Our Solution(s)   Run Code

Solution 1   **Solution 2**

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

# O(nm) time | O(nm) 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

    cache = [[None for j in range(len(two) + 1)] for i in range(len(o
    return areInterwoven(one, two, three, 0, 0, cache)


def areInterwoven(one, two, three, i, j, cache):
    if cache[i][j] is not None:
        return cache[i][j]

    k = i + j
    if k == len(three):
        return True

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

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

    cache[i][j] = False
    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.