

Our Solution(s)Run Code

Solution 1Solution 2

1

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

2

3

class Program {

4

// O(nm) time | O(nm) space

5

public static int levenshteinDistance(String str1, String str2) {

6

int[][] edits = new int[str2.length() + 1][str1.length() + 1];

7

for (int i = 0; i < str2.length() + 1; i++) {

8

for (int j = 0; j < str1.length() + 1; j++) {

9

edits[i][j] = j;

10

}

11

edits[i][0] = i;

12

}

13

for (int i = 1; i < str2.length() + 1; i++) {

14

for (int j = 1; j < str1.length() + 1; j++) {

15

if (str2.charAt(i - 1) == str1.charAt(j - 1)) {

16

edits[i][j] = edits[i - 1][j - 1];

17

} else {

18

edits[i][j] =

19

1 + Math.min(edits[i - 1][j - 1], Math.min(edits[i - 1][j], edits[i][j - 1]));

20

}

21

}

22

}

23

return edits[str2.length()][str1.length()];

24

}

25

}

26

Your SolutionsRun Code

Solution 1Solution 2Solution 3

1

class Program {

2

public static int levenshteinDistance(String str1, String str2) {

3

// Write your code here.

4

return -1;

5

}

6

}

7

