

Worksheet 28: Binary Search Trees

Group 20 members:

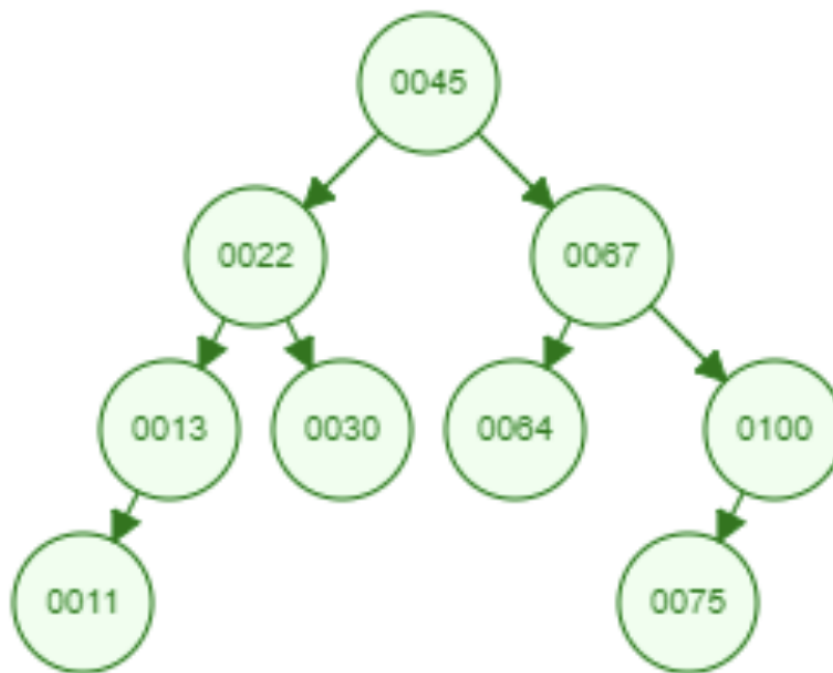
Edward Francis
Danielle Goodman
Justin Kruse
Charlotte Murphy
Garret Sweetwood
Woohyuk Yang

francise@oregonstate.edu
goodmada@oregonstate.edu
krusej@oregonstate.edu
murpchar@oregonstate.edu
sweetwog@oregonstate.edu
yangwo@oregonstate.edu

In Preparation: Read Chapter 8 to learn more about the Bag data type, and chapter 10 to learn more about the basic features of trees. If you have not done so already, read Worksheets 21 and 22 for alternative implementation of the Bag.

In this worksheet we will practice the concepts of using a Binary Search Tree for the Bag interface. For each of the following problems, draw the resulting Binary Search Tree.

1. Add the following numbers, in the order given to a binary search tree. 45, 67, 22, 100, 75, 13, 11, 64, 30

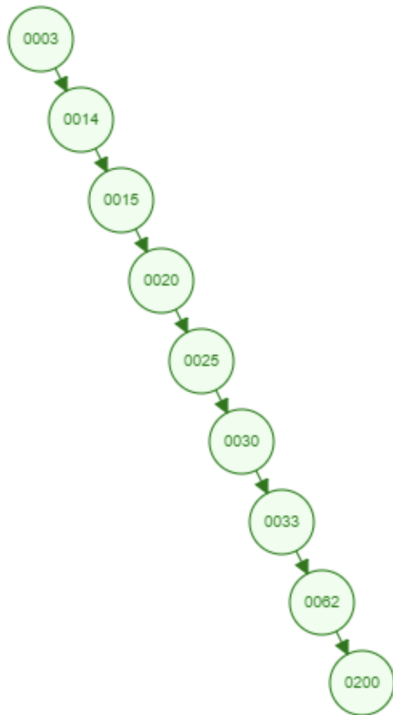


2. What is the height of the tree from #1? 3 (maximum arcs from root to leaf)

What is the height of the subtree rooted at the node holding the value 22? 2 (same approach, but viewing the subtree in isolation)

What is the depth of the node holding the value 22? 1 (distance from root)

3. Add the following numbers, in the order given to a binary search tree. 3, 14, 15, 20, 25, 30, 33, 62, 200.



4. Is the tree from #3 balanced? Why not?

No, because the height difference between the left branch and the right branch is greater than 0 (for all nodes).

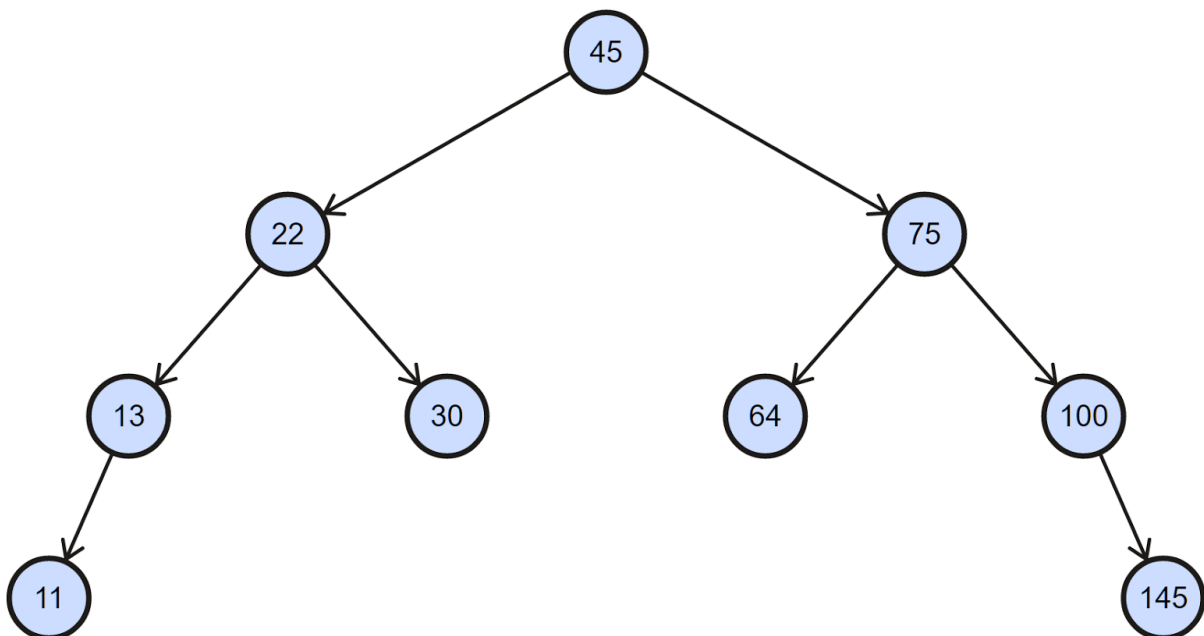
What is the execution time required for searching for a value in this tree?

$O(n)$ (based on Chapter 10 readings; structure is equivalent to a linked list)

5. Add a new value, 145, to the tree from #1



6. Remove the value 67 from the tree from #1. What value did you replace it with and why?



The node containing 67 is replaced with 75, because 75 is the lowest value of the right subtree of the node to be removed.

<http://btv.melezinek.cz/binary-search-tree.html>