

```

1 import components.sequence.Sequence;
2
3 /**
4  * Simple HelloWorld program (clear of Checkstyle and FindBugs warnings).
5  *
6  * @author Sam Espanioly
7  */
8 public final class HW21 {
9
10     /**
11      * Default constructor--private to prevent instantiation.
12      */
13     private HW21() {
14         // no code needed here
15     }
16
17     /**
18      * Returns the size of the given {@code Tree<T>}.
19      *
20      * @param <T>
21      *         the type of the {@code Tree} node labels
22      * @param t
23      *         the {@code Tree} whose size to return
24      * @return the size of the given {@code Tree}
25      * @ensures size = |t|
26      */
27     public static <T> int size(Tree<T> t) {
28         int i = 0;
29         Sequence<Tree<T>> children = new Sequence1L();
30         T root = t.root();
31         t.disassemble(children);
32         i = i + children.length();
33         t.assemble(root, children);
34         return i;
35     }
36
37     /**
38      * Returns the height of the given {@code Tree<T>}.
39      *
40      * @param <T>
41      *         the type of the {@code Tree} node labels
42      * @param t
43      *         the {@code Tree} whose height to return
44      * @return the height of the given {@code Tree}
45      * @ensures height = ht(t)
46      */
47     public static <T> int height(Tree<T> t) {
48         int i = 0;
49         if (t.size() > 0) {
50             Sequence<Tree<T>> children = new Sequence1L();
51             T root = t.root();
52             t.disassemble(children);
53             i = i + 1 + height(t);
54             t.assemble(root, children);
55         }
56         return i;
57     }
58 }

```

```
62
63 /**
64  * Returns the largest integer in the given {@code Tree<Integer>}.
65  *
66  * @param t
67  *         the {@code Tree<Integer>} whose largest integer to return
68  * @return the largest integer in the given {@code Tree<Integer>}
69  * @requires |t| > 0
70  * @ensures <pre>
71  * max is in labels(t) and
72  * for all i: integer where (i is in labels(t)) (i <= max)
73  * </pre>
74  */
75 public static int max(Tree<Integer> t) {
76     int i = -1;
77     if (t.size() > 0) {
78         Sequence<Tree<Integer>> children = new Sequence1L();
79         Integer root = t.root();
80         t.disassemble(children);
81         i = max t;
82         if (root > i) {
83             i = root;
84         }
85         t.assemble(root, children);
86     }
87     return i;
88 }
89
90 /**
91  * Main method.
92  *
93  * @param args
94  *         the command line arguments; unused here
95  */
96 public static void main(String[] args) {
97     SimpleWriter out = new SimpleWriter1L();
98     out.println("Hello World!");
99     out.close();
100 }
101
102 }
103
```