Our Solution(s) Run Code

Solution 1 Solution 2

```
// 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
function interweavingStrings(one, two, three) {
  if (three.length !== one.length + two.length) {
    return false;
  }

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

function areInterwoven(one, two, three, i, j) {
  const k = i + j;
  if (k === three.length) return true;

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

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

  return false;
}

exports.interweavingStrings = interweavingStrings;

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
function interweavingStrings(one, two, three) {
  // Write your code here.
}

// Do not edit the line below.
exports.interweavingStrings = interweavingStrings;

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.