Problem 2060: Check if an Original String Exists Given Two Encoded Strings

Problem Information

Difficulty: Hard
Acceptance Rate: 43.42%
Paid Only: No
Tags: String, Dynamic Programming

Problem Description

An original string, consisting of lowercase English letters, can be encoded by the following steps:

* Arbitrarily **split** it into a **sequence** of some number of **non-empty** substrings. * Arbitrarily choose some elements (possibly none) of the sequence, and **replace** each with **its length** (as a numeric string). * **Concatenate** the sequence as the encoded string.

For example, **one way** to encode an original string `"abcdefghijklmnop"` might be:

* Split it as a sequence: `["ab", "cdefghijklmn", "o", "p"]`. * Choose the second and third elements to be replaced by their lengths, respectively. The sequence becomes `["ab", "12", "1", "p"]`. * Concatenate the elements of the sequence to get the encoded string: `"ab121p"`.

Given two encoded strings `s1` and `s2`, consisting of lowercase English letters and digits `1-9` (inclusive), return `true` _if there exists an original string that could be encoded as**both** _`s1` _and_`s2` _. Otherwise, return_`false`.

Note : The test cases are generated such that the number of consecutive digits in `s1` and `s2` does not exceed `3`.

Example 1:

Input: s1 = "internationalization", s2 = "i18n" **Output:** true **Explanation:** It is possible that "internationalization" was the original string. - "internationalization" -> Split: ["internationalization"] -> Do not replace any element -> Concatenate: "internationalization",

which is s1. - "internationalization" -> Split: ["i", "nternationalizatio", "n"] -> Replace: ["i", "18", "n"] -> Concatenate: "i18n", which is s2

Example 2:

Input: s1 = "l123e", s2 = "44" **Output:** true **Explanation:** It is possible that "leetcode" was the original string. - "leetcode" -> Split: ["l", "e", "et", "cod", "e"] -> Replace: ["l", "1", "2", "3", "e"] -> Concatenate: "l123e", which is s1. - "leetcode" -> Split: ["leet", "code"] -> Replace: ["4", "4"] -> Concatenate: "44", which is s2.

Example 3:

Input: s1 = "a5b", s2 = "c5b" **Output:** false **Explanation:** It is impossible. - The original string encoded as s1 must start with the letter 'a'. - The original string encoded as s2 must start with the letter 'c'.

Constraints:

* `1 <= s1.length, s2.length <= 40` * `s1` and `s2` consist of digits `1-9` (inclusive), and lowercase English letters only. * The number of consecutive digits in `s1` and `s2` does not exceed `3`.

Code Snippets

C++:

```
class Solution {
public:
bool possiblyEquals(string s1, string s2) {

}
};
```

Java:

```
class Solution {
public boolean possiblyEquals(String s1, String s2) {

}
}
```

Python3:

```
class Solution:
def possiblyEquals(self, s1: str, s2: str) -> bool:
```