Problem 323: Number of Connected Components in an Undirected Graph

Problem Information

Difficulty: Medium
Acceptance Rate: 64.56%
Paid Only: Yes
Tags: Depth-First Search, Breadth-First Search, Union Find, Graph

Problem Description

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

Return _the number of connected components in the graph_.

Example 1:

![](https://assets.leetcode.com/uploads/2021/03/14/conn1-graph.jpg)

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

Example 2:

![](https://assets.leetcode.com/uploads/2021/03/14/conn2-graph.jpg)

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 <= ai <= bi < n` * `ai != bi` * There are no repeated edges.

Code Snippets

C++:

```
class Solution {
public:
int countComponents(int n, vector<vector<int>>& edges) {

}
};
```

Java:

```
class Solution {
public int countComponents(int n, int[][] edges) {

}
}
```

Python3:

```
class Solution:
def countComponents(self, n: int, edges: List[List[int]]) -> int:
```