**Two-Sat, >O(n)**

**Tested on LightOJ 1251 - Forming the Council**

bool vis[MX];

vector<int>topo, scc[MX], edge[MX],edge\_rev[MX], scc\_scc[MX], edge\_scc[MX];

int sccrm[MX],nod,rev[MX],res[MX];

void dfstopo(int node)

{

vis[node]=1;

for(int i=0;i<edge[node].size();i++)

{

int v=edge[node][i];

if(!vis[v]) dfstopo(v);

}

topo.push\_back(node);

return;

}

void dfstopo\_scc(int node)

{

vis[node]=1;

for(int i=0;i<edge\_scc[node].size();i++)

{

int v=edge\_scc[node][i];

if(!vis[v]) dfstopo\_scc(v);

}

topo.push\_back(node);

return;

}

void dfsscc(int node, int cnt)

{

vis[node]=1;

scc[cnt].push\_back(node);

sccrm[node]=cnt;

for(int i=0;i<edge\_rev[node].size();i++)

{

int v=edge\_rev[node][i];

if(!vis[v]) dfsscc(v, cnt);

}

return;

}

void dfs\_scc\_mark(int node)

{

vis[node]=1;

for(int i=0;i<scc[node].size();i++)

{

int v=scc[node][i];

res[v]=1;

}

for(int i=0;i<edge\_scc[node].size();i++)

{

int v=edge\_scc[node][i];

if(!vis[v]) dfs\_scc\_mark(v);

}

return;

}

bool two\_sat()

{

CLR(vis);

topo.clear();

for(int i=1;i<=nod\*2;i++)

if(!vis[i])

dfstopo(i);

int cnt=1;

CLR(vis);

for(int i=topo.size()-1;i>=0;i--)

if(!vis[topo[i]])

{

scc[cnt].clear();

dfsscc(topo[i], cnt++);

}

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

if(sccrm[i]==sccrm[i+nod])

return false;

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

edge\_scc[i].clear();

set<int>tmp[cnt];

set<int>:: iterator it;

for(int i=nod\*2;i>0;i--)

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

{

int v=edge[i][j];

tmp[sccrm[i]].insert(sccrm[v]);

}

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

for(it=tmp[i].begin();it!=tmp[i].end();it++)

if(i!=\*it)

edge\_scc[i].push\_back(\*it);

topo.clear();

CLR(vis);

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

if(!vis[i]) dfstopo\_scc(i);

CLR(res);

CLR(vis);

for(int i=topo.size()-1;i>=0;i--)

if(!vis[topo[i]])

{

vis[topo[i]]=1;

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

{

int v=scc[topo[i]][j];

dfs\_scc\_mark(sccrm[rev[v]]);

}

}

return true;

}

int main()

{

int tc,kk=1,n,m,edg,u,v;

cin>>tc;

while(tc--)

{

cin>>edg>>nod;

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

{

rev[i]=nod+i;

rev[i+nod]=i;

edge[i].clear(); edge[i+nod].clear();

edge\_rev[i].clear(); edge\_rev[i+nod].clear();

}

CLR(sccrm);

while(edg--)

{

cin>>u>>v;

if(u<0) u=nod-u;

if(v<0) v=nod-v;

edge[rev[u]].push\_back(v);

edge[rev[v]].push\_back(u);

edge\_rev[v].push\_back(rev[u]);

edge\_rev[u].push\_back(rev[v]);

}

printf("Case %d: ",kk++);

if(two\_sat())

{

cout<<"Yes"<<endl;

vector<int>ans;

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

if(res[i]) ans.push\_back(i);

cout<<ans.size();

for(int i=0;i<ans.size();i++)

cout<<" "<<ans[i];

cout<<endl;

}

else cout<<"No"<<endl;

}

return 0;

}