Problem 547: Number of Provinces

Problem Information

Difficulty: Medium
Acceptance Rate: 69.59%
Paid Only: No
Tags: Depth-First Search, Breadth-First Search, Union Find, Graph

Problem Description

There are `n` cities. Some of them are connected, while some are not. If city `a` is connected directly with city `b`, and city `b` is connected directly with city `c`, then city `a` is connected indirectly with city `c`.

A **province** is a group of directly or indirectly connected cities and no other cities outside of the group.

You are given an `n x n` matrix `isConnected` where `isConnected[i][j] = 1` if the `ith` city and the `jth` city are directly connected, and `isConnected[i][j] = 0` otherwise.

Return _the total number of**provinces**_.

Example 1:

Input: isConnected = [[1,1,0],[1,1,0],[0,0,1]] **Output:** 2

Example 2:

Input: isConnected = [[1,0,0],[0,1,0],[0,0,1]] **Output:** 3

Constraints:

* `1 <= n <= 200` * `n == isConnected.length` * `n == isConnected[i].length` * `isConnected[i][j]` is `1` or `0`. * `isConnected[i][i] == 1` * `isConnected[i][j] == isConnected[j][i]`

Code Snippets

C++:

```
class Solution {
public:
int findCircleNum(vector<vector<int>>& isConnected) {

}
};
```

Java:

```
class Solution {
public int findCircleNum(int[][] isConnected) {

}
}
```

Python3:

```
class Solution:
def findCircleNum(self, isConnected: List[List[int]]) -> int:
```