**Rhezo and HackerEarth**

Attempted by: **899**

/

Accuracy: **45%**

/

Maximum Points: **20**

/

15 Votes

/

Depth-first search, Graph Theory

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

[**DISCUSSIONS**](https://www.hackerearth.com/practice/algorithms/graphs/articulation-points-and-bridges/practice-problems/algorithm/rhezo-and-hackerearth-3/discussion/)NEW

Rhezo has just joined HackerEarth as an intern. Like most companies, HackerEarth too has a lot of computers and some departments. For efficient communication, it is very important that every computer can connect with each other in the same department, with the connection not being necessarily direct.

When each computer can connect with any other computer in the same department, that department is said to be happy. Initially all departments are happy.

Rhezo's friend Lonewolf wants to have some fun, and will cut some connection between any two computers for *Q* days. You being the Happy Manager of HackerEarth, need to tell for each of the *Q* days, if some department becomes unhappy.

**Input:**

First line contains 2 integers *N* and *M*, denoting the number of computers and number of wire connections at HackerEarth. Each of the next *M* lines contain 2 integers Ai and Bi, meaning that computer number Ai is connected to computer number Bi.

Next line contains an integer *Q*, denoting the number of days Lonewolf cuts a connection between 2 computers at HackerEarth. Each of the next *Q* lines contain a single integer *P*, which means that connection between computer Ap and computer Bp is cut.

**Output:**

For each of the *Q* days, you need to tell whether any department is unhappy or not. If all departments are happy, print "Happy"(without quotes), else print "Unhappy"(without quotes).

**Constraints:**

1≤N,M≤105

1≤Ai,Bi≤N

1≤P≤M

1≤M≤max(105,N⋅(N−1)/2)

**SAMPLE INPUT**

5 5

1 2

2 3

3 4

4 5

3 5

5

1

2

3

4

5

**SAMPLE OUTPUT**

Unhappy

Unhappy

Happy

Happy

Happy

**Explanation**

For the 3rd query, when we delete the connection between computer *3* and computer *4*, these two computers can still connect via this connection : 3→5→4.

#include<bits/stdc++.h>

#define pb push\_back

using namespace std;

vector<int> adj[100001];

map<pair<int,int>,int> bridge;

int vis[100001];

int low[100001],tin[100001],timer;

void dfs(int node,int par)

{

vis[node]=1;

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

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]);

if(low[child]>tin[node])

{

bridge[{node,child}]=1;

bridge[{child,node}]=1;

}

}

}

}

int main()

{

int n,m;

cin>>n>>m;

int ar1[m+1],ar2[m+1];

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

{

int u,v;

cin>>u>>v;

adj[u].pb(v);

adj[v].pb(u);

ar1[i]=u;

ar2[i]=v;

}

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

if(!vis[i])

dfs(i,-1);

int q;

cin>>q;

while(q--)

{

int a;

cin>>a;

if(bridge[{ar1[a],ar2[a]}])

cout<<"Unhappy\n";

else

cout<<"Happy\n";

}

}