Our Solution(s) Run Code

Solution 1   Solution 2

```go
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

package main

// O(nm) time | O(nm) space - where n is the length of the
// first string and m is the length of the second string
func InterweavingStrings(one, two, three string) bool {
  if len(three) != len(one)+len(two) {
    return false
  }
  cache := make([][]*bool, len(one)+1)
  for i := 0; i < len(one)+1; i++ {
    cache[i] = make([]*bool, len(two)+1)
  }
  return areInterwoven(one, two, three, 0, 0, cache)
}

func areInterwoven(one, two, three string, i, j int, cache [][]*bool)
  if cache[i][j] != nil {
    return *cache[i][j]
  }

  k := i + j
  if k == len(three) {
    return true
  }

  if i < len(one) && one[i] == three[k] {
    result := areInterwoven(one, two, three, i+1, j, cache)
    cache[i][j] = &result
    if result {
      return true
    }
```

Your Solutions Run Code

Solution 1   Solution 2   Solution 3

```go
package main

func InterweavingStrings(one, two, three string) bool {
  // Write your code here.
  return false
}
```

Our Tests

Custom Output Submit Code