LEETCODE | 547|

Number of Provinces

Input : isConnected = $\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Output : 2

Approach Graph Problem → Let's use DFS
We will try to go in all path available (unvisited)
& try to find disconnected component → no. of provinces

Step1

```
int m = isConnected.length;
int numProvinces = 0;
int [] visited = new int [m];
```

Step2 Run a forLoop and start from 0th index of visited

```
for (int i=0; i<m; i++){
        if (visited [i]==0){
            dFSGraph (isConnected, visited, m, i);
            numProvinces++;
        }
}
```

Step3

```
void dFSGraph(isConnected, visited, m, current){
↳       visited[current] =1;
        for (int i=0; i<m, i++){
            if (i != current && isConnected[current][i]
            ==1 && visited [i]==0){
            dFSGraph(isConnected, visited, m, i);
        }
    }
```

Matrix
(i != current)
↳ diagonal visit is irrelevant (same city)

Step4 Return numProvinces;