

Our Solution(s)

Run Code

Your Solutions

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(nm*min(n, m)) space
7 vector<char> longestCommonSubsequence(string str1, string str2) {
8     vector<vector<vector<char>>> lcs;
9     for (int i = 0; i < str2.length() + 1; i++) {
10         lcs.push_back(vector<vector<char>>());
11         for (int j = 0; j < str1.length() + 1; j++) {
12             lcs[i].push_back(vector<char>());
13         }
14     }
15     for (int i = 1; i < str2.length() + 1; i++) {
16         for (int j = 1; j < str1.length() + 1; j++) {
17             if (str2[i - 1] == str1[j - 1]) {
18                 vector<char> copy = lcs[i - 1][j - 1];
19                 copy.push_back(str2[i - 1]);
20                 lcs[i][j] = copy;
21             } else {
22                 lcs[i][j] = lcs[i - 1][j].size() > lcs[i][j - 1].size() ? lcs[i - 1][j] : lcs[i][j - 1];
23             }
24         }
25     }
26 }
27 return lcs[str2.length()][str1.length()];
28 }
29
```

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

Custom Output

Submit Code

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

