CS 400 Lecture 3:

AVL Review,
Nested Classes,
Red-Black Trees

Andy Kuemmel, Instructor

Agenda for Tonight

5:30	Discuss Homework 0 …
5:40	Review of AVL Trees: Balancing and Upkeep
6:10	X-Team activity
6:20	Break
6:35	Notes: Red-Black Trees
7:36	Break
7:40	Nested Classes
7:50	Putting it all together: AVLTree.java
8:00	Starting Program 2a: AVLTree.java
8:30	Adjourn

Homework 0:
answers should be available on Canvas tomorrow morning

Review of AVL Trees

Visualizer Web Site

Worksheet

Notes about AVL Trees

Named after George Adelson-Velsky and Evgenii Landis

Complexity Analysis:
Insert, Delete, Lookup all are O(Height)
They proved that the height worst-case is about
1.44 * log2(N)

There must be Balance in the Force

```
private int getBalance(Node n){
     return n.getRight().getHeight() - n.getLeft().getHeight();
}

private int getHeight(Node n) {
     return recursiveHeight(n);
     // OR have an instance variable for height for each node
     // and constantly update this on every rotation/insert/delete
}

private int recursiveHeight(Node n){
     if (n == null)
          return 0;
     return 1 + Math.max(recursiveHeight(n.getLeft())
                         + recursiveHeight(n.getRight()));
     }
}
```

Algorithm for AVL Insert

1.) use the recursive insert code from a Binary Search Tree

2.) update the height of this node by looking at height of left and right

3.) calculate the balance factor of this node

4.) check the four cases for rebalancing

```
if ( getBalance() >=2   && just inserted a value to right of this node)
        // Right-Right case
if ( getBalance() >= 2  && just inserted a value to the left of this node)
        // Right-Left case
if ( getBalance() <=-2   && just inserted a value to right of this node)
        // Left-Right case
if ( getBalance() <=- 2  && just inserted a value to the left of this node)
        // Left-Left case
```

AVL Tree Delete?

0. Try some examples based on the Worksheet
1. Perform a BST Delete (see code from previous lecture)
2. Update the heights and balances.
3. Determine what rotations are needed.
4. There may be cascading rotations that need to be checked.

X-Team Activity

We will FORM teams tonight for in-class and out-of-class assignments during the semester.

These will not be your final project teams. You will choose your team members for the final project.

Break until:
6:35

Notes: Red-Black Trees

Red-Black Trees: Reference

Visualizer:

CS department's summary page

Break Until:
7:40

Nested Class:

A class that is defined within another class.

Both the nested and outer classes have access to the other's private methods and fields.

Increases Encapsulation (hiding details)

Inner (Nested) Classes

Reading: Oracle's Java Tutorial: Nested Classes

Example: AVLTree with an inner class