AlgoExpert Quad Layout C# 12px Sublime Monokai 00:00:00

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1 Solution 2 Solution 3 Solution 4

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

using System.Collections.Generic;

public class Program {
  // O(nm*min(n, m)) time | O((min(n, m))^2) space
  public static List<char> LongestCommonSubsequence(string str1, string str2) {
    string small = str1.Length < str2.Length ? str1 : str2;
    string big = str1.Length >= str2.Length ? str1 : str2;
    List<List<char> > evenLcs = new List<List<char> >();
    List<List<char> > oddLcs = new List<List<char> >();
    for (int i = 0; i < small.Length + 1; i++) {
      evenLcs.Add(new List<char>());
    }
    for (int i = 0; i < small.Length + 1; i++) {
      oddLcs.Add(new List<char>());
    }
    for (int i = 1; i < big.Length + 1; i++) {
      List<List<char> > currentLcs;
      List<List<char> > previousLcs;
      if (i % 2 == 1) {
        currentLcs = oddLcs;
        previousLcs = evenLcs;
      } else {
        currentLcs = evenLcs;
        previousLcs = oddLcs;
      }
      for (int j = 1; j < small.Length + 1; j++) {
        if (big[i - 1] == small[j - 1]) {
          List<char> copy = new List<char>(previousLcs[j - 1]);
          currentLcs[j] =  copy;
          currentLcs[j].Add(big[i - 1]);
        } else {
          if (previousLcs[j].Count > currentLcs[j - 1].Count) {
            currentLcs[j] =  previousLcs[j];
          } else {
            currentLcs[j] =  currentLcs[j - 1];
          }
        }
      }
    }
    return big.Length % 2 == 0 ? evenLcs[small.Length] : oddLcs[small.Length];
  }
}
```