DFS Traversal 5

Given an undirected and disconnected graph G(V, E), print its DFS traversal.
Note:
Here you need to consider that you need to print DFS path starting from vertex 0 only.
V is the number of vertices present in graph G and vertices are numbered from 0 to V-1.
E is the number of edges present in graph G.
Take graph input in the adjacency matrix.
Handle for Disconnected Graphs as well

Input Format

The first line of input contains two integers, that denote the value of V and E.
Each of the following E lines contains space separated two integers, that denote that there exists an edge between vertex a and b.

Constraints

0 <= V <= 1000
0 <= E <= (V * (V - 1)) / 2
0 <= a <= V - 1

0 <= b <= V - 1

Time Limit: 1 second

Output Format

Print the DFS Traversal, as described in the task.

Sample Input 0

```
4 4
0 1
0 3
1 2
2 3
```

Sample Output 0

```
0 1 2 3
```

Contest ends in 17 days

Submissions: 103
Max Score: 10
Difficulty: Medium

Rate This Challenge:

More

Java 15

```
1 ▾import java.io.*;
```

```java
import java.util.*;

public class Solution {
    static void display(int mat[][],int st,int v,boolean vis[]){
        System.out.print(st+" ");
        vis[st]=true;
        for(int en=0;en<v;en++){
            if(mat[st][en]==1 && !vis[en]){
                display(mat,en,v,vis);
            }
        }
    }
    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
        Scanner sc=new Scanner(System.in);
        int v=sc.nextInt();
        int e=sc.nextInt();
        int mat[][]=new int[v][v];
        for(int i=0;i<e;i++){
            int sv=sc.nextInt();
            int ev=sc.nextInt();
            mat[sv][ev]=1;
            mat[ev][sv]=1;
        }
        boolean vis[]=new boolean[v];
        display(mat,0,v,vis);
    }
}
```

Line: 9 Col: 30

Upload Code as File Test against custom input Run Code Submit Code

Testcase 0 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
4 4
0 1
0 3
1 2
2 3
```

Your Output (stdout)

```
0 1 2 3
```

Expected Output

```
0 1 2 3
```