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

using System.Collections.Generic;

public class Program {
  // O(nm) time | O(nm) space - where n is the length of the
  // first string and m is the length of the second string
  public static bool Interweavingstrings(string one, string two, string three) {
    if (three.Length != one.Length + two.Length) {
      return false;
    }

    bool?[,] cache = new bool?[one.Length+1, two.Length+1];
    return areInterwoven(one, two, three, 0, 0, cache);
  }

  public static bool areInterwoven(string one, string two, string three, int i, int j,
    bool?[,] cache) {
    if (cache[i,j].HasValue) {
      return cache[i,j].Value;
    }

    int k = i + j;
    if (k == three.Length) {
      return true;
    }

    if (i < one.Length && one[i] == three[k]) {
      cache[i,j] = areInterwoven(one, two, three, i + 1, j, cache);
      if (cache[i,j].HasValue && cache[i,j].Value) {
        return true;
      }
    }

    if (j < two.Length && two[j] == three[k]) {
      cache[i,j] = areInterwoven(one, two, three, i, j + 1, cache);
      return cache[i,j].Value;
    }

    cache[i,j] = false;
    return false;
  }
}

```