**CMSC 341 Spring 2017**

**Homework #4**

**Due: Thursday, April 6, 8:59:59pm**

**Instructions (same as before):** Save a copy of this Google Doc. Type in your answers. Save your document as PDF. Copy the PDF file to GL. Move the PDF file to the hw4 directory of your shared submission directories.

**Addendum:** A well written proof by induction must:

1. State the induction hypothesis
2. State what you are allowed to assume from the induction hypothesis
3. State what you need to show to establish the induction step
4. Clearly indicate which part(s) of the proof uses the induction hypothesis

Your proof by induction must include a narrative written in complete English sentences. In particular, a sequence algebraic manipulations without such a narrative does not constitute a proof and will not receive very much credit.

**Question #1:**

Suppose that you have a set of keys { *x*1, *x*2, *x*3, ..., *xn* } stored in a binary search tree *T*. Explain why an inorder traversal of *T* will produce the same listing of the keys regardless of whether *T* is a "normal" binary search tree, a red-black tree or an AVL tree.

Is this true for post-order traversal, too? Explain.

**Question #2:**

Let us define a *full node* in a binary tree to be a node that has two children. For a binary tree *T*, let *full*( *T* ) be the number of full nodes in *T* and let *leaves*( *T* ) be the number of leaves in *T*. Prove by **induction** on the number of nodes in a binary tree, that *leaves*( *T* ) = *full*( *T* ) + 1. Hint: consider the cases where the root has two children, one child and zero children.

Hypothesis:

*leaves*( *T* ) = *full*( *T* ) + 1

Base case:

full(t) = 0. This represents the root having no children. Here root is not a full(t), so it is also a leaf node and the tree only has one leaf node. Thus the equation *leaves*( *T* ) = *full*( *T* ) + 1 is translated to 1 = 0 + 1 which is true. This is also true for when the root has only a left or only a right child node. Here the full(t) is also 0, and while the root isn't a leaf, its child is. So the hypothesis can be written 1 = 0 + 1 which is true. Finally for the base case we can consider when root has two children, here counting the components of the tree we see the full(t) = 1 and the leaves(t) = 2. So the hypothesis becomes 2 = 1 + 1 which is true.

Induction:

Assume:

*leaves*( *T* ) = *full*( *T* ) + 1

Prove:

*leaves*( *T* ) + 1 = ( *full*( *T* ) + 1 ) + 1

Substitute leaves(T) for the hypothesis:

( *full*( *T* ) + 1 ) + 1 = ( *full*( *T* ) + 1 ) + 1

**Question #3:**

In Project 3, you are implementing a weight-balanced binary search tree where you guarantee that at every node of the binary search tree, the size of the left subtree is less than or equal to twice the size of the right subtree, and vice versa. Let us define *size*( *x* ) to be the number of nodes in the subtree rooted at node *x* (including *x*). Then the condition that you are guaranteeing is that for all nodes *x* in *T*, *size*( *xL* ) ≤ 2 *size*( *xR* ) and *size*( *xR* ) ≤ 2 *size*( *xL* ) where *xL* and *xR* are, respectively, the left and right child of *x*.

Prove by **induction** on *n* that the LazyBSTs in Project 3 are also height balanced by showing that every subtree in a LazyBST with *n* nodes has height less than or equal to 3 log *n*.