Our Solution(s) | Run Code

Solution 1   Solution 2   Solution 3   Solution 4

```csharp
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

using System;
using System.Collections.Generic;

public class Program {
  // O(nm) time | O(nm) space
  public static List<char> LongestCommonSubsequence(string str1, stri
    int[,] lengths = new int[str2.Length + 1,str1.Length + 1];
    for (int i = 1; i < str2.Length + 1; i++) {
      for (int j = 1; j < str1.Length + 1; j++) {
        if (str2[i - 1] == str1[j - 1]) {
          lengths[i,j] = lengths[i - 1,j - 1] + 1;
        } else {
          lengths[i,j] = Math.Max(lengths[i - 1,j], lengths[i,j - 1]);
        }
      }
    }
    return buildSequence(lengths, str1);
  }

  public static List<char> buildSequence(int[,] lengths, string str) {
    List<char> sequence = new List<char>();
    int i = lengths.GetLength(0) - 1;
    int j = lengths.GetLength(1) - 1;
    while (i != 0 && j != 0) {
      if (lengths[i,j] == lengths[i - 1,j]) {
        i--;
      } else if (lengths[i,j] == lengths[i,j - 1]) {
        j--;
      } else {
        sequence.Insert(0, str[j - 1]);
        i--;
```

Your Solutions | Run Code

Solution 1   Solution 2   Solution 3

```csharp
using System.Collections.Generic;

public class Program {
  public static List<char> LongestCommonSubsequence(string str1, strin
    // Write your code here.
    return null;
  }
}
```

Our Tests

Custom Output | Submit Code

Run or submit code when you're ready.