

## SegmentTree.java

```

1  package TreesTesting;
2
3  public class SegmentTree {
4      static class Node {
5
6          private final int start, end; // start and end of the segment represented by this node
7          private int value; // value is the sum of all elements in the range [start, end)
8          private int lazy; // lazied value that should be added to children nodes
9          Node left, right; // left and right children
10
11         public Node(int start, int end, int value) {
12             this.start = start;
13             this.end = end;
14             this.value = value;
15             this.lazy = 0;
16             this.left = null;
17             this.right = null;
18         }
19
20         public void applyUpdate(int diff) {
21             1 this.lazy += diff;
22             3 this.value += (this.end - this.start) * diff;
23         }
24
25         public void shift() {
26             1 if (lazy == 0) return;
27             2 if (this.left == null && this.right == null) return;
28             1 this.value += this.lazy;
29             2 if (this.left != null) this.left.applyUpdate(this.lazy);
30             2 if (this.right != null) this.right.applyUpdate(this.lazy);
31             this.lazy = 0;
32         }
33
34         static Node merge(Node left, Node right) {
35             2 if (left == null) return right;
36             2 if (right == null) return left;
37             Node result = new Node(
38                 left.start,
39                 right.end,
40                 1 left.value + right.value
41             );
42             result.left = left;
43             result.right = right;
44             1 return result;
45         }
46
47         public int getValue() {
48             1 return value;
49         }
50
51         public Node getLeft() {
52             1 return left;
53         }
54
55         public Node getRight() {
56             1 return right;
57         }

```

```

58     }
59
60     private Node root;
61     public SegmentTree(int[] array) {
62         this.root = buildTree(array, 0, array.length);
63     }
64
65     private Node buildTree(int[] array, int start, int end) {
66         if (end - start < 2) return new Node(start, end, array[start]);
67         int mid = (start + end) >> 1;
68         Node left = buildTree(array, start, mid);
69         Node right = buildTree(array, mid, end);
70         return Node.merge(left, right);
71     }
72
73     private void updateRange(int left, int right, int diff, Node curr) {
74         if (left <= curr.start && curr.end <= right) {
75             curr.applyUpdate(diff);
76             return;
77         }
78         if (left >= curr.end || right <= curr.start) return;
79         curr.shift();
80         updateRange(left, right, diff, curr.left);
81         updateRange(left, right, diff, curr.right);
82         Node merge = Node.merge(curr.left, curr.right);
83         curr.value = merge.value;
84     }
85
86     private Node getRange(int left, int right, Node curr) {
87         if (left <= curr.start && curr.end <= right) return curr;
88         if (left >= curr.end || right <= curr.start) return null;
89         curr.shift();
90         return Node.merge(
91             getRange(left, right, curr.left),
92             getRange(left, right, curr.right)
93         );
94     }
95
96     public int getRange(int left, int right) {
97         Node result = getRange(left, right, root);
98         return result == null ? 0 : result.getValue();
99     }
100
101     public void updateRange(int left, int right, int diff) {
102         updateRange(left, right, diff, root);
103     }
104
105     public Node getRoot() {
106         return root;
107     }
108 }

```

## Mutations

- [21](#) 1. Replaced integer addition with subtraction → KILLED
- [22](#) 1. Replaced integer subtraction with addition → KILLED
- [22](#) 2. Replaced integer multiplication with division → SURVIVED
- [26](#) 3. Replaced integer addition with subtraction → KILLED
- [26](#) 1. negated conditional → KILLED
- [27](#) 1. negated conditional → SURVIVED
- [27](#) 2. negated conditional → NO\_COVERAGE
- [28](#) 1. Replaced integer addition with subtraction → SURVIVED

