Problem 3298: Count Substrings That Can Be Rearranged to Contain a String II

Problem Information

Difficulty: Hard
Acceptance Rate: 55.66%
Paid Only: No
Tags: Hash Table, String, Sliding Window

Problem Description

You are given two strings `word1` and `word2`.

A string `x` is called **valid** if `x` can be rearranged to have `word2` as a prefix.

Return the total number of **valid** substrings of `word1`.

Note that the memory limits in this problem are **smaller** than usual, so you **must** implement a solution with a _linear_ runtime complexity.

Example 1:

Input: word1 = "bcca", word2 = "abc"

Output: 1

Explanation:

The only valid substring is `"bcca"` which can be rearranged to `"abcc"` having `"abc"` as a prefix.

Example 2:

Input: word1 = "abcabc", word2 = "abc"

Output: 10

Explanation:

All the substrings except substrings of size 1 and size 2 are valid.

Example 3:

Input: word1 = "abcabc", word2 = "aaabc"

Output: 0

Constraints:

* `1 <= word1.length <= 106` * `1 <= word2.length <= 104` * `word1` and `word2` consist only of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
long long validSubstringCount(string word1, string word2) {

}
};
```

Java:

```
class Solution {
public long validSubstringCount(String word1, String word2) {

}
}
```

Python3:

```
class Solution:
def validSubstringCount(self, word1: str, word2: str) -> int:
```