CountInversions.java

```
   package com.example;

   import java.util.ArrayList;

   public class CountInversions {

            public static int numberOfInversions1(int[] a, int n) {
                   // Count the number of pairs:
                   int cnt = 0;
3                  for (int i = 0; i < n; i++) {
4                      for (int j = i + 1; j < n; j++) {
3                          if (a[i] > a[j]) cnt++;
                       }
                   }
1                  return cnt;
               }

            private static int merge(int[] arr, int low, int mid, int high) {
                   ArrayList<Integer> temp = new ArrayList<>(); // temporary array
                   int left = low;      // starting index of left half of arr
1                  int right = mid + 1;   // starting index of right half of arr

                   //Modification 1: cnt variable to count the pairs:
                   int cnt = 0;

                   //storing elements in the temporary array in a sorted manner//

4                  while (left <= mid && right <= high) {
2                      if (arr[left] <= arr[right]) {
                           temp.add(arr[left]);
1                          left++;
                       } else {
                           temp.add(arr[right]);
3                          cnt += (mid - left + 1); //Modification 2
1                          right++;
                       }
                   }

                   // if elements on the left half are still left //

2                  while (left <= mid) {
                       temp.add(arr[left]);
1                      left++;
                   }

                   //  if elements on the right half are still left //
2                  while (right <= high) {
                       temp.add(arr[right]);
1                      right++;
                   }

                   // transfering all elements from temporary to arr //
3                  for (int i = low; i <= high; i++) {
1                      arr[i] = temp.get(i - low);
                   }
1                  return cnt; // Modification 3
               }
58
59               public static int mergeSort(int[] arr, int low, int high) {
60                   int cnt = 0;
61 3                 if (low >= high) return cnt;
62 2                 int mid = (low + high) / 2 ;
63 1                 cnt += mergeSort(arr, low, mid);  // left half
64 2                 cnt += mergeSort(arr, mid + 1, high); // right half
65 1                 cnt += merge(arr, low, mid, high);  // merging sorted halves
66 1                 return cnt;
67               }
68
69               public static int numberOfInversions(int[] a, int n) {
70                   // Count the number of pairs:
71 2                 return mergeSort(a, 0, n - 1);
72               }
73
74   }
```

Mutations

10	1. changed conditional boundary → SURVIVED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → SURVIVED
11	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. Replaced integer addition with subtraction → KILLED 4. negated conditional → KILLED
12	1. changed conditional boundary → SURVIVED 2. Changed increment from 1 to -1 → NO_COVERAGE 3. negated conditional → KILLED
15	1. replaced int return with 0 for com/example /CountInversions::numberOfInversions1 → SURVIVED
21	1. Replaced integer addition with subtraction → KILLED
28	1. changed conditional boundary → KILLED 2. changed conditional boundary → KILLED 3. negated conditional → KILLED 4. negated conditional → KILLED
29	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
31	1. Changed increment from 1 to -1 → KILLED
34	1. Replaced integer subtraction with addition → KILLED 2. Replaced integer addition with subtraction → KILLED 3. Replaced integer addition with subtraction → KILLED
35	1. Changed increment from 1 to -1 → KILLED
41	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
43	1. Changed increment from 1 to -1 → KILLED
47	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
49	1. Changed increment from 1 to -1 → KILLED
53	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
54	1. Replaced integer subtraction with addition → KILLED
56	1. replaced int return with 0 for com/example/CountInversions::merge → KILLED
61	1. changed conditional boundary → KILLED 2. negated conditional → KILLED 3. replaced int return with 0 for com/example/CountInversions::mergeSort → SURVIVED
62	1. Replaced integer addition with subtraction → KILLED 2. Replaced integer division with multiplication → KILLED
63	1. Replaced integer addition with subtraction → KILLED
64	1. Replaced integer addition with subtraction → KILLED 2. Replaced integer addition with subtraction → KILLED
65	1. Replaced integer addition with subtraction → KILLED

66	1. replaced int return with 0 for com/example/CountInversions::mergeSort → KILLED
71	1. Replaced integer subtraction with addition → KILLED 2. replaced int return with 0 for com/example/CountInversions::numberOfInversions → 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.CountInversionsTest.test(com.example.CountInversionsTest) (1 ms)

Report generated by PIT 1.5.0