

Преброяване на области

locked

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

В държавата Графана имало множество градове и пътища между тях. Градовете били разделени в области, като всяка област се характеризира с това, че всички градове в нея са свързани само помежду си(няма връзки към градове от друга област). Направете програма, която по дадени градове и връзките в тях да отговори колко е броя на областите в държавата.

Input Format

T - брой тестове
X Y - брой градове и брой пътища
x_i x_j - указва, че има връзка между два града (всички пътища са двупосочни, като може да има по-повече от един път между два града)

Constraints

1 < T <= 20
1 < X <= 100,000
0 < Y <= 500,000

Output Format

N1 N2 .. NT - броя на областите във всеки тест

Sample Input 0

```
2
5 3
2 0
0 3
3 0
5 3
3 3
1 4
0 1
```

Sample Output 0

```
3 3
```

Current Buffer (saved locally, editable)

C++

```
1 #include <cmath>
2 #include <cstdio>
3 #include <vector>
4 #include <iostream>
5 #include <algorithm>
6 #include <list>
7 using namespace std;
8
9 struct Node{
10     list<int> neighbours;
11     bool hasNeighbour(int index){
12         for(auto it:neighbours){
13             if(it==index){
14                 return true;
15             }
16         }
17         return false;
18     }
19 }
20 void addNeighbour(int index){
21     neighbours.push_back(index);
22 }
23 };
24 struct Graph{
25     vector<Node> nodes;
26     Graph(int nodeCount=0){
27         nodes.resize(nodeCount);
28     }
29     void connect(int from,int to){
30         nodes[from].addNeighbour(to);
31         nodes[to].addNeighbour(from);
32     }
33     void dfs(int start,vector<bool>& visited){
34         visited[start]=true;
35         for(auto neighbour:nodes[start].neighbours){
36             if(visited[neighbour]==false){
37                 dfs(neighbour,visited);
38             }
39         }
40     }
41     int dfsAll(){
42         int result=0;
43         vector<bool> visited(nodes.size(),false);
44         for(int i=0;i<nodes.size();i++){
45             if(visited[i]==false){
46                 result++;
47                 dfs(i,visited);
48             }
49         }
50         return result;
51     }
52 };
53
54 int main(){
55     int testCount;
56     cin>>testCount;
57     for(int i=0;i<testCount;i++){
58         int nodeCount,edgeCount;
59         cin>>nodeCount>>edgeCount;
60         int from,to,weight;
61         Graph g(nodeCount);
62         for(int j=0;j<edgeCount;j++){
63             cin>>from>>to;
64             g.connect(from,to);
65         }
66         cout<<g.dfsAll()<<" ";
67     }
68     return 0;
69 }
```

Line: 1 Col: 1