Goals: To familiarize you with manipulation of trees.

Getting Started: Copy the files for Homework 3 from the directory $master/hw/hw3.

Exercises: As you complete the following exercises, place the code you write within the specified files. The files you copied contain framework necessary for this assignment. Do not change their structure except where instructed by comments in the framework.

1. (Binary Search Trees) Complete the following methods, placing your code in the provided file BST.java.
   a. Write a method void insert(TreeNode n) that inserts the given TreeNode into this BST by its key field. If a TreeNode with an equal key already exists, replace the existing TreeNode with n.
   b. Write a method TreeNode search(Comparable key) that searches this BST for the TreeNode with the given key and returns it. If no such TreeNode exists, return null. Your implementation should run in O(h) time.
   c. Write a method void remove(Comparable key) that removes the existing TreeNode that has the given key, if it exists.
   d. Write a method boolean isBalanced() that returns true if the difference in height between the left and right subtrees of this BST is one or zero and the left and right subtrees are themselves balanced.
   e. Write a method boolean isComplete() that returns true if this BST is a complete tree. Your implementation should run in O(n) time.