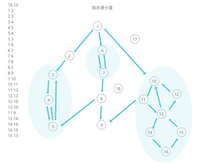
http://www.huasuqia.cn/ion/edge.html

```
int ans=0,ansx=0,ansy=0;
void dfs(int x,int fa)
{
    dfn[x]=low[x]=++idx;
    stk[++top]=x;
    for(int i=head[x];i;i=g[i].nxt)
    {
        int y=g[i].y;
        if (!dfn[y])
        {
            tarjan(y);
            low[x]=min(low[x],low[y]);
        }
        else if (!scc[y]) low[x]=min(low[x],low[y]);

    }
    if (dfn[x]==low[x])
    {
        scc[x]=++sccnt;
        while(stk[top]!=x)
        {
            scc[stk[top]]=sccnt;
            top--;
        }
        scc[stk[top]]=sccnt;
        top--;
    }

set <pair<int,int>> qc;
signed main()
{
    std ::ios ::sync_with_stdio(false),cin.tie(0),cout.tie(0);
    cin >> n >> m;
    for (int i=1; i<=m; i++)
    {
        int x,y;
        cin >> x >> y;
        e.push_back({x,y});
        add(x,y);
    }
    for (int i=1; i<=m; i++) if (!dfn[i]) tarjan(i);
    for (int i=1; i<=m; i++){
        g[i].y=0; g[i].nxt=0;
    }
    for (int i=0; i<=stot-2; i++) head[i]=0;
    etot=0;
    for (int i=0; i<e.size(); i++)
    {
        int x=e[i].first,y=e[i].second;
        x=scc[x],y=scc[y];
        if (x==y ||scc[x]==0||scc[y]==0) continue;
        if (qc.count({x,y}[y]==0)
        {
            add(x,y);
            qc.insert({x,y});
        }
    }
}
```

强连通分量

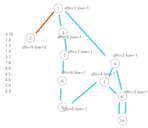


```
low[x]=min(low[x],low[y]);
```

?

```
for(int i=head[x];i;i=g[i].nxt)
{
    int y=g[i].y;
    if (!dfn[y])
    {
        tarjan(y);
        low[x]=min(low[x],low[y]);
    }
    else if (!scc[y]) low[x]=min(low[x],low[y]);
```

割点（桥）

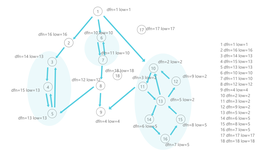


```
if (low[y]>dfn[x])
{
    BRIDGE.push_back(g[i].id);
}
if (i!=(in_edge^1)) low[x]=min(low[x],low[y]);
```

出度 dcc
考虑dfs

```
for (int i=1; i<=n; i++) if (!dcc[i]) { ++dccnt; dfs(i); }
void dfs(int x)
{
    dcc[x]=dccnt;
    for (int i=head[x]; i; i=g[i].nxt)
    {
        int y=g[i].y;
        if (dcc[y] || bridge[i]) continue;
        dfs(y);
    }
}
```

```
if (low[y]>dfn[x])
{
    bridge[i]=bridge[i^1]=true;
}
```



1 dfn=1 low=1
2 dfn=16 low=16
3 dfn=14 low=13
4 dfn=15 low=13
5 dfn=13 low=13
6 dfn=10 low=10
7 dfn=11 low=10
8 dfn=12 low=12
9 dfn=9 low=2
10 dfn=3 low=2
11 dfn=2 low=2
12 dfn=8 low=2
13 dfn=7 low=2
14 dfn=6 low=5
15 dfn=4 low=5
16 dfn=5 low=5
17 dfn=17 low=17
18 dfn=18 low=18

```
dfn[x]=low[x]
```

```
if (dfn[x]==low[x])
{
    sccnt++;
    while(sta[top]!=x)
    {
        scc[sta[top]]=sccnt;
        top--;
    }
    scc[sta[top]]=sccnt;
    top--;
}
```

缩点

```
void tarjan(int x,int fa)
{
    int child=0;
    dfn[x]=low[x]=++idx;
    for (int i=head[x]; i; i=g[i].nxt)
    {
        int y=g[i].y;
        if (!dfn[y])
        {
            tarjan(y,x);
            low[x]=min(low[x],low[y]);
            if (low[y]>=dfn[x] && x!=fa) cut[x]=1;
            if (x==fa) child++;
        }
        else low[x]=min(low[x],dfn[y]);
    }
    if (child==2 && x==fa) cut[x]=1;
    cutot+=cut[x];
}

map<pair<int,int>,bool> qc;
int main()
{
    std ::ios ::sync_with_stdio(false),cin.tie(0),cout.tie(0);
    cin >> n >> m;
    for (int i=1; i<=m; i++)
    {
        int x,y;
        cin >> x >> y;
        if (qc[{x,y}]==0) { add(x,y); qc[{x,y}]=1; }
        if (qc[{y,x}]==0) { add(y,x); qc[{y,x}]=1; }
    }
    cutot=0;
    for (int i=1; i<=n; i++) if (!dfn[i]) tarjan(i,i);
    cout << cutot << '\n';
    for (int i = 1; i <= n; i++) if (cut[i]) cout << i << " ";
}
```

