

<https://course.acciojob.com/idle?question=f10e32c0-3a7f-450c-a074-541fc8e7540f>

MEDIUM

Max Score: 40 Points

Is Graph Tree?

Ben Tennyson recently joined Coding Academy to become better at competitive programming.

To check whether Ben is seriously studying at Coding Academy or not, Gwen challenged Ben to solve a problem and Ben accepts the challenge of Gwen. Gwen gives Ben an undirected graph of N nodes numbered from 0 to $N - 1$ having M edges. The task of Ben is to check whether the graph given by Gwen is a tree or not. As Ben is busy fighting with Evil doers so he asked you to solve this problem.

Note

There are no parallel edges and self-loops in the graph given by Gwen.

A tree is a connected graph having no cycles.

Input Format

The first line contains two space-separated integers N and M . Here N denotes the total number of nodes and M denotes the total number of edges in the graph given by Gwen.

The next M lines contain two space-separated integers u and v . Here u and v represent the nodes that share an edge between them.

Output Format

Print 1 if the graph given by Gwen is a tree, else you need to print 0 .

Example 1

Input

```
7 6
0 1
0 4
1 2
1 3
4 5
4 6
```

Output

```
1
```

Explanation

Given graph is a connected graph with no cycles. Hence this graph is a tree. So we need to print 1.

Example 2

Input

```
3 3
0 1
0 2
1 2
```

Output

```
0
```

Explanation

Given graph is a connected graph but it has a cycles. Hence this graph is not a tree. So we need to print 0.

Constraints

$1 \leq N \leq 5000$

$1 \leq M \leq \min(5000, N*(N-1)/2)$

$1 \leq U, V \leq N - 1$

Topic Tags

Graphs

Trees

My code

```
// in java
import java.util.*;
import java.lang.*;
import java.io.*;

public class Main
{
    public static void main (String[] args) throws
java.lang.Exception
    {
        //your code here
        Scanner s=new Scanner(System.in);
        int n=s.nextInt();//no of node
        int m=s.nextInt();//no of eg
        int a[]=new int[n];
        for(int i=0;i<n;i++)
            a[i]=0;
        for(int i=0;i<2*m;i++)
        {
            int t=s.nextInt();
```

```

        a[t]=a[t]+1;
    }
    if(m!=(n-1))//we must have n-1 eg for uncycle + connectede
    {
        System.out.print("0");
        return;
    }
    for(int i=0;i<n;i++)
        if(a[i]==0)//if degree=0 ie disconnected
        {
            System.out.print("0");
            return;
        }
    System.out.print("1");
}
}

```