

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(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)
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
```

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     # Test Case 1: str1 = "abcde", str2 = "ace"
5     # Expected Output: ["a", "c", "e"]
6
7     # Test Case 2: str1 = "abcde", str2 = "bce"
8     # Expected Output: ["b", "c", "e"]
9
10    # Test Case 3: str1 = "abcde", str2 = "aceb"
11    # Expected Output: ["a", "c", "e"]
12
13    # Test Case 4: str1 = "abcde", str2 = "aceb"
14    # Expected Output: ["a", "c", "e"]
15
```

