CS251 Data Structures – Fall 2011
Homework 6 – 80 points
Due: Nov. 22

1. (10 points)
   (a) Write a recursive method `int numFound(Object x)` which returns the number of times that the item `x` occurs in a binary tree (NOTE: not a binary search tree). This method should be a non-static method in the `BinaryNode` class.
   (b) Write a static version of the same function

2. (15 points) Assume we insert the following values into an initially empty binary search tree: 6, 2, 5, 4, 11, 12, 9, 8, 1, 7, 3, 10
   (a) Show the resulting tree after the 4, 8 and 10 have been added.
   (b) Give the pre-, in- and post-order traversals of the final tree.
   (c) What is the height of the final tree?
   (d) Show the two possible trees that can result from deleting the root value from the tree.

3. (10 points) Assume we insert the following values into an initially empty AVL search tree: 6, 2, 5, 4, 11, 12, 9, 8, 1, 7, 3, 10. Show the tree prior to and after any rotation.

4. (15 points)
   (a) Show what a (2,4) tree would look like after inserting the values 1 through 13 in order.
   (b) Assume we now delete the odd values 1, 3, 5, ..., 13. Show what the tree looks like after any transfer or fusion operation. Each deletion should use the tree from the previous deletion.

5. (10 points) Write a `print()` method for the (2,4) node class which outputs the values of the tree rooted by the receiver of the `print()` call in sorted order. You may assume the following definitions:

```
public class TwoFourNode<T>
{
   <T> [] vals;
   TwoFourNode<T> [] children;
   int numVals;
}
```

6. (15 points)
   (a) Give examples of AVL trees of height 1, 2, and 3 which have the minimum number of nodes.
   (b) Derive a recurrence relation $T(h)$ for the minimum number of nodes in an AVL tree of height h.
   (c) Solve this recurrence relation and show that for large n it implies that the maximum height for an AVL tree with n nodes is approximately $1.44 \log(n+1) - 1.328$.

7. (5 points) Suppose we had a (2,4) tree of height 3 and after deleting a value we had to use fusion operations all the way to the root (resulting in a new tree of height 2). What could be the minimum number of values in the original tree? What could be the maximum number of values?