Solution 1

Run Code

Our Solution(s)

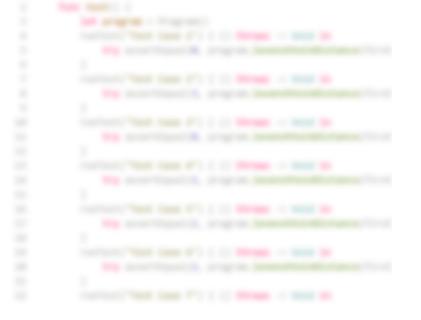
Run Code

```
Your Solutions
```

Solution 2

```
Solution 1
             Solution 2
 1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
 3
   class Program {
        // O(nd) time | O(min(n, m) space)
 5
        func levenshteinDistance(firstString: String, secondString:
            let small = firstString.count < secondString.count ? fil</pre>
6
            let big = firstString.count >= secondString.count ? firs
9
            var evenEdits = [Int]()
10
            var oddEdits = Array(repeating: 0, count: small.count +
11
12
            for i in 0 ..< small.count + 1 {</pre>
13
                evenEdits.append(i)
14
15
16
            for i in 1 ..< big.count + 1 {</pre>
17
                if i % 2 == 1 {
18
                    optimizedLevenshteinHelper(bigIndex: i, smallStr
19
                } else {
20
                    optimizedLevenshteinHelper(bigIndex: i, smallStr
21
23
            return big.count % 2 == 0 ? evenEdits[small.count] : odd
24
25
26
27
        func optimizedLevenshteinHelper(bigIndex: Int, smallString:
28
            currentEdits[0] = bigIndex
29
30
            for j in 1 ... smallString.count + 1 {
31
                let firstIndex = bigString.index(bigString.startInde
32
                let secondIndex = smallString.index(smallString.star
33
```

Solution 3



Run or submit code when you're ready.