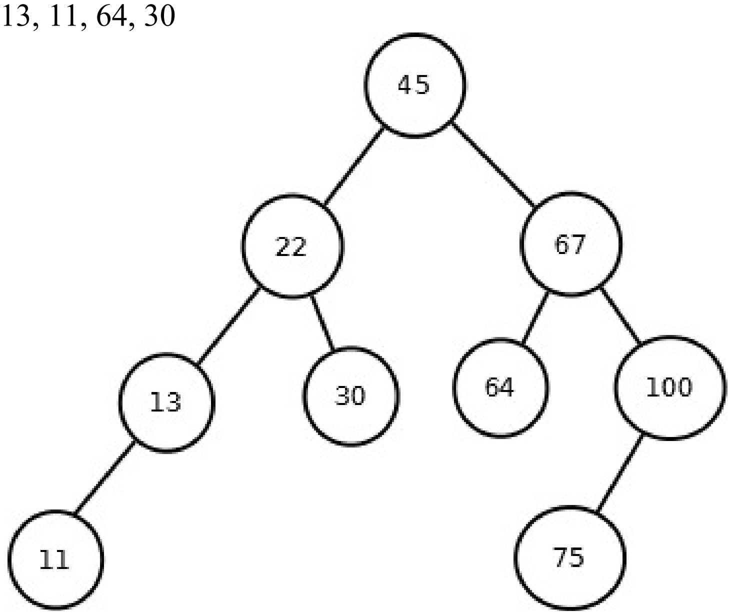
Name: Leonard Harold, Toan Ngo, Eric Rouse, Jeff Toy

Worksheet 28: Binary Search Trees

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

   What is the height of the subtree rooted at the node holding the value 22?

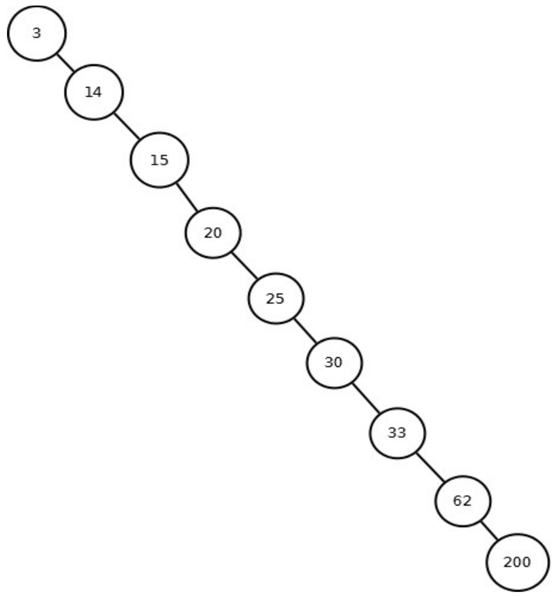
   2

   What is the depth of the node holding the value 22?

   1

Name: Leonard Harold, Toan Ngo, Eric Rouse, Jeff Toy

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?

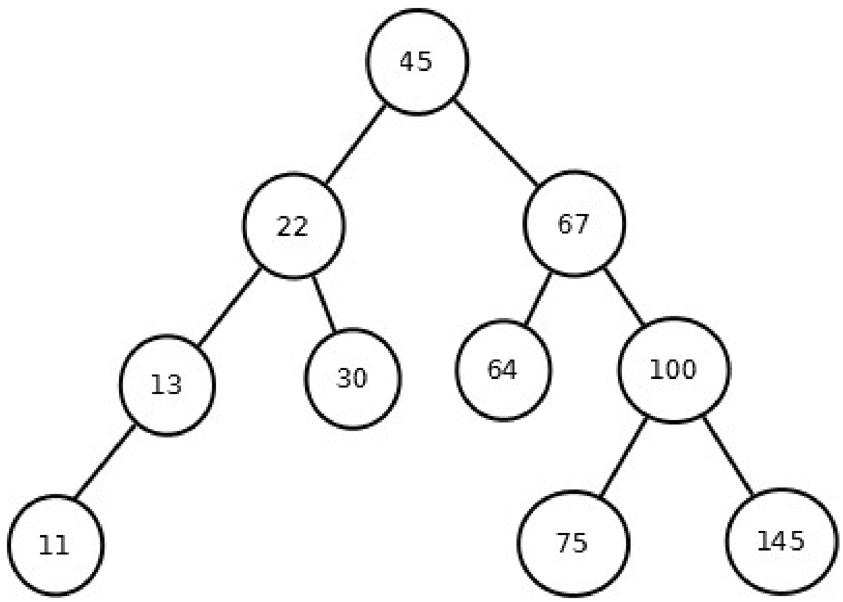
   No, because adding the elements in order caused them to be inserted on the right branches, leaving the "tree" more like a linked list.

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

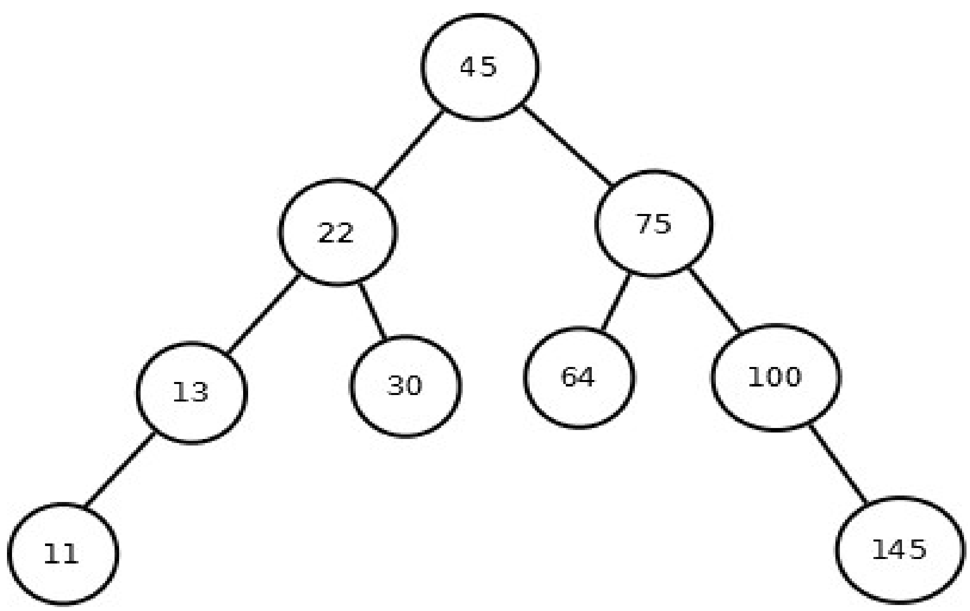
   O(n)

Name: Leonard Harold, Toan Ngo, Eric Rouse, Jeff Toy

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?



I replaced 67 with 75 because it was the leftmost leaf of the right side of the subtree of 67.