

Our Solution(s)		Run Code	Your Solutions			Run Code
-----------------	--	----------	----------------	--	--	----------

Solution 1	Solution 2
<pre>1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved. 2 3 using namespace std; 4 5 // O(nm) time O(min(n, m)) space 6 int levenshteinDistance(string str1, string str2) { 7 string small = str1.length() < str2.length() ? str1 : str2; 8 string big = str1.length() >= str2.length() ? str1 : str2; 9 int *evenEdits = new int[small.length() + 1]; 10 int *oddEdits = new int[small.length() + 1]; 11 for (int j = 0; j < small.length() + 1; j++) { 12 evenEdits[j] = j; 13 } 14 int *currentEdits; 15 int *previousEdits; 16 for (int i = 1; i < big.length() + 1; i++) { 17 if (i % 2 == 1) { 18 currentEdits = oddEdits; 19 previousEdits = evenEdits; 20 } else { 21 currentEdits = evenEdits; 22 previousEdits = oddEdits; 23 } 24 currentEdits[0] = i; 25 for (int j = 1; j < small.length() + 1; j++) { 26 if (big[i - 1] == small[j - 1]) { 27 currentEdits[j] = previousEdits[j - 1]; 28 } else { 29 currentEdits[j] = 1 + min(previousEdits[j - 1], 30 min(previousEdits[j], currentE 31) 32 } 33 }</pre>	

Solution 1	Solution 2	Solution 3
<pre>1 using namespace std; 2 3 int levenshteinDistance(string str1, string str2) { 4 // Write your code here. 5 return -1; 6 } 7</pre>		

