Problem 1143: Longest Common Subsequence

Problem Information

Difficulty: Medium
Acceptance Rate: 58.63%
Paid Only: No
Tags: String, Dynamic Programming

Problem Description

Given two strings `text1` and `text2`, return _the length of their longest**common subsequence**. _If there is no **common subsequence** , return `0`.

A **subsequence** of a string is a new string generated from the original string with some characters (can be none) deleted without changing the relative order of the remaining characters.

* For example, `"ace"` is a subsequence of `"abcde"`.

A **common subsequence** of two strings is a subsequence that is common to both strings.

Example 1:

Input: text1 = "abcde", text2 = "ace" **Output:** 3 **Explanation:** The longest common subsequence is "ace" and its length is 3.

Example 2:

Input: text1 = "abc", text2 = "abc" **Output:** 3 **Explanation:** The longest common subsequence is "abc" and its length is 3.

Example 3:

Input: text1 = "abc", text2 = "def" **Output:** 0 **Explanation:** There is no such common subsequence, so the result is 0.

Constraints:

* `1 <= text1.length, text2.length <= 1000` * `text1` and `text2` consist of only lowercase English characters.

Code Snippets

C++:

```
class Solution {
public:
int longestCommonSubsequence(string text1, string text2) {

}
};
```

Java:

```
class Solution {
public int longestCommonSubsequence(String text1, String text2) {

}
}
```

Python3:

```
class Solution:
def longestCommonSubsequence(self, text1: str, text2: str) -> int:
```