Problem 3365: Rearrange K Substrings to Form Target String

Problem Information

Difficulty: Medium
Acceptance Rate: 56.48%
Paid Only: No
Tags: Hash Table, String, Sorting

Problem Description

You are given two strings `s` and `t`, both of which are anagrams of each other, and an integer `k`.

Your task is to determine whether it is possible to split the string `s` into `k` equal-sized substrings, rearrange the substrings, and concatenate them in _any order_ to create a new string that matches the given string `t`.

Return `true` if this is possible, otherwise, return `false`.

An **anagram** is a word or phrase formed by rearranging the letters of a different word or phrase, using all the original letters exactly once.

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

Example 1:

Input: s = "abcd", t = "cdab", k = 2

Output: true

Explanation:

* Split `s` into 2 substrings of length 2: `["ab", "cd"]`. * Rearranging these substrings as `["cd", "ab"]`, and then concatenating them results in `"cdab"`, which matches `t`.

Example 2:

Input: s = "aabbcc", t = "bbaacc", k = 3

Output: true

Explanation:

* Split `s` into 3 substrings of length 2: `["aa", "bb", "cc"]`. * Rearranging these substrings as `["bb", "aa", "cc"]`, and then concatenating them results in `"bbaacc"`, which matches `t`.

Example 3:

Input: s = "aabbcc", t = "bbaacc", k = 2

Output: false

Explanation:

* Split `s` into 2 substrings of length 3: `["aab", "bcc"]`. * These substrings cannot be rearranged to form `t = "bbaacc"`, so the output is `false`.

Constraints:

* `1 <= s.length == t.length <= 2 * 105` * `1 <= k <= s.length` * `s.length` is divisible by `k`. * `s` and `t` consist only of lowercase English letters. * The input is generated such that `s` and `t` are anagrams of each other.

Code Snippets

C++:

```
class Solution {
public:
bool isPossibleToRearrange(string s, string t, int k) {

}
};
```

Java:

```
class Solution {
public boolean isPossibleToRearrange(String s, String t, int k) {

}
}
```

Python3:

```
class Solution:
def isPossibleToRearrange(self, s: str, t: str, k: int) -> bool:
```