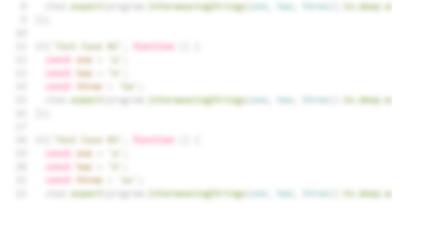**Our Solution(s)**   Run Code

Solution 1   Solution 2

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

  const cache = new Array(one.length + 1).fill(0).map(_ => new Array(
  return areInterwoven(one, two, three, 0, 0, cache);
}

function areInterwoven(one, two, three, i, j, cache) {
  if (cache[i][j] !== null) return cache[i][j];

  const 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]) 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];
  }

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

Your Solutions   Run Code

Solution 1   Solution 2   Solution 3

```javascript
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.