Our Solution(s) Run Code

Solution 1 Solution 2 Solution 3 Solution 4

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

import java.util.*;

class Program {
  // O(nm) time | O(nm) space
  public static List<Character> longestCommonSubsequence(String str1,
    int[][][] lcs = new int[str2.length() + 1][str1.length() + 1][];
    for (int i = 0; i < str2.length() + 1; i++) {
      for (int j = 0; j < str1.length() + 1; j++) {
        lcs[i][j] = new int[] {0, 0, 0, 0};
      }
    }
    for (int i = 1; i < str2.length() + 1; i++) {
      for (int j = 1; j < str1.length() + 1; j++) {
        if (str2.charAt(i - 1) == str1.charAt(j - 1)) {
          int[] newEntry = {(int) str2.charAt(i - 1), lcs[i - 1][j - 1
          lcs[i][j] = newEntry;
        } else {
          if (lcs[i - 1][j][1] > lcs[i][j - 1][1]) {
            int[] newEntry = {-1, lcs[i - 1][j][1], i - 1, j};
            lcs[i][j] = newEntry;
          } else {
            int[] newEntry = {-1, lcs[i][j - 1][1], i, j - 1};
            lcs[i][j] = newEntry;
          }
        }
      }
    }
    return buildSequence(lcs);
  }

  public static List<Character> buildSequence(int[][][] lcs) {
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```java
import java.util.*;

class Program {
  public static List<Character> longestCommonSubsequence(String str1,
    // Write your code here.
    return null;
  }
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.