```cpp
// strongly connected component (short) by Shadman

#include <bits/stdc++.h>
using namespace std;
int indeg[2001],outdeg[2001];
vector<int>v1[2002],dg[2002];

int pre[2001], lowlink[2001], sccno[2001], dfs_clock, scc_cnt;
stack<int> S;

void dfs_scc(int u) {
    pre[u] = lowlink[u] = ++dfs_clock;
    S.push(u);
    for (int i = 0; i < v1[u].size(); i++) {
        int v = v1[u][i];
        if (!pre[v]) {
            dfs_scc(v);
            lowlink[u] = min(lowlink[u], lowlink[v]);
        } else if (!sccno[v])
            lowlink[u] = min(lowlink[u], pre[v]);
    }
    if (lowlink[u] == pre[u]) {
        scc_cnt++;
        while (!S.empty()) {
            int x = S.top(); S.pop();
            sccno[x] = scc_cnt;
            if (x == u) break;
        }
    }
}

void find_scc(int n) {
    dfs_clock = scc_cnt = 0;
    memset(sccno, 0, sizeof(sccno));
    memset(pre, 0, sizeof(pre));
    for (int i = 0; i < n; i++)
        if (!pre[i]) dfs_scc(i);
}
int cnt(int node)
{
    if(outdeg[node])return outdeg[node];

    int i,e,mx=0,x1;
    e=dg[node].size();
    for(i=0;i<e;i++)
    {
        x1=dg[node][i];
        mx=max(mx,cnt(x1));
    }
    outdeg[node]=mx+indeg[node];
    return outdeg[node];


}

int main()
{
    int a,b,c,d,e,f,i,j,k,p,q,T;
    scanf("%d",&T);
    for(i=1;i<=T;i++)
    {   memset(indeg,0,sizeof(indeg));
        memset(outdeg,0,sizeof(outdeg));
        scanf("%d %d",&a,&b);


        for(j=0;j<=2000;j++){
                v1[j].clear();
                dg[j].clear();
        }



        for(j=1;j<=b;j++)
        {
            scanf("%d %d",&c,&d);
            c--;
            d--;

            //v2.push_back(pr);
            v1[c].push_back(d);
        }
         find_scc(a);

    for(j=0;j<a;j++)
    {
        indeg[sccno[j]]++;
    }

    for(j=0;j<a;j++)
    {
        e=v1[j].size();
        for(k=0;k<e;k++)
        {
            f=v1[j][k];
            if(sccno[j]!=sccno[f])
            {
                dg[sccno[j]].push_back(sccno[f]);
            }
        }
    }

    q=0;
    for(j=0;j<a;j++)
    {
        q=max(q,cnt(sccno[j]));
    }

        printf("%d\n", q);
    }
    return 0;


}
```