

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

Solution 3

Solution 4

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 // O(nm*min(n, m)) time | O((min(n, m))^2) space
4 function longestCommonSubsequence(str1, str2) {
5   const small = str1.length < str2.length ? str1 : str2;
6   const big = str1.length >= str2.length ? str1 : str2;
7   const evenLcs = new Array(small.length + 1).fill([]);
8   const oddLcs = new Array(small.length + 1).fill([]);
9   for (let i = 1; i < big.length + 1; i++) {
10     let currentLcs, previousLcs;
11     if (i % 2 === 1) {
12       currentLcs = oddLcs;
13       previousLcs = evenLcs;
14     } else {
15       currentLcs = evenLcs;
16       previousLcs = oddLcs;
17     }
18     for (let j = 1; j < small.length + 1; j++) {
19       if (big[i - 1] === small[j - 1]) {
20         currentLcs[j] = previousLcs[j - 1].concat(big[i - 1]);
21       } else {
22         currentLcs[j] = previousLcs[j].length > currentLcs[j - 1].length ? previousLcs[j] : currentLcs[j - 1];
23       }
24     }
25   }
26   return big.length % 2 === 0 ? evenLcs[small.length] : oddLcs[small.length];
27 }
28
29 exports.longestCommonSubsequence = longestCommonSubsequence;
30
```

Solution 1

Solution 2

Solution 3

```
1 function longestCommonSubsequence(str1, str2) {
2   // Write your code here.
3 }
4
5 // Do not edit the line below.
6 exports.longestCommonSubsequence = longestCommonSubsequence;
7
```

Our Tests

Custom Output

Submit Code

```
1 // Test Case 1: str1 = "abcde", str2 = "ace"
2 // Expected: "ace"
3
4 // Test Case 2: str1 = "abcde", str2 = "bce"
5 // Expected: "bce"
6
7 // Test Case 3: str1 = "abcde", str2 = "aceb"
8 // Expected: "ace"
9
10 // Test Case 4: str1 = "abcde", str2 = "aceb"
11 // Expected: "ace"
12
13 // Test Case 5: str1 = "abcde", str2 = "aceb"
14 // Expected: "ace"
15
16 // Test Case 6: str1 = "abcde", str2 = "aceb"
17 // Expected: "ace"
18
19 // Test Case 7: str1 = "abcde", str2 = "aceb"
20 // Expected: "ace"
21
22 // Test Case 8: str1 = "abcde", str2 = "aceb"
23 // Expected: "ace"
24
25 // Test Case 9: str1 = "abcde", str2 = "aceb"
26 // Expected: "ace"
27
28 // Test Case 10: str1 = "abcde", str2 = "aceb"
29 // Expected: "ace"
30
```

```
10 # Test Case 40 - Success (2/2)
11 # Test Case 40: Input: [1, 2, 3, 4, 5], [10, 20, 30, 40, 50]
12 # Expected Output: [10, 20, 30, 40, 50]
13 # Test Case 40: Success (2/2)
14 # Test Case 41 - Success (2/2)
15 # Test Case 41: Input: [1, 2, 3, 4, 5], [10, 20, 30, 40, 50]
16 # Expected Output: [10, 20, 30, 40, 50]
17 # Test Case 41: Success (2/2)
18 # Test Case 42 - Success (2/2)
19 # Test Case 42: Input: [1, 2, 3, 4, 5], [10, 20, 30, 40, 50]
20 # Expected Output: [10, 20, 30, 40, 50]
21 # Test Case 42: Success (2/2)
22 # Test Case 43 - Success (2/2)
23 # Test Case 43: Input: [1, 2, 3, 4, 5], [10, 20, 30, 40, 50]
24 # Expected Output: [10, 20, 30, 40, 50]
25 # Test Case 43: Success (2/2)
```

Run or submit code when you're ready.