LIS.java

```
   package com.example;

   import java.util.Arrays;

   public class LIS {
       public int method1(int[] nums) {
           int n = nums.length;
           int [] dp = new int[n];
           int [] count = new int[n];
1          Arrays.fill(dp, 1);
1          Arrays.fill(count,1);
           int max = 1;

3          for (int i = 1; i < n; i++) {
3              for (int j = 0; j < i; j++) {
2                  if (nums[i] > nums[j]) {
3                      if (dp[j] + 1 > dp[i]) {
1                          dp[i] = dp[j] + 1;
                           count[i] = count[j];
2                      } else if (dp[j] + 1 == dp[i]) {
1                          count[i] += count[j];
                       }
                   max = Math.max(dp[i],max);
                   }
               }
           }
           int ans = 0;
3          for (int i = 0; i < n; i++) {
1              if (dp[i] == max) {
1                  ans += count[i];
               }
           }
1          return ans;

       }
       public int method2(int[] nums) {
1          if(nums.length==0){
               return 0;
           }
           int n = nums.length;
           int[] lis = new int[n];
           int[] fq = new int[n];
           lis[0] = 1;
           fq[0] = 1;
           int lo = 1;

3          for (int i = 1; i < nums.length; i++) {
```

```
48               int mx = 0;
49               int c = 1;
50 3             for (int j = 0; j < i; j++) {
51 2                 if (nums[j] < nums[i]) {
52 2                     if (lis[j] > mx) {
53                           mx = lis[j];
54                           c = fq[j];
55 1                     } else if (lis[j] == mx) {
56 1                         c += fq[j];
57                       }
58                   }
59               }
60               fq[i] = c;
61 1             lis[i] = mx + 1;
62 2             if (lo < lis[i]) {
63                   lo = lis[i];
64               }
65           }
66
67           int count = 0;
68 3         for (int i = 0; i < nums.length; i++) {
69 1             if (lis[i] == lo) {
70 1                 count += fq[i];
71               }
72           }
73
74 1         return count;
75       }
76   }
```

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. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
15	1. changed conditional boundary → SURVIVED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
16	1. changed conditional boundary → SURVIVED 2. negated conditional → KILLED
17	1. changed conditional boundary → KILLED 2. Replaced integer addition with subtraction → KILLED 3. negated conditional → 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. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
29	1. negated conditional → KILLED

30	1. Replaced integer addition with subtraction → KILLED
33	1. replaced int return with 0 for com/example/LIS::method1 → KILLED
37	1. negated conditional → KILLED
47	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
50	1. changed conditional boundary → SURVIVED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
51	1. changed conditional boundary → SURVIVED 2. negated conditional → KILLED
52	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
55	1. negated conditional → KILLED
56	1. Replaced integer addition with subtraction → KILLED
61	1. Replaced integer addition with subtraction → KILLED
62	1. changed conditional boundary → SURVIVED 2. negated conditional → SURVIVED
68	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
69	1. negated conditional → KILLED
70	1. Replaced integer addition with subtraction → KILLED
74	1. replaced int return with 0 for com/example/LIS::method2 → KILLED

Active mutators

- BOOLEAN_FALSE_RETURN
- BOOLEAN_TRUE_RETURN
- CONDITIONALS_BOUNDARY_MUTATOR
- EMPTY_RETURN_VALUES
- INCREMENTS_MUTATOR
- INVERT_NEGS_MUTATOR
- MATH_MUTATOR
- NEGATE_CONDITIONALS_MUTATOR
- NULL_RETURN_VALUES
- PRIMITIVE_RETURN_VALS_MUTATOR
- VOID_METHOD_CALL_MUTATOR

Tests examined

- com.example.LISTest.testMethod1(com.example.LISTest) (1 ms)
- com.example.LISTest.testMethod1WithEmptyArray(com.example.LISTest) (0 ms)
- com.example.LISTest.testMethod2WithEmptyArray(com.example.LISTest) (0 ms)
- com.example.LISTest.testMethod1WithDescendingOrder(com.example.LISTest) (0 ms)
- com.example.LISTest.testMethod1WithSingleElement(com.example.LISTest) (1 ms)
- com.example.LISTest.testMethod2WithSingleElement(com.example.LISTest) (1 ms)
- com.example.LISTest.testMethod2(com.example.LISTest) (0 ms)
- com.example.LISTest.testMethod2WithDescendingOrder(com.example.LISTest) (1 ms)

Report generated by PIT 1.5.0