

CSCI-1200 Data Structures — Fall 2025

Lab 11 — Trees & ds_set Implementation, part II

Checkpoint 1

estimate: 40-60 minutes

Checkpoint 1 will be available at the start of Wednesday's lab.

Checkpoint 2

estimate: 15-30 minutes

Download these files:

http://www.cs.rpi.edu/academics/courses/fall25/csci1200/labs/11_trees_II/ds_set.h

http://www.cs.rpi.edu/academics/courses/fall25/csci1200/labs/11_trees_II/test_ds_set.cpp

Implement and test the decrement operator for `tree_iterator`. Determine an appropriate sequence to *insert* the numbers 1-15 such that the resulting tree is *exactly balanced*. After using the `print_sideways` function to confirm the construction of this tree, test your iterators on the structure. Similarly, create a couple unbalanced trees to demonstrate that both the increment and decrement operators for iterators are debugged. Your decrement operator should correctly decrement the `end()` iterator. You can use the same “trick” we used in Lab 7 to make this work for `ds_list` iterators. Ask a TA or mentor if you have any questions.

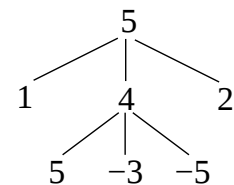
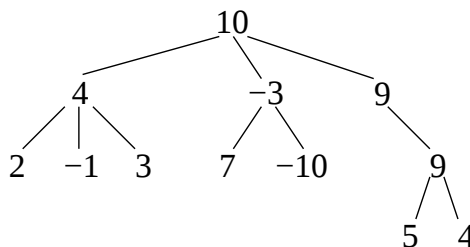
To complete this checkpoint: Show your TA or mentor your iterator decrement code and your test cases.

Checkpoint 3

estimate: 15-30 minutes

A *trinary tree* is similar to a binary tree except that each node has at most 3 children. Write a *recursive* function named `EqualsChildrenSum` that takes one argument, a pointer to the root of a trinary tree, and returns true if the value at each non-leaf node is the sum of the values of all of its children and false otherwise. In the examples below, the tree on the left will return true and the tree on the right will return false.

```
class Node {
public:
    int value;
    Node* left;
    Node* middle;
    Node* right;
};
```



If time allows... type up your solution, write a simple main function that constructs the sample trees above and then calls your function. Compile, test, & debug your implementation.

To complete this checkpoint: Explain your implementation to a TA or mentor.