2750. Ways to Split Array Into Good Subarrays

My Submissions (/contest/weekly-contest-351/problems/ways-to-split-array-into-good-subarrays/submissions/)

Back to Contest (/contest/weekly-contest-351/)

User Accepted:	7119
User Tried:	9575
Total Accepted:	7317
Total Submissions:	26281
Difficulty:	Medium

You are given a binary array `nums`.

A subarray of an array is **good** if it contains **exactly one** element with the value `1`.

Return *an integer denoting the number of ways to split the array* `nums` *into* ***good*** *subarrays*. As the number may be too large, return it **modulo** $10^9 + 7$.

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

Example 1:

```
Input: nums = [0,1,0,0,1]
Output: 3
Explanation: There are 3 ways to split nums into good subarrays:
- [0,1] [0,0,1]
- [0,1,0] [0,1]
- [0,1,0,0] [1]
```

Example 2:

```
Input: nums = [0,1,0]
Output: 1
Explanation: There is 1 way to split nums into good subarrays:
- [0,1,0]
```

Constraints:

- $1 <= nums.length <= 10^5$
- `0 <= nums[i] <= 1`

Discuss (https://leetcode.com/problems/ways-to-split-array-into-good-subarrays/discuss)

JavaScript

```javascript
const mod = 1e9 + 7;
const initialize2DArray = (n, m) => [...Array(n)].map(() => Array(m).fill(0));

const numberOfGoodSubarraySplits = (a) => {
    // f[i][state] : i为前i个元素，state:0,1代表该子串是否有1个1了
    let n = a.length, f = initialize2DArray(n, 2);
    f[0][0] = a[0] ^ 1;
    f[0][1] = a[0];
    for (let i = 1; i < n; i++) {
        if (a[i] == 0) {
            f[i][0] = (f[i - 1][0] + f[i - 1][1]) % mod;
            f[i][1] = f[i - 1][1];
        } else {
            f[i][0] = 0;
            f[i][1] = (f[i - 1][0] + f[i - 1][1]) % mod;
        }
    }
    return f[n - 1][1];
};
```

☐ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/979101615/)

More Details (/submissions/detail/979101615/)

Share your acceptance!

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

United States (/region)