

PromptScratchpadOur Solution(s)Video Explanation

Run Code

Solution 1Solution 2Solution 3Solution 4

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 // O(nm) time | O(nm) space
6 func LongestCommonSubsequence(str1 string, str2 string) string {
7     lengths := make([][]int, len(str2)+1)
8     for i := range lengths {
9         lengths[i] = make([]int, len(str1)+1)
10    }
11    for i := 1; i < len(str2)+1; i++ {
12        for j := 1; j < len(str1)+1; j++ {
13            if str2[i-1] == str1[j-1] {
14                lengths[i][j] = lengths[i-1][j-1] + 1
15            } else {
16                lengths[i][j] = max(lengths[i-1][j], lengths[i][j-1])
17            }
18        }
19    }
20
21    return buildSequence(lengths, str1)
22 }
23
24 func buildSequence(lengths [][]int, str1 string) string {
25     sequence := make([]byte, 0)
26     i := len(lengths) - 1
27     j := len(lengths[0]) - 1
28     for i != 0 && j != 0 {
29         if lengths[i][j] == lengths[i-1][j] {
30             i -= 1
31         } else if lengths[i][j] == lengths[i][j-1] {
32             j -= 1
33         } else {
34             sequence = append(sequence, str1[j-1])
35             i -= 1
36             j -= 1
37         }
38     }
39     return string(reverse(sequence))
40 }
41
42 func reverse(arr []byte) []byte {
43     for i, j := 0, len(arr)-1; i < j; i, j = i+1, j-1 {
44         arr[i], arr[j] = arr[j], arr[i]
45     }
46     return arr
47 }
48
49 func max(i, j int) int {
50     if i > j {
51         return i
52     }
53     return j
54 }
55
```