Our Solution(s) Run Code

Solution 1 Solution 2

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

// O(nm) time | O(min(n, m)) space
function levenshteinDistance(str1, str2) {
  const small = str1.length < str2.length ? str1 : str2;
  const big = str1.length >= str2.length ? str1 : str2;
  const evenEdits = [];
  const oddEdits = new Array(small.length + 1);
  for (let j = 0; j < small.length + 1; j++) {
    evenEdits.push(j);
  }
  for (let i = 1; i < big.length + 1; i++) {
    let currentEdits, previousEdits;
    if (i % 2 === 1) {
      currentEdits = oddEdits;
      previousEdits = evenEdits;
    } else {
      currentEdits = evenEdits;
      previousEdits = oddEdits;
    }
    currentEdits[0] = i;
    for (let j = 1; j < small.length + 1; j++) {
      if (big[i - 1] === small[j - 1]) {
        currentEdits[j] = previousEdits[j - 1];
      } else {
        currentEdits[j] = 1 + Math.min(previousEdits[j - 1], previousEdits[j], cur
      }
    }
  }
  return big.length % 2 === 0 ? evenEdits[small.length] : oddEdits[small.length];
}

exports.levenshteinDistance = levenshteinDistance;

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
function levenshteinDistance(str1, str2) {
  // Write your code here.
}

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

```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.