6910. 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:	34
User Tried:	59
Total Accepted:	34
Total Submissions:	77
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`

JavaScript

```javascript
const mod = 1e9 + 7;

// 相邻的1中间找分割点
const numberOfGoodSubarraySplits = (a) => {
    let se = new Set(a);
    if (se.size == 1 && se.values().next().value == 0) return 0;
    let ia = [], res = 1;
    a.map((x, i) => {
        if (x == 1) ia.push(i);
    });
    for (let i = 1; i < ia.length; i++) {
        res *= ia[i] - ia[i - 1];
        res %= mod;
    }
    return res;
};
```

□ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/979020761/) More Details (/submissions/detail/979020761/)

Share your acceptance!

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

United States (/region)