Problem 531: Lonely Pixel I

Problem Information

Difficulty: Medium
Acceptance Rate: 62.55%
Paid Only: Yes
Tags: Array, Hash Table, Matrix

Problem Description

Given an `m x n` `picture` consisting of black `'B'` and white `'W'` pixels, return _the number of**black** lonely pixels_.

A black lonely pixel is a character `'B'` that located at a specific position where the same row and same column don't have **any other** black pixels.

Example 1:

![](https://assets.leetcode.com/uploads/2020/11/11/pixel1.jpg)

Input: picture = [["W","W","B"],["W","B","W"],["B","W","W"]] **Output:** 3 **Explanation:** All the three 'B's are black lonely pixels.

Example 2:

![](https://assets.leetcode.com/uploads/2020/11/11/pixel2.jpg)

Input: picture = [["B","B","B"],["B","B","W"],["B","B","B"]] **Output:** 0

Constraints:

* `m == picture.length` * `n == picture[i].length` * `1 <= m, n <= 500` * `picture[i][j]` is `'W'` or `'B'`.

Code Snippets

C++:

```
class Solution {
public:
int findLonelyPixel(vector<vector<char>>& picture) {

}
};
```

Java:

```
class Solution {
public int findLonelyPixel(char[][] picture) {

}
}
```

Python3:

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