

Problem 1987: Number of Unique Good Subsequences

Problem Information

Difficulty: Hard

Acceptance Rate: 52.48%

Paid Only: No

Tags: String, Dynamic Programming

Problem Description

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 <= 105 * `binary` consists of only `0`'s and `1`'s.

Code Snippets

C++:

```
class Solution {
public:
    int numberOfUniqueGoodSubsequences(string binary) {

    }
};
```

Java:

```
class Solution {
    public int numberOfUniqueGoodSubsequences(String binary) {

    }
}
```

Python3:

```
class Solution:
    def numberOfUniqueGoodSubsequences(self, binary: str) -> int:
```