Number of Islands & Leetcode (200)

Question: m×n 2D binary grid   1 → land 0 → water

An island is surrounded by water formed by connecting adjacent lands horizontally / vertically & not diagonally.

We can assume all 4 edges of grid are surrounded by water

Approach:

- Start from the leftmost topmost point of grid
- if you encounter a land(1) keep looking around in all possible direction for adjacent lands and continue to do this until an island is formed.
- Once an island is found increase the count of no. of islands.
- Traversal is done :→ for every row check every column.
  So if its already visited in the previous call of dfs (make sure it was changed from 1 to 0.
- So when you encounter '1' it was definitely not part of the previous islan
  Start the dfs again
- At last return the count (no. of islands).

Pseudocode: **Main fn**

```
for every row
    Check every column
        if land (1)
            call dfs f^n.
            increase count
```

note count will be increased when for a (i, j) dfs is done on allpossible ways completely

Dfs fn (private)

```
if (i, j) ∈ grid & grid[i][j] = 1
                        (land)
    change it to '0'
    grid[i][j] = 0
    then do dfs on ~~p~~ possible
                adjacent dir^nS.
```

$$\begin{tikzcd} & \uparrow & \\ \leftarrow & (i,j) & \rightarrow \\ & \downarrow & \end{tikzcd}$$

Code: **Main fn**

```
public int numIslands (char[][] grid){
    int count = 0;
    for(int i=0; i<grid.length; i++)
        for(int j=0; j<grid[0].length; j++){
            if(grid[i][j] == '1'){  → == '1'
                                       1
                dfsFill(grid, i, j);
                count ++;          ∵ it is
                                   a character
            }
        }
} return count;
```

Dfs fn

```
private void dfsFill(char[][] grid, int i, int j){
    if(i≥0 && j≥0 && i<grid.length && j<grid[0].
                                          length
                                      &&
                                      grid[i][j]
                                        == '1'){
        grid[i][j] = '0';
        dfsFill(grid, i+1, j);
        dfsFill(grid, i-1, j);
        dfsFill(grid, i, j+1);
    }   dfsFill(grid, i, j-1);
  }
}
```