

Lab 12
CSC 1052 - Algorithms and Data Structures II
Grading: 30 points
Due Date: April 28th, 2017

Description: In this lab, you will be coding new methods for binary search trees. The first method you will implement is a recursive minimum computation on a tree, `min2()`. You may look at the iterative version in the code to understand the functionality, but your implementation should use recursion and return the smallest data element in the tree.

The second operation you will be coding is a count of the number of leaves in the tree, `leafCount()`. For this method, you should develop an iterative (i.e., NOT recursive) method that returns an integer count of the number of leaves in the tree. Your implementation will use a similar algorithm to the iterative size method discussed in class, you just need to figure out what operations to perform when you visit each node. To implement the method, use the Java Stack class in `Java.util`.

The last operation you will be coding is a count of the height of the tree, `height()`. For this method, use recursion to determine the number of edges between the root of the tree and the deepest leaf node. As an example, in the tree pictured in Figure 1, the height of the tree is the number of edges between “hello” and “is”.

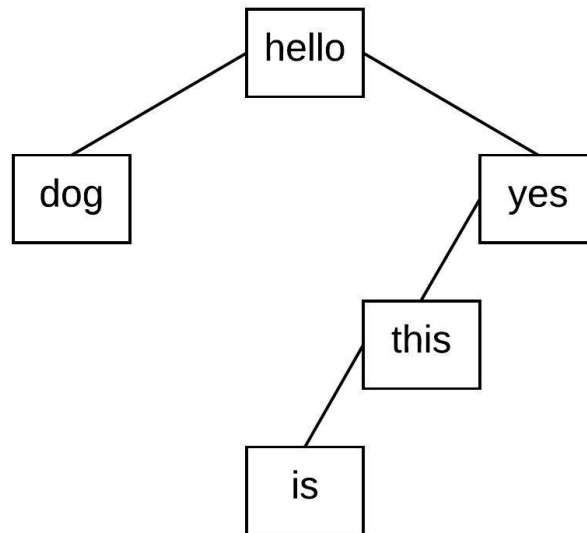


Figure 1: An example binary tree.

Use the template code provided online as a framework for this lab. All method headers are already implemented, you only need to fill in the methods where commented. You can use the included `Lab12.java` as test driver code to see if your program is working correctly. You will be editing the included file `BinarySearchTree.java`.

Rubric:
(5 points) Compiles without errors.

- (5 points) Recursive min correctly implemented.
- (10 points) Iterative leafCount correctly implemented with built-in stacks.
- (10 points) Recursive height operation correctly implemented.

Deliverables: Submit your modified `BinarySearchTree.java` on blackboard.