Prompt | Scratchpad | Our Solution(s) | Video Explanation | Run Code

Solution 1 | Solution 2

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

using System;
public class Program {
  // O(nm) time | O(nm) space
  public static int LevenshteinDistance(string str1, string str2) {
    int[,] edits = new int[str2.Length + 1,str1.Length + 1];
    for (int i = 0; i < str2.Length + 1; i++) {
      for (int j = 0; j < str1.Length + 1; j++) {
        edits[i,j] = j;
      }
      edits[i,0] = i;
    }
    for (int i = 1; i < str2.Length + 1; i++) {
      for (int j = 1; j < str1.Length + 1; j++) {
        if (str2[i  -1] == str1[j - 1]) {
          edits[i,j] = edits[i - 1,j - 1];
        } else {
          edits[i,
            j] = 1 +
            Math.Min(edits[i - 1,j - 1],
              Math.Min(edits[i - 1,j],
              edits[i,j - 1]));
        }
      }
    }
    return edits[str2.Length,str1.Length];
  }
}

```

Your Solutions | Run Code

Solution 1 | Solution 2 | Solution 3

```
public class Program {
  public static int LevenshteinDistance(string str1, string str2) {
    // Write your code here.
    return -1;
  }
}

```

Custom Output | Raw Output | Submit Code

Run or submit code when you're ready.