

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) time | O(nm) space
4 function longestCommonSubsequence(str1, str2) {
5   const lcs = [];
6   for (let i = 0; i < str2.length + 1; i++) {
7     const row = [];
8     for (let j = 0; j < str1.length + 1; j++) {
9       const entry = new Array(4);
10      entry[1] = 0;
11      row.push(entry);
12    }
13    lcs.push(row);
14  }
15  for (let i = 1; i < str2.length + 1; i++) {
16    for (let j = 1; j < str1.length + 1; j++) {
17      if (str2[i - 1] === str1[j - 1]) {
18        lcs[i][j] = [str2[i - 1], lcs[i - 1][j - 1][1] + 1, i - 1, j - 1];
19      } else {
20        if (lcs[i - 1][j][1] > lcs[i][j - 1][1]) {
21          lcs[i][j] = [null, lcs[i - 1][j][1], i - 1, j];
22        } else {
23          lcs[i][j] = [null, lcs[i][j - 1][1], i, j - 1];
24        }
25      }
26    }
27  }
28  return buildSequence(lcs);
29 }
30
31 function buildSequence(lcs) {
32   const sequence = [];
33   let i = lcs.length - 1;
```

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

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

```
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.