

ReversePairs.java

```
1 package com.example;
2
3 import java.util.ArrayList;
4
5 public class ReversePairs {
6
7     public static int countPairs(int[] a, int n) {
8
9         // Count the number of pairs:
10         int cnt = 0;
11         for (int i = 0; i < n; i++) {
12             for (int j = i + 1; j < n; j++) {
13                 if (a[i] > 2 * a[j])
14                     cnt++;
15             }
16         }
17         return cnt;
18     }
19
20     public static int team1(int[] skill, int n) {
21         return countPairs(skill, n);
22     }
23
24     private static void merge(int[] arr, int low, int mid, int high) {
25         ArrayList<Integer> temp = new ArrayList<>(); // temporary array
26         int left = low; // starting index of left half of arr
27         int right = mid + 1; // starting index of right half of arr
28
29         //storing elements in the temporary array in a sorted manner//
30
31         while (left <= mid && right <= high) {
32             if (arr[left] <= arr[right]) {
33                 temp.add(arr[left]);
34                 left++;
35             } else {
36                 temp.add(arr[right]);
37                 right++;
38             }
39         }
40
41         // if elements on the left half are still left //
42
43         while (left <= mid) {
44             temp.add(arr[left]);
45             left++;
46         }
47
48         // if elements on the right half are still left //
49         while (right <= high) {
50             temp.add(arr[right]);
51             right++;
52         }
53
54         // transferring all elements from temporary to arr //
55         for (int i = low; i <= high; i++) {
56             arr[i] = temp.get(i - low);
57         }
58     }
```

```

59
60     public static int countPairs(int[] arr, int low, int mid, int high) {
61 1         int right = mid + 1;
62         int cnt = 0;
63 3         for (int i = low; i <= mid; i++) {
64 6             while (right <= high && arr[i] > 2 * arr[right]) right++;
65 3             cnt += (right - (mid + 1));
66         }
67 1         return cnt;
68     }
69
70     public static int mergeSort(int[] arr, int low, int high) {
71         int cnt = 0;
72 3         if (low >= high) return cnt;
73 2         int mid = (low + high) / 2 ;
74 1         cnt += mergeSort(arr, low, mid); // left half
75 2         cnt += mergeSort(arr, mid + 1, high); // right half
76 1         cnt += countPairs(arr, low, mid, high); //Modification
77 1         merge(arr, low, mid, high); // merging sorted halves
78 1         return cnt;
79     }
80
81     public static int team(int[] skill, int n) {
82 2         return mergeSort(skill, 0, n - 1);
83     }
84 }

```

Mutations

```

11 1. changed conditional boundary → SURVIVED
    2. Changed increment from 1 to -1 → KILLED
    3. negated conditional → KILLED
12 1. changed conditional boundary → KILLED
    2. Changed increment from 1 to -1 → KILLED
    3. Replaced integer addition with subtraction → KILLED
    4. negated conditional → KILLED
13 1. changed conditional boundary → SURVIVED
    2. Replaced integer multiplication with division → KILLED
    3. negated conditional → KILLED
14 1. Changed increment from 1 to -1 → NO_COVERAGE
17 1. replaced int return with 0 for com/example/ReversePairs::countPairs → SURVIVED
21 1. replaced int return with 0 for com/example/ReversePairs::team1 → SURVIVED
27 1. Replaced integer addition with subtraction → KILLED
31 1. changed conditional boundary → KILLED
    2. changed conditional boundary → KILLED
    3. negated conditional → KILLED
    4. negated conditional → KILLED
32 1. changed conditional boundary → SURVIVED
    2. negated conditional → KILLED
34 1. Changed increment from 1 to -1 → KILLED
37 1. Changed increment from 1 to -1 → SURVIVED
43 1. changed conditional boundary → KILLED
    2. negated conditional → KILLED
45 1. Changed increment from 1 to -1 → KILLED
49 1. changed conditional boundary → KILLED
    2. negated conditional → KILLED
51 1. Changed increment from 1 to -1 → KILLED
55 1. changed conditional boundary → KILLED
    2. Changed increment from 1 to -1 → KILLED
    3. negated conditional → KILLED
56 1. Replaced integer subtraction with addition → KILLED
61 1. Replaced integer addition with subtraction → KILLED
63 1. changed conditional boundary → KILLED
    2. Changed increment from 1 to -1 → KILLED
    3. negated conditional → KILLED

```

64	1. changed conditional boundary → KILLED 2. changed conditional boundary → KILLED 3. Changed increment from 1 to -1 → KILLED 4. Replaced integer multiplication with division → KILLED 5. negated conditional → KILLED 6. negated conditional → KILLED
65	1. Replaced integer addition with subtraction → KILLED 2. Replaced integer subtraction with addition → KILLED 3. Replaced integer addition with subtraction → KILLED
67	1. replaced int return with 0 for com/example/ReversePairs::countPairs → KILLED
72	1. changed conditional boundary → KILLED 2. negated conditional → KILLED 3. replaced int return with 0 for com/example/ReversePairs::mergeSort → SURVIVED
73	1. Replaced integer addition with subtraction → KILLED 2. Replaced integer division with multiplication → KILLED
74	1. Replaced integer addition with subtraction → SURVIVED
75	1. Replaced integer addition with subtraction → KILLED 2. Replaced integer addition with subtraction → KILLED
76	1. Replaced integer addition with subtraction → KILLED
77	1. removed call to com/example/ReversePairs::merge → KILLED
78	1. replaced int return with 0 for com/example/ReversePairs::mergeSort → KILLED
82	1. Replaced integer subtraction with addition → KILLED 2. replaced int return with 0 for com/example/ReversePairs::team → 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.ReversePairsTest.testReversePairs(com.example.ReversePairsTest) (1 ms)

Report generated by [PIT](#) 1.5.0