

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1Solution 2

```
1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 # O(nm) time | O(min(n, m)) space
4 def levenshteinDistance(str1, str2):
5     small = str1 if len(str1) < len(str2) else str2
6     big = str1 if len(str1) >= len(str2) else str2
7     evenEdits = [x for x in range(len(small) + 1)]
8     oddEdits = [None for x in range(len(small) + 1)]
9     for i in range(1, len(big) + 1):
10         if i % 2 == 1:
11             currentEdits = oddEdits
12             previousEdits = evenEdits
13         else:
14             currentEdits = evenEdits
15             previousEdits = oddEdits
16         currentEdits[0] = i
17         for j in range(1, len(small) + 1):
18             if big[i - 1] == small[j - 1]:
19                 currentEdits[j] = previousEdits[j - 1]
20             else:
21                 currentEdits[j] = 1 + min(previousEdits[j - 1], previousEdits[j],
22                 return evenEdits[-1] if len(big) % 2 == 0 else oddEdits[-1]
```

Solution 1Solution 2Solution 3

1 def levenshteinDistance(str1, str2):
2 # Write your code here.
3 pass
4

Custom OutputRaw OutputSubmit Code

Run or submit code when you're ready.