# 200. Number of Islands 

By<br>Afif Mansib C. 6028129<br>Phoo Pwint Sone 6118147

## Question

Given an mxn 2d grid map of '1's (land) and '0's (water), return the number of islands.
An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.

## Example 1:

```
Input: grid = [
    ["1","1","1","1","0"],
    ["1","1", "0", "1", "0"],
    ["1","1","0","0","0"],
    ["0", "0", "0", "0", "0"]
]
Output: 1
```


## Example 2:

```
Input: grid = [
    ["1", "1", "0", "0", "0"],
    ["1", "1", "0", "0", "0"],
    ["0", "0", "1", "0", "0"],
    ["0", "0", "0", "1", "1"]
]
Output: 3
```

Input: grid = [

$$
\begin{aligned}
& {\left[\begin{array}{l}
{[" 1 ", ~ " 1 ", ~ " 1 ", ~ " 1 ", ~ " 0 "], ~} \\
{[\text { "1", "1", "0", "1", "0"], }} \\
{[\text { "1", "1", } 0 ", ~ " 0 ", ~ " 0 "], ~}
\end{array}\right.} \\
& \text { ["0", "0", "0", "0", "0"] }
\end{aligned}
$$

]

Input: grid = [
$[$ "1", "1", "0", "0", "0"],
$[" 1 ", ~ " 1 ", ~ " 0 ", ~ " 0 ", ~ " 0 "], ~$
["0", "0", "0", "1", "1"]
]

The output is 1.

Island is formed by connecting adjacent lands horizontally and vertically. Diagonal doesn't count.

- Once we start on index 0 , we search top, bottom, left and right.
- Once visited, we mark it as 0. (do DFS)

]

$$
\begin{aligned}
& \text { Input: grid = [ } \\
& \text { ["0", "1", "0", "0", "0"], } \\
& \text { ["1", "1", "0" , "0", "0"], } \\
& \text { ["0", "0", "1", "0", "0"], } \\
& \text { ["0", "0", "0", "1", "1"] } \\
& \text { ] }
\end{aligned}
$$

Input: grid = [
["0", "0", "0", "0", "0"]
["0", "0", "0", "0", "0"],
["0", "0", "0", "0", "0"],
["0", "0", "0", "0", "0"]
]
["1", "1", "0", "0", "0"], ["1", "1", "0", "0", "0"], ["0", "0", "1", "0", "0"],
islands = 0 ["0", "0", "0", "1", "1"]
for $r$ in range(len(grid)):
for $c$ in range(len(grid[r])):
if grid[r][c] == '1':
self.dfs(grid, r, c)
islands += 1
return islands

$$
\begin{aligned}
& \text { ["1", "1", "0", "0" , "0"], } \\
& \text { ["1", "1", "0", "0", "0"], } \\
& \text { ["0", "0", "1", "0", "0"], } \\
& \text { ["0", "0", "0", "1", "1"] }
\end{aligned}
$$

def dfs(self, grid, $r, c)$ :
$\operatorname{grid}[r][c]=$ '0'
Lst $=[(r-1, c),(r+1, c),(r, c-1),(r, c+1)]$
for row, col in lst:
if row $>=0$ and col $>=0$ and row < len(grid) $\backslash$ and col < len(grid[row]) and grid[row][col] == '1':
self.dfs(grid, row, col)

$$
\begin{aligned}
& \text { ["0", "0", "0" , "0", "0"], } \\
& \text { ["0", "0", "0", "0", "0"], } \\
& \text { ["0", "0", "1", "0", "0"], } \\
& \text { ["0", "0", "0", "1", "1"] }
\end{aligned}
$$

The input

```
grid = """11001
11010
00100
10000"""
grid = [list(line) for line in grid.splitlines()]
print(Solution().numIslands(grid))
```


## The output

```
/Users/phoopwintsone/PycharmProjec
```

/Users/phoopwintsone/PycharmProjec
5
Process finished with exit code 0

```

\section*{Submission on LeetCode}
\begin{tabular}{|l|l|l|l|l|}
\hline Time Submitted & Status & Runtime & Memory & Language \\
\hline \(03 / 10 / 202122: 15\) & Accepted & 128 ms & 15.4 MB & python3 \\
\hline
\end{tabular}

\section*{References}
https://leetcode.com/problems/number-of-islands/discuss/1103775/DFS-and-BFS-with-Easy-Explanation
https://www.youtube.com/watch?v=IFseCKh6Ndw\&t=705s
https://github.com/kamyu104/LeetCode-Solutions/blob/master/Python/number-of-island s.py


Thank You
```

