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

using System.Collections.Generic;

public class Program {
  // O(nm) time | O(nm) space
  public static List<char> LongestCommonSubsequence(string str1, string str2) {
    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[i - 1] == str1[j - 1]) {
          int[] newEntry =
          {(int)str2[i - 1], lcs[i - 1,j - 1][1] + 1, 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<char> buildSequence(int[,][] lcs) {
    List<char> sequence = new List<char>();
    int i = lcs.GetLength(0) - 1;
    int j = lcs.GetLength(1) - 1;
    while (i != 0 && j != 0) {
      int[] currentEntry = lcs[i,j];
      if (currentEntry[0] != -1) {
        sequence.Insert(0, (char)currentEntry[0]);
      }
      i = currentEntry[2];
      j = currentEntry[3];
    }
    return sequence;
  }
}

```