

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1Solution 2Solution 3Solution 4

```
1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 # O(nm*min(n, m)) time | O((min(n, m))^2) space
4 def longestCommonSubsequence(str1, str2):
5     small = str1 if len(str1) < len(str2) else str2
6     big = str1 if len(str1) >= len(str2) else str2
7     evenLcs = [[] for x in range(len(small) + 1)]
8     oddLcs = [[] for x in range(len(small) + 1)]
9     for i in range(1, len(big) + 1):
10         if i % 2 == 1:
11             currentLcs = oddLcs
12             previousLcs = evenLcs
13         else:
14             currentLcs = evenLcs
15             previousLcs = oddLcs
16         for j in range(1, len(small) + 1):
17             if big[i - 1] == small[j - 1]:
18                 currentLcs[j] = previousLcs[j - 1] + [big[i - 1]]
19             else:
20                 currentLcs[j] = max(previousLcs[j], currentLcs[j - 1],
21                                     return evenLcs[-1] if len(big) % 2 == 0 else oddLcs[-1]
```

Solution 1Solution 2Solution 3

```
1 def longestCommonSubsequence(str1, str2):
2     # Write your code here.
3     pass
4
```

Our Tests

Custom Output

Submit Code

```
1 def longestCommonSubsequence(str1, str2):
2     # Write your code here.
3
4     return longestCommonSubsequence(str1, str2)
```

```
1 def longestCommonSubsequence(str1, str2):
2     # Write your code here.
3
4     return longestCommonSubsequence(str1, str2)
```

Run or submit code when you're ready.