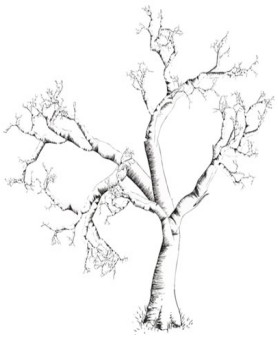
CAS CS 112A1 – Spring 2008, Assignment 4

Programming part due at 10:00 pm on Thursday, March 27

Written part due at the beginning of class on March 27



Binary Search Trees

In this assignment, you will augment the Binary Search Tree implementation presented in class by adding functionality to insert multiple items, remove the leafs, and graphically display the tree. The assignment consists of the following components:

1. Start from the Binary Search Tree implementation of Chapter 4.3. You can find the source code on the class website or here:

   `http://www.cs.fiu.edu/~weiss/dsaajava2/code/`.

2. (15 points) First, add functionality to support bulk insertion of integer values. Insertions will be specified in the form of a text string in JSON (JavaScript Object Notation). The JSON string "[12, 15, 76, 23, 99]" should result in an insertion of 12, followed by an insertion of 15, etc. Implement this as a member function **public void BulkInsert(String s)**. Note: the **split** function of String will be useful to you in accessing the comma-seperated items.

3. (25 points) Next, add functionality to remove all the leafs from the tree by implementing a function **public void removeLeafs()**. This should be very straightforward if you follow the spirit of the **remove** function.

4. (40 points) Then you will add functionality to graphically display the tree. In order to accomplish this, you will logically assign an (x, y) coordinate to each tree node. It is appropriate for the x-coordinate of the node to be proportional to the inorder traversal number of the node in the tree, and for the y-coordinate of the node to be proportional to the (negative of) the depth of the node in the tree. Therefore, you will need to add member variables to store, and routines to compute the inorder traversal number and depth of each node in the tree (you may choose what to name these). Then write a function **public void displayTree()** that displays the tree in a pop-up window.

   The pop-up window will be a Java Swing object, you will see an example of creating one in lab, call it **displayWindow.java**. To display the tree, you will need to add

a **public void addGraphics(Graphics G)** method to your BST implementation which will use the coordinates of each node together with the **DrawCircle** and **DrawLine** methods of **Graphics** to "draw" the tree. You will call **addGraphics(Graphics G)** from within the **paint()** method of **displayWindow**. Then your **displayTree()** method will simply create and make visible a **displayWindow** object (as the main in the lab does).

You will recieve ten points each for correctly computing the depth and inorder traversal number of each node, and 20 points for implementing the graphical display.

5. Finally, you will need to write a main function that tests the routines you have written, call it **driver.java** Your code should repeatedly query the user for a JSON string corresponding to a sequence of insertions, create a BST tree resulting from that sequence of insertions, remove the leafs, and launch a window that displays the tree.

Extra Credit

(20 points) Write a function **public void PrintAsText()** that prints a BST to screen. For example, printing the tree resulting from the bulk insertion above should look something like:

```
12
     15
         76
       23   99
```

Hint: you will need to traverse the nodes level-order to accomplish this.

Submissions

Submit your code in three files called BST.java, displayWindow.java and driver.java.

Written questions

1. (10 points) Show by induction that the number of degree-2 nodes in any nonempty binary tree is 1 less than the number of leaves (use induction on the height of the tree).

2. (10 points) Draw step by step the AVL tree that results when you insert items with keys E, A, S, Y, Q, U, T, I, O, N in that order into an initially empty tree.