

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 type entry struct {
6     letter byte
7     length int
8     previ int
9     prevj int
10 }
11
12 // O(nm) time | O(nm) space
13 func LongestCommonSubsequence(str1 string, str2 string) string {
14     lcs := make([][]entry, len(str2)+1)
15     for i := range lcs {
16         lcs[i] = make([]entry, len(str1)+1)
17         for j := range lcs[i] {
18             lcs[i][j].letter = 0
19             lcs[i][j].length = 0
20             lcs[i][j].previ = -1
21             lcs[i][j].prevj = -1
22         }
23     }
24
25     for i := 1; i < len(str2)+1; i++ {
26         for j := 1; j < len(str1)+1; j++ {
27             if str2[i-1] == str1[j-1] {
28                 lcs[i][j] = entry{str2[i-1], lcs[i-1][j-1].length + 1, i - 1, j - 1}
29             } else {
30                 if lcs[i-1][j].length > lcs[i][j-1].length {
31                     lcs[i][j] = entry{0, lcs[i-1][j].length, i - 1, j}
32                 } else {
33                     lcs[i][j] = entry{0, lcs[i][j-1].length, i, j - 1}
34                 }
35             }
36         }
37     }
38     return buildSequence(lcs)
39 }
40
41 func buildSequence(lcs [][]entry) string {
42     sequence := make([]byte, 0)
43     i := len(lcs) - 1
44     j := len(lcs[0]) - 1
45     for i != 0 && j != 0 {
46         current := lcs[i][j]
47         if current.letter != 0 {
48             sequence = append(sequence, current.letter)
49         }
50         i = current.previ
51         j = current.prevj
52     }
53     return string(reverse(sequence))
54 }
55
56 func reverse(arr []byte) []byte {
57     for i, j := 0, len(arr)-1; i < j; i, j = i+1, j-1 {
58         arr[i], arr[j] = arr[j], arr[i]
59     }
60     return arr
61 }
62
```