

PromptScratchpadOur Solution(s)Video Explanation

Run Code

Solution 1Solution 2Solution 3Solution 4

```
1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 # O(nm*min(n, m)) time | O(nm*min(n, m)) space
4 def longestCommonSubsequence(str1, str2):
5     lcs = [[[] 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] = lcs[i - 1][j - 1] + [str2[i - 1]]
10            else:
11                lcs[i][j] = max(lcs[i - 1][j], lcs[i][j - 1], key=len)
12    return lcs[-1][-1]
13
```