Problem 419: Battleships in a Board

Problem Information

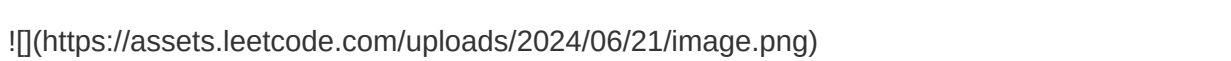
Difficulty: Medium
Acceptance Rate: 77.10%
Paid Only: No
Tags: Array, Depth-First Search, Matrix

Problem Description

Given an `m x n` matrix `board` where each cell is a battleship `'X'` or empty `'.'`, return _the number of the**battleships** on_ `board`.

Battleships can only be placed horizontally or vertically on `board`. In other words, they can only be made of the shape `1 x k` (`1` row, `k` columns) or `k x 1` (`k` rows, `1` column), where `k` can be of any size. At least one horizontal or vertical cell separates between two battleships (i.e., there are no adjacent battleships).

Example 1:



Input: board = [["X",".",".","X"],[".",".",".","X"],[".",".",".","X"]] **Output:** 2

Example 2:

Input: board = [["."]] **Output:** 0

Constraints:

* `m == board.length` * `n == board[i].length` * `1 <= m, n <= 200` * `board[i][j]` is either `'.'` or `'X'`.

Follow up: Could you do it in one-pass, using only `O(1)` extra memory and without modifying the values `board`?

Code Snippets

C++:

```
class Solution {
public:
int countBattleships(vector<vector<char>>& board) {

}
};
```

Java:

```
class Solution {
public int countBattleships(char[][] board) {

}
}
```

Python3:

```
class Solution:
def countBattleships(self, board: List[List[str]]) -> int:
```