

Our Solution(s)

Run Code

Solution 1Solution 2Solution 3Solution 4

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 using namespace std;
5
6 vector<char> buildSequence(vector<vector<vector<int>>> lcs);
7
8 // O(nm) time | O(nm) space
9 vector<char> longestCommonSubsequence(string str1, string str2) {
10     vector<vector<vector<int>>> lcs(
11         str2.length() + 1,
12         vector<vector<int>>>(str1.length() + 1, vector<int>(4, 0)));
13     for (int i = 1; i < str2.length() + 1; i++) {
14         for (int j = 1; j < str1.length() + 1; j++) {
15             if (str2[i - 1] == str1[j - 1]) {
16                 lcs[i][j] = {str2[i - 1], lcs[i - 1][j - 1][1] + 1, i - 1, j - 1};
17             } else {
18                 if (lcs[i - 1][j][1] > lcs[i][j - 1][1]) {
19                     lcs[i][j] = {-1, lcs[i - 1][j][1], i - 1, j};
20                 } else {
21                     lcs[i][j] = {-1, lcs[i][j - 1][1], i, j - 1};
22                 }
23             }
24         }
25     }
26     return buildSequence(lcs);
27 }
28
29 vector<char> buildSequence(vector<vector<vector<int>>> lcs) {
30     vector<char> sequence;
31     int i = lcs.size() - 1;
32     int j = lcs[0].size() - 1;
33     while (i != 0 && j != 0) {
```

Your Solutions

Run Code

Solution 1Solution 2Solution 3

```
1 #include <vector>
2 using namespace std;
3
4 vector<char> longestCommonSubsequence(string str1, string str2) {
5     // Write your code here.
6     return {};
7 }
8
```

Our Tests

```
1 // Test 1: str1 = "abcde", str2 = "ace"
2 // Expected: "ace"
3
4 // Test 2: str1 = "abcde", str2 = "abcde"
5 // Expected: "abcde"
6
7 // Test 3: str1 = "abcde", str2 = "fghij"
8 // Expected: ""
```

Custom Output

Submit Code

```
1
2
3
4
5
6
7
8
9
10
```

