

PromptScratchpadOur Solution(s)Video Explanation

Run Code

Solution 1Solution 2Solution 3Solution 4

```
1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 # O(nm) time | O(nm) space
4 def longestCommonSubsequence(str1, str2):
5     lcs = [[None, 0, None, None] for x in range(len(str1) + 1) for y in range(len(str2) + 1)]
6     for i in range(1, len(str2) + 1):
7         for j in range(1, len(str1) + 1):
8             if str2[i - 1] == str1[j - 1]:
9                 lcs[i][j] = [str2[i - 1], lcs[i - 1][j - 1][1] + 1, i - 1, j - 1]
10            else:
11                if lcs[i - 1][j][1] > lcs[i][j - 1][1]:
12                    lcs[i][j] = [None, lcs[i - 1][j][1], i - 1, j]
13                else:
14                    lcs[i][j] = [None, lcs[i][j - 1][1], i, j - 1]
15    return buildSequence(lcs)
16
17
18 def buildSequence(lcs):
19     sequence = []
20     i = len(lcs) - 1
21     j = len(lcs[0]) - 1
22     while i != 0 and j != 0:
23         currentEntry = lcs[i][j]
24         if currentEntry[0] is not None:
25             sequence.append(currentEntry[0])
26         i = currentEntry[2]
27         j = currentEntry[3]
28    return list(reversed(sequence))
29
```