```cpp
/*** Uva 11504- Dominos:
Given the layout of some domino tiles, Your task is to determine, the minimum number of dominos
that must be knocked down by hand in order for all of the dominos to fall.

The first line of input contains one integer specifying the number of test cases to follow.
Each test case begins with a line containing two integers, each no larger than 100000.
The first integer n is the number of domino tiles and the second integer m is the number of
lines to follow in the test case. The domino tiles are numbered from 1 to n.
Each of the following m lines contains two integers x and y indicating that if domino number
x falls, it will cause domino number y to fall as well.

For each test case, output a line containing one integer, the minimum number of dominos
that must be knocked over by hand in order for all the dominos to fall.
***/
#include<bits/stdc++.h>
using namespace std;
#define mx 100005
vector<int>ed[mx];
stack<int>st;
bool vis[mx];
void TOPSORT(int u){
    vis[u]=true;
    for(int i=0; i<ed[u].size(); i++){
        int v = ed[u][i];
        if(vis[v]==false) TOPSORT(v);
    }
    st.push(u);
}
void DFS(int u){
    vis[u]=true;
    for(int i=0; i<ed[u].size(); i++){
        int v = ed[u][i];
        if(vis[v]==false) DFS(v);
    }
}
int main(){
    int tt; scanf("%d",&tt);
    for(int ks=1; ks<=tt; ks++){
        int n,m; scanf("%d%d",&n,&m);

        for(int i=1; i<=m; i++){
            int u,v; scanf("%d%d",&u,&v);
            ed[u].push_back(v);
        }

        memset(vis,false,sizeof(vis));
        for(int i=1; i<=n; i++){
            if(vis[i]==false){
                TOPSORT(i);
            }
        }

        int ans = 0;
        memset(vis,false,sizeof(vis));

        while(!st.empty()){
            int u = st.top(); st.pop();
            if(vis[u]==false){
                ans++;
                DFS(u);
            }
        }
        printf("%d\n",ans);

        for(int i=1; i<=n; i++)ed[i].clear();
    }
    return 0;
}
```