

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 // O(2^(n + m)) time | O(n + m) space - where n is the length
4 // of the first string and m is the length of the second string
5 function interweavingStrings(one, two, three) {
6   if (three.length !== one.length + two.length) {
7     return false;
8   }
9
10  return areInterwoven(one, two, three, 0, 0);
11 }
12
13 function areInterwoven(one, two, three, i, j) {
14   const k = i + j;
15   if (k === three.length) return true;
16
17   if (i < one.length && one[i] === three[k]) {
18     if (areInterwoven(one, two, three, i + 1, j)) return true;
19   }
20
21   if (j < two.length && two[j] === three[k]) {
22     return areInterwoven(one, two, three, i, j + 1);
23   }
24
25   return false;
26 }
27
28 exports.interweavingStrings = interweavingStrings;
29
```

Solution 1

Solution 2

Solution 3

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

Our Tests

Custom Output

Submit Code

```
10 # Print out the number of times the word 'cherry' appears in the list
11
12 # Print out two elements from fruit, the first and the last element
13 fruit[0]
14 fruit[-1]
15
16 # Sum of all elements in fruit: num_fruit
17 num_fruit = len(fruit)
18
19 # Print out variable num_fruit
20 print(num_fruit)
```

Run or submit code when you're ready.