Problem 828: Count Unique Characters of All Substrings of a Given String

Problem Information

Difficulty: Hard
Acceptance Rate: 53.29%
Paid Only: No
Tags: Hash Table, String, Dynamic Programming

Problem Description

Let's define a function `countUniqueChars(s)` that returns the number of unique characters in `s`.

* For example, calling `countUniqueChars(s)` if `s = "LEETCODE"` then `"L"`, `"T"`, `"C"`, `"O"`, `"D"` are the unique characters since they appear only once in `s`, therefore `countUniqueChars(s) = 5`.

Given a string `s`, return the sum of `countUniqueChars(t)` where `t` is a substring of `s`. The test cases are generated such that the answer fits in a 32-bit integer.

Notice that some substrings can be repeated so in this case you have to count the repeated ones too.

Example 1:

Input: s = "ABC" **Output:** 10 **Explanation:** All possible substrings are: "A","B","C","AB","BC" and "ABC". Every substring is composed with only unique letters. Sum of lengths of all substring is 1 + 1 + 1 + 2 + 2 + 3 = 10

Example 2:

Input: s = "ABA" **Output:** 8 **Explanation:** The same as example 1, except countUniqueChars("ABA") = 1.

Example 3:

Input: s = "LEETCODE" **Output:** 92

Constraints:

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

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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