September 16, 2019

Computer Science 282
Programming Assignment #2

Due: September 30, 2019

In addition to requiring you to get AVL trees to work, I also want you to learn some basic things about Java Strings. For example, you will need to use compareTo (or compareToIgnoreCase – we will need to discuss which) to compare two Strings.

Like last time, add whatever is necessary so that the two classes below behave correctly. Leave all variables, method names, and code fragments exactly as they appear. Only add code to make the methods work. Do not add a setItem method.

You might want to write a simple toString method to help with your debugging but I have provided a "display" method in my test program that you might also find helpful in your test programs. With a little study you should be able to figure out what it is doing and how it is doing it.

A simple test program has already been posted, mainly for the purpose of helping you get the basic syntax going. A more comprehensive test program will be posted later that will include tests for all additional functions. Do not use my test programs for debugging your programs, though feel free to use the first test program as a model for yours. Using the second test program for debugging, when it appears, would be a mistake. You need a test program that provides much more information to help you track down your problems.

Work on the insert method and getting debugging apparatus up and running first. Then work on everything but delete. Save delete for last because it is the hardest and there is a chance that I may make it optional. When programming delete choose the replacement node to be the largest in the left subtree, not the smallest in the right subtree.

You should email me your Java classes in a single file named prog2.java. Do not email me your output, do not email me the test program, and be sure the file has exactly two classes in it. The subject of your email should be "Magnus Carlsen – prog 2 – 2:00" if your name is Magnus Carlsen and you are in the 2:00 class. Adjust your name and class time accordingly. The usual warning applies about following all instructions.

```
class StringAVLNode {

   private String item;
   private int balance;
   private StringAVLNode left, right;

   // just one constructor, please
   public StringAVLNode(String str) {

   public int getBalance() {

   public void setBalance(int bal) {

   public String getItem() {

   // no setItem

   public StringAVLNode getLeft() {

   public void setLeft(StringAVLNode pt) {

   public StringAVLNode getRight() {

   public void setRight(StringAVLNode pt) {

} // StringAVLNode
```

```
class StringAVLTree {

   // should really be private but I need access
   // for my test program to work
   StringAVLNode root;

   // just one constructor
   public StringAVLTree() {


   // Rotate the node to the right
   private static StringAVLNode rotateRight(StringAVLNode t) {

   // Rotate the node to the left
   private static StringAVLNode rotateLeft(StringAVLNode t) {


// For these next four, be sure not to use any global variables
// and no extra “counting” parameters in the recursive methods, e.g.,
// the recursive height method should just have one parameter, the
// StringAVLNode

   // Return the height of the tree - not to be used anywhere in insert or delete
   public int height() {

   // Return the number of leaves in the tree
   public int leafCt() {

   // Return the number of perfectly balanced AVL nodes
   public int balanced() {

   // Return the inorder successor, i.e., the next larger value in the tree
   // or null if there is none or str is not in the tree
   public String successor(String str) {


   public void insert(String str) {

   private StringAVLNode insert(String str, StringAVLNode t) {

   public void delete(String d) {

   private StringAVLNode delete(StringAVLNode t, String d) {

   // who are you? Put your name here!
   public static String myName() {
       return "Magnus Carlsen";
   }

}  // end of StringAVLTree class
```