点双 dcc
看成点双连通分量
https://zh.wikipedia.org/zh-cn/%E7%9B%B4%E6%9F%B1%E5%9B%BE%E8%AE%BA

4 5 6

4 5 6 7 8 9

13 14 15

```
#include <bits/stdc++.h>
using namespace std;
const int maxn=5e5+5;
const int mxn=1e6+5;
typedef long long ll;
int n,m;
int idx=0,sccnt=0,top=0;
vector <pair<int,int> e;
int dfn[maxn],low[maxn],scc[maxn],sta[maxn];
int etot=1;
struct edge{
    int y,nxt;
}g[mxn];
int head[maxn];
void add(int x,int y)
{
    ++etot;
    g[etot].y=y;
    g[etot].nxt=head[x];
    head[x]=etot;
}
void tarjan(int x)
{
    dfn[x]=low[x]=++idx;
    stk[++top]=x;
    for(int i=head[x];i;i=g[i].nxt)
    {
        int y=g[i].y;
        if (!dfn[y])
        {
            tarjan(y);
            low[x]=min(low[x],low[y]);
        }
        else if (!scc[y]) low[x]=min(low[x],low[y]);

    }
    if (dfn[x]==low[x])
    {
        sccnt++;
        while(stk[top]!=x)
        {
            scc[stk[top]]=sccnt;
            top--;
        }
        scc[stk[top]]=sccnt;
        top--;
    }

set <pair<int,int>> qc;
signed main()
{
    std ::ios ::sync_with_stdio(false),cin.tie(0),cout.tie(0);
    cin >> n >> m;
    for (int i=1; i<=m; i++)
    {
        int x,y;
        cin >> x >> y;
        e.push_back({x,y});
        add(x,y);
    }
    for (int i=1; i<=m; i++) if (!dfn[i]) tarjan(i);
    for (int i=1; i<=m; i++){
        g[i].y=0; g[i].nxt=0;
    }
    for (int i=0; i<=stot-2; i++) head[i]=0;
    etot=0;
    for (int i=0; i<e.size(); i++)
    {
        int x=e[i].first,y=e[i].second;
        x=scc[x],y=scc[y];
        if (x==y ||scc[x]==0||scc[y]==0) continue;
        if (qc.count({x,y}[y]==0)
        {
            add(x,y);
            qc.insert({x,y});
        }
    }
}
```

```
#include <bits/stdc++.h>
using namespace std;
const int maxn=5e5+5;
const int mxn=1e6+5;
typedef long long ll;
int n,m;
int idx=0,sccnt=0,top=0;
vector <pair<int,int> e;
int dfn[maxn],low[maxn],scc[maxn],sta[maxn];
int etot=1;
struct edge{
    int y,nxt;
}g[mxn];
int head[maxn];
void add(int x,int y)
{
    ++etot;
    g[etot].y=y;
    g[etot].nxt=head[x];
    head[x]=etot;
}

int idx=0,
bool bridge[maxn*2];
int dfn[maxn],low[maxn];
void tarjan(int x,int in_edge) //边双
{
    dfn[x]=low[x]=++idx;
    for (int i=head[x]; i; i=g[i].nxt)
    {
        int y=g[i].y;
        if (!dfn[y])
        {
            tarjan(y,i);
            low[x]=min(low[x],low[y]);
            if (low[y]>dfn[x])
            {
                bridge[i]=bridge[i^1]=true;
            }
        }
        else if (i!=(in_edge^1)) low[x]=min(low[x],low[y]);
    }
}
int dccnt=0;
int dcc[maxn];
int ansx,ansy;
vector <int> G[maxn];
void dfs(int x)
{
    dcc[x]=dccnt;
    for (int i=head[x]; i; i=g[i].nxt)
    {
        int y=g[i].y;
        if (dcc[y] || bridge[i]) continue;
        dfs(y);
    }
}
int main()
{
    std ::ios ::sync_with_stdio(false),cin.tie(0),cout.tie(0);
    cin >> n >> m;
    for (int i=1; i<=m; i++)
    {
        int x,y;
        cin >> x >> y;
        add(x,y); add(y,x);
    }
    for (int i=1; i<=m; i++) if (!dfn[i]) tarjan(i,0);
    for (int i=1; i<=m; i++) if (!dcc[i]) { ++dccnt; dfs(i); }
    for (int i=1; i<=m; i++)
    {
        if (bridge[i])
        {
            int x=g[i].x,
            int y=g[i].y;
            G[x].push_back(y);
            G[y].push_back(x);
