

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*min(n, m)) time | O((min(n, m))^2) space
6 func LongestCommonSubsequence(s1 string, s2 string) string {
7     small, big := s1, s2
8     if len(s1) > len(s2) {
9         small, big = big, small
10    }
11    evenlcs := make([]string, len(small)+1)
12    oddlcs := make([]string, len(small)+1)
13    currentlcs, previouslcs := evenlcs, oddlcs
14    for i := 1; i < len(big)+1; i++ {
15        if i%2 == 1 {
16            currentlcs, previouslcs = oddlcs, evenlcs
17        } else {
18            currentlcs, previouslcs = evenlcs, oddlcs
19        }
20        for j := 1; j < len(small)+1; j++ {
21            if big[i-1] == small[j-1] {
22                currentlcs[j] = string(append([]byte(previouslcs[j-1]), big[i-1]))
23            } else {
24                if len(previouslcs[j]) > len(currentlcs[j-1]) {
25                    currentlcs[j] = previouslcs[j]
26                } else {
27                    currentlcs[j] = currentlcs[j-1]
28                }
29            }
30        }
31    }
32
33    if len(big)%2 == 0 {
34        return evenlcs[len(small)]
35    } else {
36        return oddlcs[len(small)]
37    }
38 }
39
```