

PromptScratchpadOur Solution(s)Video Explanation

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 // O(nm*min(n, m)) time | O((min(n, m))^2) space
7 vector<char> longestCommonSubsequence(string str1, string str2) {
8     string small = str1.length() < str2.length() ? str1 : str2;
9     string big = str1.length() >= str2.length() ? str1 : str2;
10    vector<vector<char>> evenLcs;
11    vector<vector<char>> oddLcs;
12    for (int i = 0; i < small.length() + 1; i++) {
13        evenLcs.push_back(vector<char>());
14    }
15    for (int i = 0; i < small.length() + 1; i++) {
16        oddLcs.push_back(vector<char>());
17    }
18    for (int i = 1; i < big.length() + 1; i++) {
19        vector<vector<char>> *currentLcs;
20        vector<vector<char>> *previousLcs;
21        if (i % 2 == 1) {
22            currentLcs = &oddLcs;
23            previousLcs = &evenLcs;
24        } else {
25            currentLcs = &evenLcs;
26            previousLcs = &oddLcs;
27        }
28        for (int j = 1; j < small.length() + 1; j++) {
29            if (big[i - 1] == small[j - 1]) {
30                vector<char> copy = previousLcs->at(j - 1);
31                copy.push_back(big[i - 1]);
32                currentLcs->at(j) = copy;
33            } else {
34                currentLcs->at(j) =
35                    previousLcs->at(j).size() > currentLcs->at(j - 1).size()
36                        ? previousLcs->at(j)
37                        : currentLcs->at(j - 1);
38            }
39        }
40    }
41    return big.length() % 2 == 0 ? evenLcs[small.length()]
42        : oddLcs[small.length()];
43 }
44
```