

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(2^(n + m)) time | O(n + m) space - where n is the length
5     // of the first string and m is the length of the second string
6     func interweavingStrings(_ one: String, _ two: String, _ three: String) -> Bool {
7         if three.length != one.length + two.length {
8             return false
9         }
10        return areInterwoven(one, two, three, 0, 0)
11    }
12
13    func areInterwoven(_ one: String, _ two: String, _ three: String, i: Int, j: Int) -> Bool {
14        let k = i + j
15        if k == three.length {
16            return true
17        }
18
19        let oneI = one.index(one.startIndex, offsetBy: i)
20        let twoJ = two.index(two.startIndex, offsetBy: j)
21        let threeK = three.index(three.startIndex, offsetBy: k)
22        if i < one.length, one[oneI] == three[threeK] {
23            if areInterwoven(one, two, three, i + 1, j) {
24                return true
25            }
26        }
27
28        if j < two.length, two[twoJ] == three[threeK] {
29            return areInterwoven(one, two, three, i, j + 1)
30        }
31
32        return false
33    }
34 }
```

Solution 1

Solution 2

Solution 3

```
1 class Program {
2     func interweavingStrings(_ one: String, _ two: String, _ three: String) -> Bool {
3         // Write your code here
4         return false
5     }
6 }
7
```

Our Tests

Custom Output

Submit Code

Test 1

Test 2

Test 3

Test 4

Test 5

Test 6

Test 7

Test 8

Test 9

Test 10

Test 11

Test 12

Test 13

Test 14

Test 15

Test 16

Test 17

Test 18

Test 19

Test 20

Test 21

Test 22

Test 23

Test 24

Test 25

Test 26

Test 27

Test 28

Test 29

Test 30

Run or submit code when you're ready.