CS 223
Laboratory Assignment #3
Due: at the end of the lab in Week 11 (Mar 30, 07)

The Problems

1. Exercise 12.2-4 (page 260 of CLRS).

2. Exercise 12.2-9 (page 260 of CLRS).

3. Given any n-node binary tree T, show that one can always use at most $n - 1$ right rotations to transform T into a right-going path.

4. In class, we covered Red-Black trees. In this lab assignment you need to implement it. You do not have to start from scratch, and you can start from the *BinarySearchTree* class we have been using since last year's CS223 (`http://www.cs.montana.edu/courses/spring2006/223`). You need to implement the following 8 functions:

insert — inserts a node into the red-black tree.
delete — deletes a given node from the red-black tree.
search — given a key value, returns a node from the tree with the same key, or report no such node exists.
printTree — you can use the existing method in BinarySearchTree, extra credit will be given if you print red nodes in *red* and black nodes in *black*.
treeHeight — returns the height of the tree.
treeBlackHeight — returns the black height of the tree.
treeMinimum — returns the smallest key value in the tree.
treeMaximum — returns the largest key value in the tree.

Test runs: While you can try your own datasets, we need the following two test results from you:

(1) Insert the following numbers into a new red black tree: 5,10,9,1,8,6,4,12,20,25,35,11,30; print out the tree. Then delete 10,20,30 and print out the tree again.

(2) Insert the numbers 25,24,...,1 into a new red black tree. Call **treeHeight, treeBlackHeight, treeMaximum** and **treeMinimum** on this tree. Print out the results of these calls.

Then call **delete(search(11)),delete(search(13)),...,delete(search(21))** on the search tree. Print out the resulting tree.