```java
import components.binarytree.BinaryTree;
import components.binarytree.BinaryTree1;

/**
 * {@code Queue} represented as a {@code Sequence} of entries, with
 * implementations of primary methods.
 *
 * @param <T>
 *            type of {@code Queue} entries
 * @correspondence this = $this.entries
 */
public class HelloWorld {

    /**
     * Returns the size of the given {@code BinaryTree<T>}.
     *
     * @param <T>
     *            the type of the {@code BinaryTree} node labels
     * @param t
     *            the {@code BinaryTree} whose size to return
     * @return the size of the given {@code BinaryTree}
     * @ensures size = |t|
     */
    public static <T> int size(BinaryTree<T> t) {
        int count = 1;

        if (t.height() > 0) {
            BinaryTree<T> left = new BinaryTree1<>();
            BinaryTree<T> right = new BinaryTree1<>();
            T root = t.disassemble(left, right);
            int leftCount = size(left);
            int rightCount = size(right);
            count += leftCount + rightCount;
            t.assemble(root, left, right);
        }

        return count;
    }

    /**
     * Returns the size of the given {@code BinaryTree<T>}.
     *
     * @param <T>
     *            the type of the {@code BinaryTree} node labels
     * @param t
     *            the {@code BinaryTree} whose size to return
     * @return the size of the given {@code BinaryTree}
     * @ensures size = |t|
     */
    public static <T> int size(BinaryTree<T> t) {
        int count = 1;

        for (T x : t) {
            count++;
        }

        return count;
    }

}
```