Problem 2486: Append Characters to String to Make Subsequence

Problem Information

Difficulty: Medium
Acceptance Rate: 72.96%
Paid Only: No
Tags: Two Pointers, String, Greedy

Problem Description

You are given two strings `s` and `t` consisting of only lowercase English letters.

Return _the minimum number of characters that need to be appended to the end of_`s` _so that_`t` _becomes a**subsequence** of _`s`.

A **subsequence** is a string that can be derived from another string by deleting some or no characters without changing the order of the remaining characters.

Example 1:

Input: s = "coaching", t = "coding" **Output:** 4 **Explanation:** Append the characters "ding" to the end of s so that s = "coachingding". Now, t is a subsequence of s ("_**co**_ aching _**ding**_ "). It can be shown that appending any 3 characters to the end of s will never make t a subsequence.

Example 2:

Input: s = "abcde", t = "a" **Output:** 0 **Explanation:** t is already a subsequence of s ("_**a**_ bcde").

Example 3:

Input: s = "z", t = "abcde" **Output:** 5 **Explanation:** Append the characters "abcde" to the end of s so that s = "zabcde". Now, t is a subsequence of s ("z _**abcde**_ "). It can be shown that appending any 4 characters to the end of s will never make t a subsequence.

Constraints:

* `1 <= s.length, t.length <= 105` * `s` and `t` consist only of lowercase English letters.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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