**Rhezo and Destructive Mind**

Attempted by: **1080**

/

Accuracy: **66%**

/

Maximum Points: **10**

/

15 Votes

/

Depth-first search, Graph Theory, Very-Easy

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

[**DISCUSSIONS**](https://www.hackerearth.com/practice/algorithms/graphs/articulation-points-and-bridges/practice-problems/algorithm/rhezo-and-destructive-mind/discussion/)NEW

Rhezo has bought a graph with *N* nodes and *M* edges. He has a destructive mind and often deletes some nodes(and of course its edges to other nodes of the graph).

He will ask you *Q* queries. Each query is denoted by an integer *X*, meaning that he will delete the node *X*. Since he has a destructive mind, he gets satisfied if the the new graph with deleted node has more number of connected components than the original graph.

If satisfied, he wants you to print "Satisfied"(without quotes) and the number of edges that were removed when deleting node *X*, else print "Not Satisfied"(without quotes). All queries are different, and the original graph is restored after each query.

**Input:**

First line contains *N* and *M* as described in the problem statement. Each of the next *M* lines contain 2 integers Ai and Bi, meaning that there is an edge between node *A* and node *B*. Next line contains a single integer *Q*, denoting the number of queries. Each of the next *Q* lines contain a single integer *X* as described in the problem statement.

**Output:**

For each of the *Q* queries, if Rhezo is satisfied(that is the number of connected components of the new graph has increased), print "Satisfied"(without quotes), and the number of edges that will be deleted after a space. Else print "Not Satisfied".

**Constraints:**

1≤N,M,Q≤102

1≤Ai,Bi≤N

1≤X≤N

**SAMPLE INPUT**

5 5

1 2

2 3

3 4

4 5

3 5

5

1

2

3

4

5

**SAMPLE OUTPUT**

Not Satisfied

Satisfied 2

Satisfied 3

Not Satisfied

Not Satisfied

**Explanation**

For the second query, if we remove node 2, the graph disconnects and 2 connected components are formed. One having only {1} and other having {3,4,5}. Clearly, the number of edges removed are 2.

* SOLUTION:

#include<bits/stdc++.h>

#define pb push\_back

using namespace std;

vector<int> adj[101];

int cut\_vertex[101];

int vis[101];

int low[101],tin[101],timer;

int dfs(int node,int par)

{

vis[node]=1;

low[node]=tin[node]=timer++;

int children=0;

for(int child: adj[node])

{

if(child==par) continue;

if(vis[child])

low[node]=min(tin[child],low[node]);

else

{

dfs(child,node);

low[node]=min(low[child],low[node]);

children++;

if(low[child]>=tin[node] && par!=-1)

{

cut\_vertex[node]=1;

}

}

}

if(par==-1 && children>1)

cut\_vertex[node]=1;

}

int main()

{

int n,m;

cin>>n>>m;

for(int i=0;i<m;i++)

{

int u,v;

cin>>u>>v;

adj[u].pb(v);

adj[v].pb(u);

}

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

if(!vis[i])

dfs(i,-1);

int q;

cin>>q;

while(q--)

{

int a;

cin>>a;

if(cut\_vertex[a])

cout<<"Satisfied "<<adj[a].size()<<"\n";

else

cout<<"Not Satisfied\n";

}

}