

# CSCI-1200 Data Structures — Fall 2024

## Lab 10 — Binary Search Trees & `ds_set` Implementation, part I

### Checkpoint 1

Checkpoint 1 will be available at the start of Wednesday's lab.  
*It will be a team-of-two paper & pencil worksheet to be completed  
with one other person from your lab section.*

### Checkpoint 2

*estimate: 10-15 minutes*

Now let's explore the implementation of the `ds_set` class, along with the use of recursive functions to manipulate binary search trees. Download and examine the files:

[http://www.cs.rpi.edu/academics/courses/fall24/csci1200/labs/10\\_trees\\_I/ds\\_set.h](http://www.cs.rpi.edu/academics/courses/fall24/csci1200/labs/10_trees_I/ds_set.h)

[http://www.cs.rpi.edu/academics/courses/fall24/csci1200/labs/10\\_trees\\_I/test\\_ds\\_set.cpp](http://www.cs.rpi.edu/academics/courses/fall24/csci1200/labs/10_trees_I/test_ds_set.cpp)

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

**To complete this checkpoint:** Show one of the TAs your new code and variety of tests. Be prepared to discuss the running time for the two different versions of `find` for various inputs – what is the best case, worst case, and average case?

### Checkpoint 3

*estimate: 10-20 minutes*

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. What is the Big O Notation of `copy_tree` for a tree with  $n$  nodes and height  $h$ ? Does it matter if the tree is balanced or unbalanced?

**OPTIONAL: Bring your finished or nearly finished Homework 7  
Data Structure Diagram to lab discuss the design with a TA/mentor  
to receive a 1 day deadline extension on Homework 7.**