Problem 1020: Number of Enclaves

Problem Information

Difficulty: Medium
Acceptance Rate: 71.25%
Paid Only: No
Tags: Array, Depth-First Search, Breadth-First Search, Union Find, Matrix

Problem Description

You are given an `m x n` binary matrix `grid`, where `0` represents a sea cell and `1` represents a land cell.

A **move** consists of walking from one land cell to another adjacent (**4-directionally**) land cell or walking off the boundary of the `grid`.

Return _the number of land cells in_ `grid` _for which we cannot walk off the boundary of the grid in any number of**moves**_.

Example 1:

![](https://assets.leetcode.com/uploads/2021/02/18/enclaves1.jpg)

Input: grid = [[0,0,0,0],[1,0,1,0],[0,1,1,0],[0,0,0,0]] **Output:** 3 **Explanation:** There are three 1s that are enclosed by 0s, and one 1 that is not enclosed because its on the boundary.

Example 2:

![](https://assets.leetcode.com/uploads/2021/02/18/enclaves2.jpg)

Input: grid = [[0,1,1,0],[0,0,1,0],[0,0,1,0],[0,0,0,0]] **Output:** 0 **Explanation:** All 1s are either on the boundary or can reach the boundary.

Constraints:

* `m == grid.length` * `n == grid[i].length` * `1 <= m, n <= 500` * `grid[i][j]` is either `0` or `1`.

Code Snippets

C++:

```
class Solution {
public:
int numEnclaves(vector<vector<int>>& grid) {

}
};
```

Java:

```
class Solution {
public int numEnclaves(int[][] grid) {

}
}
```

Python3:

```
class Solution:
def numEnclaves(self, grid: List[List[int]]) -> int:
```