|                     |  |
|---------------------|--|
| <a href="#">29</a>  | 1. negated conditional → KILLED  |
|                     | 2. removed call to TreesTesting/SegmentTree\$Node::applyUpdate → KILLED                  |
| <a href="#">30</a>  | 1. negated conditional → KILLED  |
|                     | 2. removed call to TreesTesting/SegmentTree\$Node::applyUpdate → KILLED                  |
| <a href="#">35</a>  | 1. negated conditional → KILLED  |
|                     | 2. replaced return value with null for TreesTesting/SegmentTree\$Node::merge → KILLED    |
| <a href="#">36</a>  | 1. negated conditional → KILLED  |
|                     | 2. replaced return value with null for TreesTesting/SegmentTree\$Node::merge → KILLED    |
| <a href="#">40</a>  | 1. Replaced integer addition with subtraction → KILLED                                   |
| <a href="#">44</a>  | 1. replaced return value with null for TreesTesting/SegmentTree\$Node::merge → KILLED    |
| <a href="#">48</a>  | 1. replaced int return with 0 for TreesTesting/SegmentTree\$Node::getValue → KILLED      |
| <a href="#">52</a>  | 1. replaced return value with null for TreesTesting/SegmentTree\$Node::getLeft → KILLED  |
| <a href="#">56</a>  | 1. replaced return value with null for TreesTesting/SegmentTree\$Node::getRight → KILLED |
|                     | 1. changed conditional boundary → KILLED   |
| <a href="#">66</a>  | 2. Replaced integer subtraction with addition → KILLED                                   |
|                     | 3. negated conditional → KILLED  |
|                     | 4. replaced return value with null for TreesTesting/SegmentTree::buildTree → KILLED      |
| <a href="#">67</a>  | 1. Replaced integer addition with subtraction → KILLED                                   |
|                     | 2. Replaced Shift Right with Shift Left → KILLED   |
| <a href="#">70</a>  | 1. replaced return value with null for TreesTesting/SegmentTree::buildTree → KILLED      |
|                     | 1. changed conditional boundary → KILLED   |
| <a href="#">74</a>  | 2. changed conditional boundary → KILLED   |
|                     | 3. negated conditional → KILLED  |
|                     | 4. negated conditional → KILLED  |
| <a href="#">75</a>  | 1. removed call to TreesTesting/SegmentTree\$Node::applyUpdate → KILLED                  |
|                     | 1. changed conditional boundary → KILLED   |
| <a href="#">78</a>  | 2. changed conditional boundary → KILLED   |
|                     | 3. negated conditional → KILLED  |
|                     | 4. negated conditional → KILLED  |
| <a href="#">79</a>  | 1. removed call to TreesTesting/SegmentTree\$Node::shift → KILLED                        |
| <a href="#">80</a>  | 1. removed call to TreesTesting/SegmentTree::updateRange → KILLED                        |
| <a href="#">81</a>  | 1. removed call to TreesTesting/SegmentTree::updateRange → KILLED                        |
|                     | 1. changed conditional boundary → KILLED   |
|                     | 2. changed conditional boundary → KILLED   |
| <a href="#">87</a>  | 3. negated conditional → KILLED  |
|                     | 4. negated conditional → KILLED  |
|                     | 5. replaced return value with null for TreesTesting/SegmentTree::getRange → KILLED       |
|                     | 1. changed conditional boundary → KILLED   |
| <a href="#">88</a>  | 2. changed conditional boundary → KILLED   |
|                     | 3. negated conditional → KILLED  |
|                     | 4. negated conditional → KILLED  |
| <a href="#">89</a>  | 1. removed call to TreesTesting/SegmentTree\$Node::shift → SURVIVED                      |
| <a href="#">90</a>  | 1. replaced return value with null for TreesTesting/SegmentTree::getRange → KILLED       |
| <a href="#">98</a>  | 1. negated conditional → KILLED  |
|                     | 2. replaced int return with 0 for TreesTesting/SegmentTree::getRange → KILLED            |
| <a href="#">102</a> | 1. removed call to TreesTesting/SegmentTree::updateRange → KILLED                        |
| <a href="#">106</a> | 1. replaced return value with null for TreesTesting/SegmentTree::getRoot → 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

- TreesTesting.AllTreeTesting.[engine:junit-jupiter]/[class:TreesTesting.AllTreeTesting]/[method:get()] (10 ms)
- TreesTesting.AllTreeTesting.[engine:junit-jupiter]/[class:TreesTesting.AllTreeTesting]/[method:update()] (13 ms)
- TreesTesting.AllTreeTesting.[engine:junit-jupiter]/[class:TreesTesting.AllTreeTesting]/[method:build()] (31 ms)
- TreesTesting.AllTreeTesting.[engine:junit-jupiter]/[class:TreesTesting.AllTreeTesting]/[method:updateAndGet()] (16 ms)

Report generated by [PIT](#) 1.6.8