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 `nums` , return *the number of **subarrays** filled with* `0` .

A **subarray** is a contiguous non-empty sequence of elements within an array.

User Accepted:	10283
User Tried:	11095
Total Accepted:	10547
Total Submissions:	19823
Difficulty:	Medium

Example 1:

```
Input: nums = [1,3,0,0,2,0,0,4]
Output: 6
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 retur
```

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 <= nums[i] <= 10^9`

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

JavaScript

```
const cutMaxConsecutive = (as) => { 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; };
const totSub = (n) => n * (n + 1) / 2;

const zeroFilledSubarray = (a) => {
    let res = 0, d = cutMaxConsecutive(a);
    for (const e of d) {
       if (e[0] == 0) res += totSub(e.length);
    }
    return res;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/755185572/) | More Details > (/submissions/detail/755185572/)

Share your acceptance!

2