Problem 2539: Count the Number of Good Subsequences

Problem Information

Difficulty: Medium
Acceptance Rate: 48.59%
Paid Only: Yes
Tags: Hash Table, Math, String, Combinatorics, Counting

Problem Description

A **subsequence** of a string is good if it is not empty and the frequency of each one of its characters is the same.

Given a string `s`, return _the number of good subsequences of_ `s`. Since the answer may be too large, return it modulo `109 + 7`.

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

Example 1:

Input: s = "aabb" **Output:** 11 **Explanation:** The total number of subsequences is 24. There are five subsequences which are not good: "**_aab_** b", "a _**abb**_ ", "**_a_** a _**bb**_ ", "_**aa**_ b** _b_** ", and the empty subsequence. Hence, the number of good subsequences is 24-5 = 11.

Example 2:

Input: s = "leet" **Output:** 12 **Explanation:** There are four subsequences which are not good: "**_l_ _ee_** t", "l _**eet**_ ", "**_leet_** ", and the empty subsequence. Hence, the number of good subsequences is 24-4 = 12.

Example 3:

Input: s = "abcd" **Output:** 15 **Explanation:** All of the non-empty subsequences are good subsequences. Hence, the number of good subsequences is 24-1 = 15.

Constraints:

* `1 <= s.length <= 104` * `s` consists of only lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
int countGoodSubsequences(string s) {

}
};
```

Java:

```
class Solution {
public int countGoodSubsequences(String s) {

}
}
```

Python3:

```
class Solution:
def countGoodSubsequences(self, s: str) -> int:
```