

PromptScratchpadOur Solution(s)Video Explanation

Run Code

Solution 1Solution 2Solution 3Solution 4

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 import java.util.*;
4
5 class Program {
6     // O(nm) time | O(nm) space
7     public static List<Character> longestCommonSubsequence(String str1, String str2) {
8         int[][] lengths = new int[str2.length() + 1][str1.length() + 1];
9         for (int i = 1; i < str2.length() + 1; i++) {
10             for (int j = 1; j < str1.length() + 1; j++) {
11                 if (str2.charAt(i - 1) == str1.charAt(j - 1)) {
12                     lengths[i][j] = lengths[i - 1][j - 1] + 1;
13                 } else {
14                     lengths[i][j] = Math.max(lengths[i - 1][j], lengths[i][j - 1]);
15                 }
16             }
17         }
18         return buildSequence(lengths, str1);
19     }
20
21     public static List<Character> buildSequence(int[][] lengths, String str) {
22         List<Character> sequence = new ArrayList<Character>();
23         int i = lengths.length - 1;
24         int j = lengths[0].length - 1;
25         while (i != 0 && j != 0) {
26             if (lengths[i][j] == lengths[i - 1][j]) {
27                 i--;
28             } else if (lengths[i][j] == lengths[i][j - 1]) {
29                 j--;
30             } else {
31                 sequence.add(0, str.charAt(j - 1));
32                 i--;
33                 j--;
34             }
35         }
36         return sequence;
37     }
38 }
39
```