/\*

两点之间若有唯一路径，则该路径上的边为无向图的桥，这些桥组成的边构成树，统计每棵树上的路径和为答案

\*/

#include<iostream>

#include<cstdio>

#include<cstring>

#include<vector>

#define N 500001

using namespace std;

vector<int> G[N];

vector<int> g[N];

int n,m,low[N],dfn[N];

bool is\_cut[N];

int father[N];

int new\_g[N];

int tim=0;

void input()

{

scanf("%d%d",&n,&m);

int a,b;

for(int i=1;i<=m;++i)

{

scanf("%d%d",&a,&b);

G[a].push\_back(b);

G[b].push\_back(a);

}

}

void Tarjan(int i,int Father)

{

father[i]=Father;

dfn[i]=low[i]=tim++;

for(int j=0;j<G[i].size();++j)

{

int k=G[i][j];

if(dfn[k]==-1)

{

Tarjan(k,i);

low[i]=min(low[i],low[k]);

}

else if(Father!=k)

low[i]=min(low[i],dfn[k]);

}

}

long long ans=0;

int cnt[N]={0};

int dfs(int u,int fa)

{

int sum=1;

new\_g[u]=false;

for(int i=0;i<g[u].size();i++)

{

int v=g[u][i];

if(v!=fa)

{

sum+=dfs(v,u);

}

}

cnt[++cnt[0]]=sum;

return sum;

}

void count()

{

int rootson=0;

for(int i=1;i<=n;i++)

{

tim=0;

if(!father[i])Tarjan(i,0);

}

for(int i=1;i<=n;i++)g[i].clear();

for(int i=1;i<=n;++i)

{

int v=father[i];

if(v>0&&low[i]>dfn[v])

{

g[v].push\_back(i);

g[i].push\_back(v);

new\_g[i]=true;

new\_g[v]=true;

}

}

for(int i=1;i<=n;i++)

{

G[i].clear();

if(new\_g[i])

{

int sum=dfs(i,0);

for(int i=1;i<=cnt[0];i++)ans+=(sum-cnt[i])\*cnt[i];

cnt[0]=0;

}

}

}

int main()

{

int T;

cin>>T;

while(T--)

{

ans=0;

input();

memset(dfn,-1,sizeof(dfn));

memset(father,0,sizeof(father));

memset(low,-1,sizeof(low));

memset(is\_cut,false,sizeof(is\_cut));

count();

cout<<ans<<endl;

}

}