CS3610/5610N

Homework #2: (120 points, 5 final pts.)

due date: March 17, Wednesday 11:59pm

Midterm exam will be held on March 23, Tuesday.

1. (Binary Tree, 15pts) Read through the `binaryTreeType` class defined on pages 609-616 and 628-632. Add a recursive function, `leafCount()`, that returns the number of leaf nodes in a binary tree. Convert this function to a non-recursive version `nr-leafCount()`. Turn in your source code for these two functions.

2. (BT, 10pts) Exercise 14, page 679.

3. (BT, 10pts) Exercise 15, page 679.

4. (BT, 10pts) Exercise 16, page 679.

5. (AVL tree, 10pts) Use a diagram to explain when a right-left double rotation (right rotation followed by a left rotation) is needed in AVL **deletion**, and how it's conducted.

6. (AVL tree, 5pts) Exercise 19, page 680.

7. (AVL tree, 5pts) Exercise 20, page 681.

8. (BST, 10pts) In Chapter 10, we will see that every comparison-based algorithm to sort n elements must take $\Omega(n\log n)$ comparisons in the worst case. With this fact, what would be the complexity of constructing a n-node binary search tree and why?

9. (Heap, 10pts) The following elements are inserted into an empty Max-Heap in the following order:

   2, 3, 1, 4, 6, 12, 15, 22, 11, 5

   Draw the resulting heap (use the logical (tree) representation).

10. (Build Heap, 10pts) Exercise 12, page 595.

11. (Heapsort, 10pts) Exercise 13, page 595.

12. (Heap, 5pts) Draw all legal Max-Heaps containing the 4 elements 1, 2, 3, 4.

13. (Heap, 5pts) Draw all legal Max-Heaps containing the 5 elements 1, 2, 3, 4, 5.

14. (Recursion 5pts) An implementation of the n-queen puzzle is attached under Project 3 assignment. Compile and test it for $n = 1, 2, 3, ...10$. Turn in the number of solutions for each n (e.g., the number of solutions for $n = 8$ is 92).