

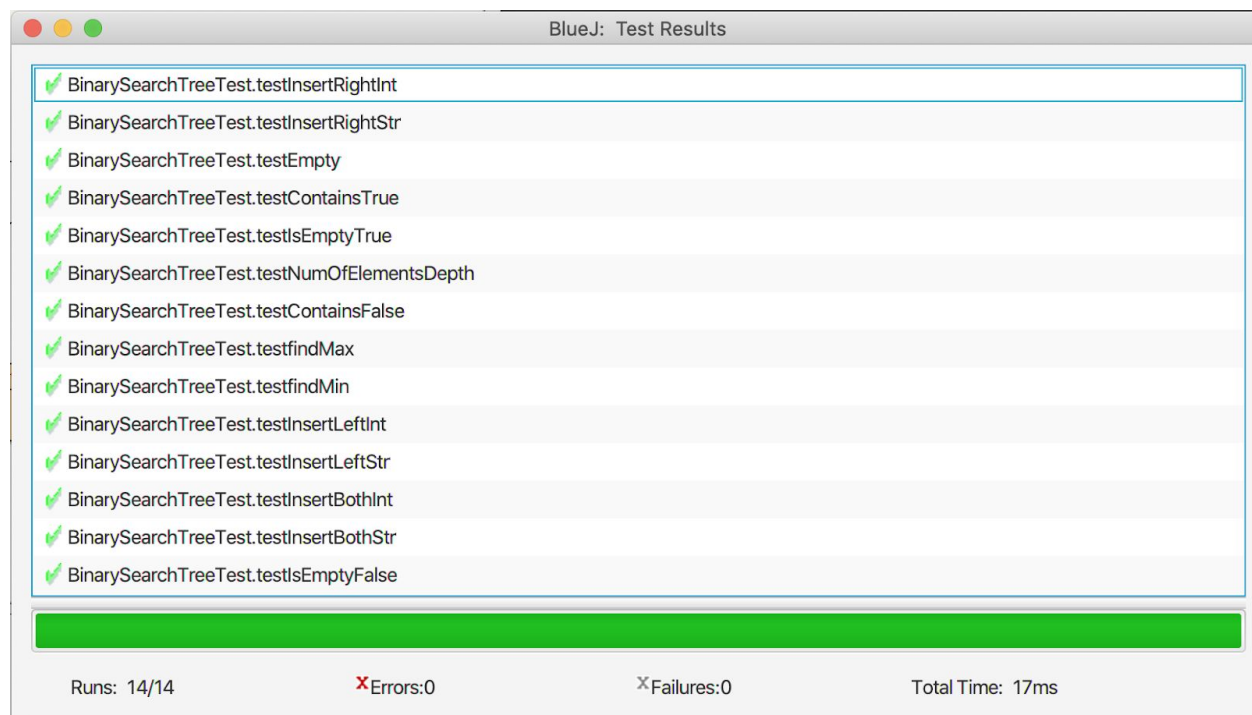
Dylan Maloy
CS150 Lab 7
Lab #7 write-up
10/27/19

Introduction:

The goal of this lab was to become more familiar with the tree data structure - more specifically a generic binary tree - and implement its common methods and structure into a program. The program allows user input to add items to the tree, and contains methods that would be of use in a binary tree. The classes are as follows :

BinarySearchTree, **BinaryTree**, **BinaryNode**. These contain the methods from the **Tree** interface.

Unit Tests:



- BlueJ unit test window output

Required Output:

```
Dylans-Macbook-Pro:lab7 DylanMaloy$ java Controller 1 5 3 23 5
1 added to the tree
5 added to the tree
3 added to the tree
23 added to the tree
5 could not be added to the tree because it is a duplicate
Tree contains 1 = true
Tree contains 2 = false
Number of elements at depth 2 = 2
Largest Element = 23
Smallest Element = 1
Inordertraversal = 1 3 5 23
Preordertraversal = 1 3 5 23
Postordertraversal = 3 5 23 1
Empty = true
```

- Command line output of command "java Controller 1 5 3 23 5"

```
Dylans-Macbook-Pro:lab7 DylanMaloy$ java Controller 52 230 4 49 392 23 1 4 3
52 added to the tree
230 added to the tree
4 added to the tree
49 added to the tree
392 added to the tree
23 added to the tree
1 added to the tree
4 could not be added to the tree because it is a duplicate
3 added to the tree
Tree contains 1 = true
Tree contains 2 = false
Number of elements at depth 2 = 3
Largest Element = 392
Smallest Element = 1
Inordertraversal = 1 3 4 23 49 52 230 392
Preordertraversal = 52 1 3 4 23 49 230 392
Postordertraversal = 1 3 4 23 49 230 392 52
Empty = true
```

- Command line output of command "java Controller 52 230 4 49 392 23 1 4 3"

Trouble Report:

This section is not applicable because all of my methods work as intended.

References:

Lysecky, R. (2019). Data Structures Essentials. ZyBooks.