EditDistance.java

```
1      package DynamicProgramming;
2
3      // Code taken from: https://github.com/TheAlgorithms/Java/blob/master/src/main/java/com/thealgorithms/dyn
4
5      /**
6       * @author SUBHAM SANGHAI
7       */
8
9      public class EditDistance {
10
11         public int minDistance(String word1, String word2) {
12             int len1 = word1.length();
13             int len2 = word2.length();
14             // len1+1, len2+1, because finally return dp[len1][len2]
15 2           int[][] dp = new int[len1 + 1][len2 + 1];
16             /* If second string is empty, the only option is to
17         insert all characters of first string into second*/
18 2           for (int i = 0; i <= len1; i++) {
19                 dp[i][0] = i;
20             }
21             /* If first string is empty, the only option is to
22         insert all characters of second string into first*/
23 2           for (int j = 0; j <= len2; j++) {
24                 dp[0][j] = j;
25             }
26             // iterate though, and check last char
27 2           for (int i = 0; i < len1; i++) {
28                 char c1 = word1.charAt(i);
29 2               for (int j = 0; j < len2; j++) {
30                     char c2 = word2.charAt(j);
31                     // if last two chars equal
32 1                   if (c1 == c2) {
33                         // update dp value for +1 length
34 2                       dp[i + 1][j + 1] = dp[i][j];
35                     } else {
36                         /* if two characters are different ,
37              then take the minimum of the various operations(i.e insertion,removal,substitution)*/
38 1                       int replace = dp[i][j] + 1;
39 2                       int insert = dp[i][j + 1] + 1;
40 2                       int delete = dp[i + 1][j] + 1;
41
42 2                       int min = replace > insert ? insert : replace;
43 2                       min = delete > min ? min : delete;
44 2                       dp[i + 1][j + 1] = min;
45                     }
46                 }
47             }
48             /* return the final answer , after traversing through both the strings*/
49 1           return dp[len1][len2];
50         }
51     }
```

Mutations

```
15  1. Replaced integer addition with subtraction → KILLED
    2. Replaced integer addition with subtraction → KILLED
18  1. changed conditional boundary → KILLED
    2. negated conditional → KILLED
23  1. changed conditional boundary → SURVIVED
    2. negated conditional → KILLED
27  1. negated conditional → KILLED
    2. changed conditional boundary → KILLED
29  1. changed conditional boundary → KILLED
    2. negated conditional → KILLED
32  1. negated conditional → KILLED
34  1. Replaced integer addition with subtraction → KILLED
    2. Replaced integer addition with subtraction → KILLED
38  1. Replaced integer addition with subtraction → KILLED
39  1. Replaced integer addition with subtraction → KILLED
    2. Replaced integer addition with subtraction → KILLED
40  1. Replaced integer addition with subtraction → KILLED
    2. Replaced integer addition with subtraction → KILLED
42  1. changed conditional boundary → SURVIVED
    2. negated conditional → KILLED
43  1. changed conditional boundary → SURVIVED
```

```
     2. negated conditional → KILLED
44   1. Replaced integer addition with subtraction → KILLED
     2. Replaced integer addition with subtraction → KILLED
49   1. replaced int return with 0 for DynamicProgramming/EditDistance::minDistance → KILLED
```

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

Tests examined

- DynamicProgramming.EditDistanceTest.testMinDistance(DynamicProgramming.EditDistanceTest) (0 ms)

Report generated by PIT 1.15.0