5968. Number of Laser Beams in a Bank

My Submissions (/contest/weekly-contest-274/problems/number-of-laser-beams-in-a-bank/submissions/)

Back to Contest (/contest/weekly-contest-274/)

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

Anti-theft security devices are activated inside a bank. You are given a **0-indexed** binary string array `bank` representing the floor plan of the bank, which is an `m x n` 2D matrix. `bank[i]` represents the i^{th} row, consisting of `'0'` s and `'1'` s. `'0'` means the cell is empty, while `'1'` means the cell has a security device.

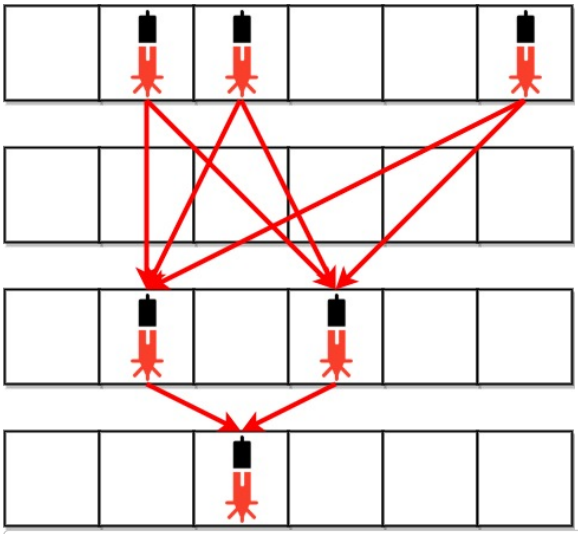
There is **one** laser beam between any **two** security devices **if both** conditions are met:

- The two devices are located on two **different rows**: r_1 and r_2, where $r_1 < r_2$.
- For **each** row i where $r_1 < i < r_2$, there are **no security devices** in the i^{th} row.

Laser beams are independent, i.e., one beam does not interfere nor join with another.

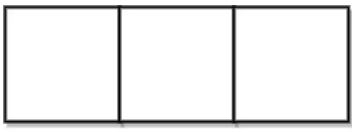
Return *the total number of laser beams in the bank*.

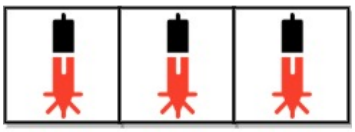
Example 1:

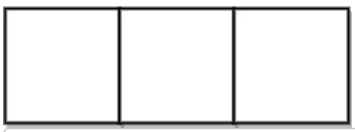


```
Input: bank = ["011001","000000","010100","001000"]
Output: 8
Explanation: Between each of the following device pairs, there is one beam. In total, there are 8 beams:
 * bank[0][1] -- bank[2][1]
 * bank[0][1] -- bank[2][3]
 * bank[0][2] -- bank[2][1]
 * bank[0][2] -- bank[2][3]
 * bank[0][5] -- bank[2][1]
 * bank[0][5] -- bank[2][3]
 * bank[2][1] -- bank[3][2]
 * bank[2][3] -- bank[3][2]
Note that there is no beam between any device on the 0th row with any on the 3rd row.
This is because the 2nd row contains security devices, which breaks the second condition.
```

Example 2:







```
Input: bank = ["000","111","000"]
Output: 0
Explanation: There does not exist two devices located on two different rows.
```

Constraints:

- `m == bank.length`
- `n == bank[i].length`
- `1 <= m, n <= 500`
- `bank[i][j]` is either `'0'` or `'1'`.

JavaScript

```
const numberOfBeams = (a) => {
    // let n = a.length, m = a[0].length;
    let d = a.map(s => {
        let one = 0;
        for (const c of s) {
            if (c == '1') one++;
        }
        return one;
    });
    // pr(d);
    let res = 0, pre;
    for (const x of d) {
        // pr(x, res)
        if (pre != undefined) {
            if (x > 0) {
                res += x * pre;
                pre = x;
            }
        } else {
            if (x > 0) {
                pre = x;
            }
        }
    }
    return res;
};
```

Custom Testcase Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/611156001/) More Details > (/submissions/detail/611156001/)

Share your acceptance!