Number Of Enclaves

You are given an **n x m** 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.

Find 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:

```
Input:
grid[][] = {{0, 0, 0, 0},
            {1, 0, 1, 0},
            {0, 1, 1, 0},
            {0, 0, 0, 0}}
Output:
3
Explanation:
0 0 0 0
1 0 1 0
0 1 1 0
0 0 0 0
The highlighted cells represents the land cells.
```

Example 2:

```
Input:
grid[][] = {{0, 0, 0, 1},
            {0, 1, 1, 0},
            {0, 1, 1, 0},
            {0, 0, 0, 1},
```

```
            {0, 1, 1, 0}}
Output:
4
Explanation:
0 0 0 1
0 1 1 0
0 1 1 0
0 0 0 1
0 1 1 0
The highlighted cells represents the land cells.
```

Your Task:

You don't need to print or input anything. Complete the function **numberOfEnclaves()** which takes a 2D integer matrix **grid** as the input parameter and returns an integer, denoting the number of land cells.

Expected Time Complexity: O(n * m)

Expected Space Complexity: O(n * m)

Constraints:

- 1 <= n, m <= 500
- grid[i][j] == 0 or 1