            ansx=x,ansy=y;
        }
    }
}
```

```
#include <bits/stdc++.h>
using namespace std;
const int maxn=5e5+5;
const int mod=998244353;
const int inf=0x3f3f3f3f;
const long long INF=0x3f3f3f3f3f3f3f;
const int mxn=1e6+5;
const int Mod=998244353;
typedef long long ll;
int n,m;
int idx=0,
int etot=1;
struct edge{
    int y,nxt;
}g[mxn*2];
int head[maxn];
void add(int x,int y)
{
    ++etot;
    g[etot].y=y;
    g[etot].nxt=head[x];
    head[x]=etot;
}

int idx=0,cutot;
int dfn[maxn],low[maxn],cut[maxn];
void tarjan(int x,int fa)
{
    int child=0;
    dfn[x]=low[x]=++idx;
    for (int i=head[x]; i; i=g[i].nxt)
    {
        int y=g[i].y;
        if (!dfn[y])
        {
            tarjan(y,x);
            low[x]=min(low[x],low[y]);
            if (low[y]>=dfn[x] && x!=fa) cut[x]=1;
            if (x==fa) child++;
        }
        else low[x]=min(low[x],dfn[y]);
    }
    if (child==2 && x==fa) cut[x]=1;
    cutot+=cut[x];
}

map<pair<int,int>,bool> qc;
int ans=0;
int main()
{
    std ::ios ::sync_with_stdio(false),cin.tie(0),cout.tie(0);
    cin >> n >> m;
    for (int i=1; i<=m; i++)
    {
        int x,y;
        cin >> x >> y;
        if (qc[{x,y}]==0) { add(x,y); qc[{x,y}]=1; }
        if (qc[{y,x}]==0) { add(y,x); qc[{y,x}]=1; }
    }
    cutot=0;
    for (int i=1; i<=n; i++) if (!dfn[i]) tarjan(i,i);
    cout << cutot << '\n';
    for (int i = 1; i <= n; i++) if (cut[i]) cout << i << " ";
}
```

```
#include <bits/stdc++.h>
using namespace std;
const int inf=0x3f3f3f3f;
const long long INF=0x3f3f3f3f3f3f3f;
const int maxn=1e6+10;
const int mod=998244353;
const int Mod=1e9+7;
typedef long long ll;
ll n,m;
ll idx=0,sccnt=0;
vector <int> g[maxn];
vector <pair<int,int>> e;
int dfn[maxn],low[maxn],scc[maxn],sta[maxn];

void tarjan(int x,int fa)
{
    dfn[x]=low[x]=++idx;
    sta[++top]=x;
    for(int i=0; i<g[x].size(); i++)
    {
        int y=g[x][i];
        if (y==fa) continue;
        if (!dfn[y])
        {
            tarjan(y,x);
            low[x]=min(low[x],low[y]);
            if (low[y]>dfn[x])
            {
                while (1)
                {
                    int tmp=sta[top];
                    scc[tmp]=y, top--;
                    if (tmp==y) break;
                }
            }
        }
        else low[x]=min(low[x],low[y]);
    }
}

vector <int> G[maxn];
map <pair<int,int>,bool> qc;
void init()
{
    ++sccnt;
    for (int i=0; i<=n+1; i++)
        dfn[i]=0, scc[i]=0, low[i]=0,
        g[i].clear(), G[i].clear();
}

qc.clear();
idx=0; top=0;
}

void solve()
{
    cin >> n >> m; init();
    for (int i=1; i<=m; i++) {
        int x,y; cin >> x >> y;
        g[x].push_back(y);
        g[y].push_back(x);
        e.push_back({x,y});
    }
    for (int i=1; i<=n; i++)
        if (!dfn[i]) {
            tarjan(i, 1);
            ans[i]=1;
        }
    for (auto it e) {
        int x=it.first;
        int y=it.second;
        x=ans[x],y=ans[y];
        if (x==y || ans[x]==0||ans[y]==0) continue;
        if (qc.count({x,y}[y]==0) {
            qc[{x,y}]=1;
            G[x].push_back(y);
            G[y].push_back(x);
        }
    }
}

int main()
{
    std ::ios ::sync_with_stdio(false),cin.tie(0),cout.tie(0);
    ll T=1; cin >> T;
    while (T--) { solve(); }
}
```