

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*min(n, m)) time | O(nm*min(n, m)) space
7     public static List<Character> longestCommonSubsequence(String str1, String str2) {
8         List<List<List<Character>>> lcs = new ArrayList<List<List<Character>>>();
9         for (int i = 0; i < str2.length() + 1; i++) {
10             lcs.add(new ArrayList<List<Character>>());
11             for (int j = 0; j < str1.length() + 1; j++) {
12                 lcs.get(i).add(new ArrayList<Character>());
13             }
14         }
15         for (int i = 1; i < str2.length() + 1; i++) {
16             for (int j = 1; j < str1.length() + 1; j++) {
17                 if (str2.charAt(i - 1) == str1.charAt(j - 1)) {
18                     List<Character> copy = new ArrayList<Character>(lcs.get(i - 1).get(j - 1));
19                     lcs.get(i).set(j, copy);
20                     lcs.get(i).get(j).add(str2.charAt(i - 1));
21                 } else {
22                     if (lcs.get(i - 1).get(j).size() > lcs.get(i).get(j - 1).size()) {
23                         lcs.get(i).set(j, lcs.get(i - 1).get(j));
24                     } else {
25                         lcs.get(i).set(j, lcs.get(i).get(j - 1));
26                     }
27                 }
28             }
29         }
30         return lcs.get(str2.length()).get(str1.length());
31     }
32 }
33
```