

CSCI-1200 Computer Science II — Spring 2006

Lab 13 Binary Search Trees and Recursion

This lab explores binary search trees and their implementation in the `cs2set` class, along with the use of recursive functions to manipulate binary search trees. Once you have a basic understanding of trees, the actual code that you have to write for the lab is quite short. Review the notes from Lectures 21 and 22 prior to starting.

Download the files:

http://www.cs.rpi.edu/academics/courses/spring06/cs2/labs/13_trees/cs2set.h

http://www.cs.rpi.edu/academics/courses/spring06/cs2/labs/13_trees/test_cs2set.cpp

Then, turn off all network connections.

Examine the code in `cs2set.h` and `testset.cpp`. The former contains the implementation discussed in Lectures 21 and 22, including the functions implemented in lecture as exercises. The latter contains code for testing the implementation. Some of this has been commented out because all the functions have not yet been implemented.

Checkpoint 1

The implementation of `find` provided in `cs2set.h` is recursive. Implement and test a non-recursive replacement for this function.

To complete this checkpoint: Show one of the TAs your new code. Be prepared to discuss the running time for the two different versions of `find` for various inputs.

Checkpoint 2

The implementation of the copy constructor and the assignment operator is not yet complete because each depends on a private member function called `copy_tree`, the body of which has not yet been written. Write `copy_tree` and then test to see if it works by “uncommenting” the appropriate code from the main function.

To complete this checkpoint: Show one of the TAs your new code and some new test cases you’ve written to exercise the copy constructor, the assignment operator, and the destructor.