Problem 1525: Number of Good Ways to Split a String

Problem Information

Difficulty: Medium
Acceptance Rate: 68.40%
Paid Only: No
Tags: Hash Table, String, Dynamic Programming, Bit Manipulation

Problem Description

You are given a string `s`.

A split is called **good** if you can split `s` into two non-empty strings `sleft` and `sright` where their concatenation is equal to `s` (i.e., `sleft + sright = s`) and the number of distinct letters in `sleft` and `sright` is the same.

Return _the number of**good splits** you can make in `s`_.

Example 1:

Input: s = "aacaba" **Output:** 2 **Explanation:** There are 5 ways to split "aacaba" and 2 of them are good. ("a", "acaba") Left string and right string contains 1 and 3 different letters respectively. ("aa", "caba") Left string and right string contains 1 and 3 different letters respectively. ("aac", "aba") Left string and right string contains 2 and 2 different letters respectively (good split). ("aaca", "ba") Left string and right string contains 2 and 2 different letters respectively (good split). ("aacab", "a") Left string and right string contains 3 and 1 different letters respectively.

Example 2:

Input: s = "abcd" **Output:** 1 **Explanation:** Split the string as follows ("ab", "cd").

Constraints:

* `1 <= s.length <= 105` * `s` consists of only lowercase English letters.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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