75 Days of Code
Day 44
Problem no : 547
Problem Title : Number of Provinces
Type : Graph / DFS

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

Shubham Agrahari

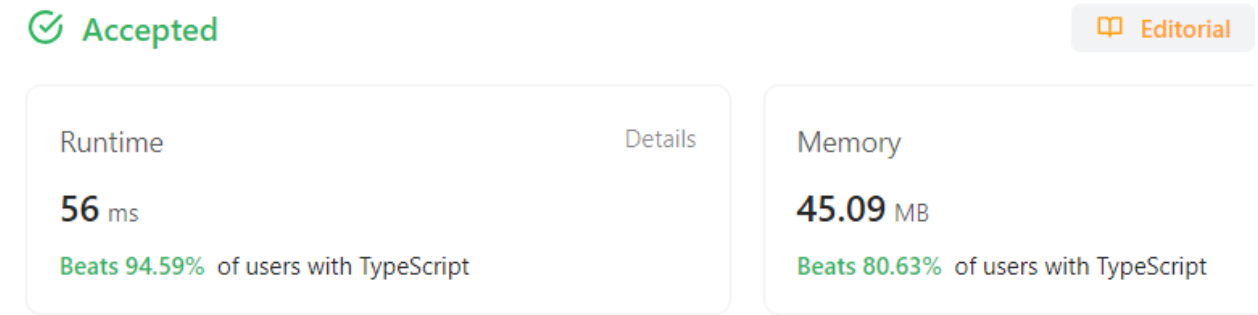
Solution

Using DFS works for this problem:

```typescript
function findCircleNum(isConnected: number[][]): number {
  const visited = new Set<number>();
  let Length = isConnected.length;
  let province = 0;

  const dfs = (node: number): void => {
    visited.add(node);
    for (let neighbour = 0; neighbour < Length; neighbour++) {
      if (isConnected[node][neighbour] === 1 && !visited.has(neighbour)) {
        dfs(neighbour);
      }
    }
  };
  for (let city = 0; city < Length; city++) {
    if (!visited.has(city)) {
      dfs(city);
      province++;
    }
  }

  return province;
}
```



Shubham Agrahari

Shubham Agrahari