Problem 1585: Check If String Is Transformable With Substring Sort Operations

Problem Information

Difficulty: Hard
Acceptance Rate: 50.93%
Paid Only: No
Tags: String, Greedy, Sorting

Problem Description

Given two strings `s` and `t`, transform string `s` into string `t` using the following operation any number of times:

* Choose a **non-empty** substring in `s` and sort it in place so the characters are in **ascending order**. * For example, applying the operation on the underlined substring in `"1 _4234_ "` results in `"1 _2344_ "`.

Return `true` if _it is possible to transform`s` into `t`_. Otherwise, return `false`.

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

Example 1:

Input: s = "84532", t = "34852" **Output:** true **Explanation:** You can transform s into t using the following sort operations: "84 _53_ 2" (from index 2 to 3) -> "84 _35_ 2" "_843_ 52" (from index 0 to 2) -> "_348_ 52"

Example 2:

Input: s = "34521", t = "23415" **Output:** true **Explanation:** You can transform s into t using the following sort operations: "_3452_ 1" -> "_2345_ 1" "234 _51_ " -> "234 _15_ "

Example 3:

Input: s = "12345", t = "12435" **Output:** false

Constraints:

* `s.length == t.length` * `1 <= s.length <= 105` * `s` and `t` consist of only digits.

Code Snippets

C++:

```
class Solution {
public:
bool isTransformable(string s, string t) {

}
};
```

Java:

```
class Solution {
public boolean isTransformable(String s, String t) {

}
}
```

Python3:

```
class Solution:
def isTransformable(self, s: str, t: str) -> bool:
```