Description | Solution | Discuss (999+) | Submissions

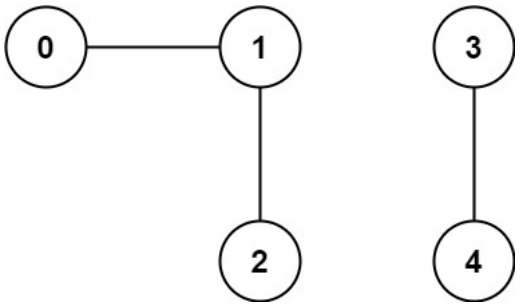
323. Number of Connected Components in an Undirected Graph

Medium 2354 80 Add to List Share

You have a graph of `n` nodes. You are given an integer `n` and an array `edges` where `edges[i] = [a_i, b_i]` indicates that there is an edge between `a_i` and `b_i` in the graph.

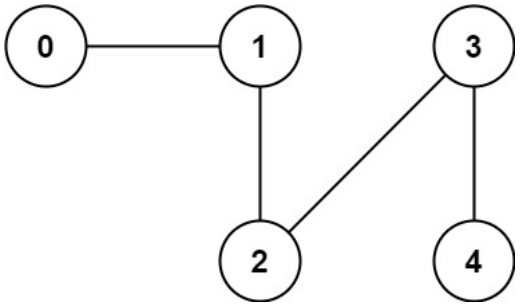
Return *the number of connected components in the graph*.

Example 1:




```
Input: n = 5, edges = [[0,1],[1,2],[3,4]]
Output: 2
```

Example 2:




```
Input: n = 5, edges = [[0,1],[1,2],[2,3],[3,4]]
Output: 1
```

Constraints:

- `1 <= n <= 2000`
- `1 <= edges.length <= 5000`
- `edges[i].length == 2`
- `0 <= a_i <= b_i < n`
- `a_i != b_i`
- There are no repeated edges.

Accepted 312,692 Submissions 503,122

Seen this question in a real interview before? Yes No

Companies

Related Topics

Similar Questions

C++ Autocomplete

```cpp
class Solution {
public:
    void dfs(int parent,vector<bool> &visited,vector<int>adj[])
    {
        visited[parent] = true;
        for(int children : adj[parent])
        {
 if(!visited[children])
 dfs(children,visited,adj);
        }
    }
    int countComponents(int n, vector<vector<int>>& edges) {
        vector<int>adj[n];

        for(int i = 0;i<edges.size();i++)
        {
            adj[edges[i][0]].push_back(edges[i][1]);
            adj[edges[i][1]].push_back(edges[i][0]);
        }


 vector<bool>visited(n,false);

        int count = 0;

        for(int i=0;i<n;i++)
        {
            if(!visited[i])
            {
                count++;

 dfs(i,visited,adj);
            }
        }
        return count;
    }
};
```

NEW

Testcase | Run Code Result | Debugger

Accepted Runtime: 2 ms

Your input: 5 [[0,1],[1,2],[2,3],[3,4]]

Output: 1 Diff

Expected: 1

Console Use Example Testcases

Problems | Pick One | Prev | /150 | Next | Run Code | Submit