## Our Solution(s)                                    Run Code

**Solution 1**    **Solution 2**

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# O(2^(n + m)) time | O(n + m) space - where n is the length
# of the first string and m is the length of the second string
def interweavingStrings(one, two, three):
    if len(three) != len(one) + len(two):
        return False

    return areInterwoven(one, two, three, 0, 0)


def areInterwoven(one, two, three, i, j):
    k = i + j
    if k == len(three):
        return True

    if i < len(one) and one[i] == three[k]:
        if areInterwoven(one, two, three, i + 1, j):
            return True

    if j < len(two) and two[j] == three[k]:
        return areInterwoven(one, two, three, i, j + 1)

    return False
```

## Your Solutions                                     Run Code

**Solution 1**    **Solution 2**    **Solution 3**

```python
def interweavingStrings(one, two, three):
    # Write your code here.
    pass
```

## Our Tests

## Custom Output                                      Submit Code

Run or submit code when you're ready.