Problem 1638: Count Substrings That Differ by One Character

Problem Information

Difficulty: Medium
Acceptance Rate: 72.10%
Paid Only: No
Tags: Hash Table, String, Dynamic Programming, Enumeration

Problem Description

Given two strings `s` and `t`, find the number of ways you can choose a non- empty substring of `s` and replace a **single character** by a different character such that the resulting substring is a substring of `t`. In other words, find the number of substrings in `s` that differ from some substring in `t` by **exactly** one character.

For example, the underlined substrings in `"_compute_ r"` and `"_computa_ tion"` only differ by the `'e'`/`'a'`, so this is a valid way.

Return _the number of substrings that satisfy the condition above._

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

Example 1:

Input: s = "aba", t = "baba" **Output:** 6 **Explanation:** The following are the pairs of substrings from s and t that differ by exactly 1 character: ("_a_ ba", "_b_ aba") ("_a_ ba", "ba _b_ a") ("ab _a_ ", "_b_ aba") ("ab _a_ ", "ba _b_ a") ("a _b_ a", "b _a_ ba") ("a _b_ a", "bab _a_ ") The underlined portions are the substrings that are chosen from s and t.

Example 2:

Input: s = "ab", t = "bb" **Output:** 3 **Explanation:** The following are the pairs of substrings from s and t that differ by 1 character: ("_a_ b", "_b_ b") ("_a_ b", "b _b_ ") ("_ab_ ", "_bb_ ") The underlined portions are the substrings that are chosen from s and t.

Constraints:

* `1 <= s.length, t.length <= 100` * `s` and `t` consist of lowercase English letters only.

Code Snippets

C++:

```
class Solution {
public:
int countSubstrings(string s, string t) {

}
};
```

Java:

```
class Solution {
public int countSubstrings(String s, String t) {

}
}
```

Python3:

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