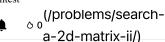
(/) Explore(/explore/) Problems(/problemset/all/) Interview Contest









2348. Number of Zero-Filled Subarrays

My Submissions (/contest/biweekly-contest-83/problems/number-of-zero-filled-subarrays/submissions/) Back to Contest (/contest/biweekly-contest-83/) Given an integer array $% \left(1\right) =\left(1\right) \left(1\right) =\left(1\right) \left(1\right) \left$ User Accepted: 10283 A subarray is a contiguous non-empty sequence of elements within an array. **User Tried:** 11095 **Total Accepted:** 10547 Example 1: **Total Submissions:** 19823 **Input:** nums = [1,3,0,0,2,0,0,4]Output: 6 Medium Difficulty: **Explanation:** There are 4 occurrences of [0] as a subarray. There are 2 occurrences of [0,0] as a subarray. There is no occurrence of a subarray with a size more than 2 filled with 0. Therefore, we return

Example 2:

```
Input: nums = [0,0,0,2,0,0]
Output: 9
Explanation:
There are 5 occurrences of [0] as a subarray.
There are 3 occurrences of [0,0] as a subarray.
There is 1 occurrence of [0,0,0] as a subarray.
There is no occurrence of a subarray with a size more than 3 filled with 0. Therefore, we return 9.
```

Example 3:

```
Input: nums = [2,10,2019]
Output: 0
Explanation: There is no subarray filled with 0. Therefore, we return 0.
```

Constraints:

- 1 <= nums.length <= 10^5
- $-10^9 \le nums[i] \le 10^9$

Discuss (https://leetcode.com/problems/number-of-zero-filled-subarrays/discuss)

```
JavaScript
                                                                                                                                \mathfrak{C}
                                                                                                                          Ø
    const cutMaxConsecutive = (as) \Rightarrow { let d = [], l = 0, n = as.length; for (let i = 0; i + 1 < n; i++) { if (as[i + 1] !=
    as[i] { d.push(as.slice(l, i + 1)); l = i + 1; } } d.push(as.slice(l)); return d; };
2
    const totSub = (n) \Rightarrow n * (n + 1) / 2;
3
4
    const zeroFilledSubarray = (a) => {
5
        let res = 0, d = cutMaxConsecutive(a);
6
        for (const e of d) {
7
           if (e[0] == 0) res += totSub(e.length);
8
        }
9
        return res;
10
   };
```

Custom Testcase

Use Example Testcases

Submission Result: Accepted (/submissions/detail/755185572/)

More Details ➤ (/submissions/detail/755185572/)

Share your acceptance!

2

Copyright © 2022 LeetCode

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)