LIS.java

```
1   package com.example.dynamicProgramming;
2
3   import java.util.Arrays;
4
5   public class LIS {
6       public int method1(int[] nums) {
7           int n = nums.length;
8           int[] dp = new int[n];
9           int[] count = new int[n];
10 1        Arrays.fill(dp, 1);
11 1        Arrays.fill(count, 1);
12          int max = 1;
13
14 2         for (int i = 1; i < n; i++) {
15 2             for (int j = 0; j < i; j++) {
16 2                 if (nums[i] > nums[j]) {
17 3                     if (dp[j] + 1 > dp[i]) {
18 1                         dp[i] = dp[j] + 1;
19                           count[i] = count[j];
20 2                     } else if (dp[j] + 1 == dp[i]) {
21 1                         count[i] += count[j];
22                       }
23                       max = Math.max(dp[i], max);
24                  }
25              }
26          }
27          int ans = 0;
28 2         for (int i = 0; i < n; i++) {
29 1             if (dp[i] == max) {
30 1                 ans += count[i];
31              }
32          }
33 1        return ans;
34
35      }
36
37      public int method2(int[] nums) {
38 1          if (nums.length == 0) {
39              return 0;
40          }
41          int n = nums.length;
42          int[] lis = new int[n];
43          int[] fq = new int[n];
44          lis[0] = 1;
45          fq[0] = 1;
46          int lo = 1;
47
48 2         for (int i = 1; i < nums.length; i++) {
49              int mx = 0;
50              int c = 1;
51 2             for (int j = 0; j < i; j++) {
52 2                 if (nums[j] < nums[i]) {
53 2                     if (lis[j] > mx) {
54                          mx = lis[j];
55                          c = fq[j];
56 1                     } else if (lis[j] == mx) {
57 1                         c += fq[j];
58                      }
59                  }
```

```
60                  }
61                  fq[i] = c;
62 1                lis[i] = mx + 1;
63 2                if (lo < lis[i]) {
64                      lo = lis[i];
65                  }
66              }
67
68              int count = 0;
69 2            for (int i = 0; i < nums.length; i++) {
70 1                if (lis[i] == lo) {
71 1                    count += fq[i];
72                  }
73              }
74
75 1            return count;
76          }
77      }
```

Mutations

10	1. removed call to java/util/Arrays::fill → KILLED
11	1. removed call to java/util/Arrays::fill → KILLED
14	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
15	1. negated conditional → KILLED 2. changed conditional boundary → SURVIVED
16	1. changed conditional boundary → SURVIVED 2. negated conditional → KILLED
17	1. Replaced integer addition with subtraction → KILLED 2. negated conditional → KILLED 3. changed conditional boundary → KILLED
18	1. Replaced integer addition with subtraction → SURVIVED
20	1. Replaced integer addition with subtraction → KILLED 2. negated conditional → KILLED
21	1. Replaced integer addition with subtraction → KILLED
28	1. negated conditional → KILLED 2. changed conditional boundary → KILLED
29	1. negated conditional → KILLED
30	1. Replaced integer addition with subtraction → KILLED
33	1. replaced int return with 0 for com/example/dynamicProgramming/LIS::method1 → KILLED
38	1. negated conditional → KILLED
48	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
51	1. changed conditional boundary → SURVIVED 2. negated conditional → KILLED
52	1. negated conditional → KILLED 2. changed conditional boundary → SURVIVED
53	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
56	1. negated conditional → KILLED
57	1. Replaced integer addition with subtraction → KILLED
62	1. Replaced integer addition with subtraction → KILLED
63	1. negated conditional → SURVIVED 2. changed conditional boundary → SURVIVED
69	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
70	1. negated conditional → KILLED
71	1. Replaced integer addition with subtraction → KILLED
75	1. replaced int return with 0 for com/example/dynamicProgramming/LIS::method2 → 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

- com.example.dynamicProgramming.LISTest.testMethod1WithDescendingOrder(com.example.dynamicProgramming.LISTest) (0 ms)
- com.example.dynamicProgramming.LISTest.testMethod1(com.example.dynamicProgramming.LISTest) (0 ms)
- com.example.dynamicProgramming.LISTest.testMethod1WithSingleElement(com.example.dynamicProgramming.LISTest) (0 ms)
- com.example.dynamicProgramming.LISTest.testMethod1WithEmptyArray(com.example.dynamicProgramming.LISTest) (0 ms)
- com.example.dynamicProgramming.LISTest.testMethod2WithDescendingOrder(com.example.dynamicProgramming.LISTest) (0 ms)
- com.example.dynamicProgramming.LISTest.testMethod2(com.example.dynamicProgramming.LISTest) (0 ms)
- com.example.dynamicProgramming.LISTest.testMethod2WithSingleElement(com.example.dynamicProgramming.LISTest) (0 ms)
- com.example.dynamicProgramming.LISTest.testMethod2WithEmptyArray(com.example.dynamicProgramming.LISTest) (0 ms)

Report generated by PIT 1.15.0