Problem 2168: Unique Substrings With Equal Digit Frequency

Problem Information

Difficulty: Medium
Acceptance Rate: 64.18%
Paid Only: Yes
Tags: Hash Table, String, Rolling Hash, Counting, Hash Function

Problem Description

Given a digit string `s`, return _the number of**unique substrings** of _`s` _where every digit appears the same number of times._

Example 1:

Input: s = "1212" **Output:** 5 **Explanation:** The substrings that meet the requirements are "1", "2", "12", "21", "1212". Note that although the substring "12" appears twice, it is only counted once.

Example 2:

Input: s = "12321" **Output:** 9 **Explanation:** The substrings that meet the requirements are "1", "2", "3", "12", "23", "32", "21", "123", "321".

Constraints:

* `1 <= s.length <= 1000` * `s` consists of digits.

Code Snippets

C++:

```
class Solution {
public:
```

```
int equalDigitFrequency(string s) {

}
};
```

Java:

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

}
}
```

Python3:

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