CS 62 Quiz

Binary Search Tree Median

Mawhorter Spring 2017

(Please write your name at the bottom of the back page of this quiz)

(Don't forget part 2 on the back.)

1. Given a binary search tree, one operation that might be useful is a method for finding the *median* element (the element that would be in the middle if the items in the tree were in a sorted list). The code below implements just such an operation non-recursively; fill in the blanks so that it correctly returns the median element (you can assume that there are an odd number of elements in the tree)

(Recall that a ***`BinaryTree`*** *supports methods inculding* ***`left()`*** *to get the left subtree,* ***`right()`*** *to get the right subtree,* ***`size()`*** *to return the size in O(1) time, and* ***`value()`*** *to return the value of a node.)*

```
public E findMedian(BinaryTree<E> tree) {
    int left = tree.left().size();
    // What is our target for the # of nodes on the left?
    // (it's okay if this only works for odd-sized trees)

    int target = ___________________________________;
    BinaryTree<E> here = tree;
    while (left != target) {
        if (left < target) {
            here = here.right();
            // update left: how many more nodes are now on the left?
            // note: be careful you're not off by 1

            left += ________________________________________;
        } else { // left > target
            here = here.left();
            // update left: how many nodes that were left are now right?
            // note: be careful you're not off by 1

            left -= ________________________________________;
        }
    }
    // what do we return?

    return _____________________________________;
}
```

2. What is the big-O run time of `findMedian` in terms of the size of the tree, n?

(Give one answer for the case where your binary tree is a stick, and another for the case where your binary tree is somewhat balanced (such that its height is O(log n*)). Are these the same or different?)*

Name: _____