Description | Accepted | Editorial | Submissions | Solutions

531. Lonely Pixel I

Premium

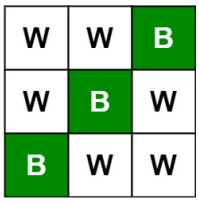
Solved

Medium | Topics | Companies

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

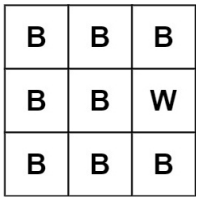
A black lonely pixel is a character `'B'` that located at a specific position where the same row and same column don't have **any other** black pixels.

Example 1:




```
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:




```
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'`.

Seen this question in a real interview before? 1/5

Yes | No

Accepted 42.8K | Submissions 68.8K | Acceptance Rate 62.2%

Topics

Companies

Similar Questions

Discussion (2)

430 | 2

Code

Python3 | Auto

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

        rows = {}
        cols = {}
        res = []

        count = 0

        for i in range(0, len(picture)):
            for j in range(0, len(picture[0])):
                if picture[i][j] == "B":
                    res.append([i, j])
                    if i not in rows:
                        rows[i] = 1
                    else:
                        rows[i] += 1
                    if j not in cols:
                        cols[j] = 1
                    else:
                        cols[j] += 1

        for (x, y) in res:
            if rows[x] == 1 and cols[y] == 1:
                count += 1

        return count
```

Saved

Ln 7, Col 1

Testcase | Test Result

Accepted Runtime: 48 ms

Case 1 | Case 2

Input

picture =

[["W","W","B"],["W","B","W"],["B","W","W"]]

Output