(/) Explore Day 28 Problems(/problemset/all/) Interview New Contest (/contest/) 🚀August LeetCoding Challenge 2021🚀 Discuss(/discuss/) (/discuss/general-discussion/655704/) Store ☆ Premium (/subscribe?ref=nb_npl)

5857. Number of Unique Good Subsequences

My Submissions (/contest/weekly-contest-256/problems/number-of-unique-good-subsequences/submissions/)

Back to Contest (/contest/weekly-contest-256/)

User Accepted:	0
User Tried:	2
Total Accepted:	0
Total Submissions:	2
Difficulty:	Hard

You are given a binary string `binary` . A **subsequence** of `binary` is considered **good** if it is **not empty** and has **no leading zeros** (with the exception of `"0"`).

Find the number of **unique good subsequences** of `binary` .

- For example, if `binary = "001"` , then all the **good** subsequences are `["0", "0", "1"]` , so the **unique** good subsequences are `"0"` and `"1"` . Note that subsequences `"00"` , `"01"` , and `"001"` are not good because they have leading zeros.

Return *the number of **unique good subsequences** of* `binary` . Since the answer may be very large, return it **modulo** $10^9 + 7$.

A **subsequence** is a sequence that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements.

Example 1:

```
Input: binary = "001"
Output: 2
Explanation: The good subsequences of binary are ["0", "0", "1"].
The unique good subsequences are "0" and "1".
```

Example 2:

```
Input: binary = "11"
Output: 2
Explanation: The good subsequences of binary are ["1", "1", "11"].
The unique good subsequences are "1" and "11".
```

Example 3:

```
Input: binary = "101"
Output: 5
Explanation: The good subsequences of binary are ["1", "0", "1", "10", "11", "101"].
The unique good subsequences are "0", "1", "10", "11", and "101".
```

Constraints:

- $1 <= binary.length <= 10^5$
- `binary` consists of only `'0'` s and `'1'` s.

JavaScript

```
const mod = 1e9 + 7;
const numberOfUniqueGoodSubsequences = (binary) => {
    let idx0 = binary.indexOf('0'), idx1 = binary.indexOf('1');
    if (idx1 == -1) return 1;
    let s = binary.slice(idx1 + 1);
    let miss0 = miss1 = 1;
    for (const c of s) {
        if (c == '0') {
            miss1 += miss0;
            miss1 %= mod;
        } else {
            miss0 += miss1;
            miss0 %= mod;
        }
    }
    let res = miss0 + miss1 + mod + (idx0 != -1 ? 0 : -1);
    res %= mod;
    return res;
};
```

□ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/545942488/) More Details (/submissions/detail/545942488/)

Share your acceptance!

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

United States (/region)