Problem 1593: Split a String Into the Max Number of Unique Substrings

Problem Information

Difficulty: Medium
Acceptance Rate: 68.51%
Paid Only: No
Tags: Hash Table, String, Backtracking

Problem Description

Given a string `s`, return _the maximum number of unique substrings that the given string can be split into_.

You can split string `s` into any list of **non-empty substrings** , where the concatenation of the substrings forms the original string. However, you must split the substrings such that all of them are **unique**.

A **substring** is a contiguous sequence of characters within a string.

Example 1:

Input: s = "ababccc" **Output:** 5 **Explanation** : One way to split maximally is ['a', 'b', 'ab', 'c', 'cc']. Splitting like ['a', 'b', 'a', 'b', 'c', 'cc'] is not valid as you have 'a' and 'b' multiple times.

Example 2:

Input: s = "aba" **Output:** 2 **Explanation** : One way to split maximally is ['a', 'ba'].

Example 3:

Input: s = "aa" **Output:** 1 **Explanation** : It is impossible to split the string any further.

Constraints:

* `1 <= s.length <= 16`

* `s` contains only lower case English letters.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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