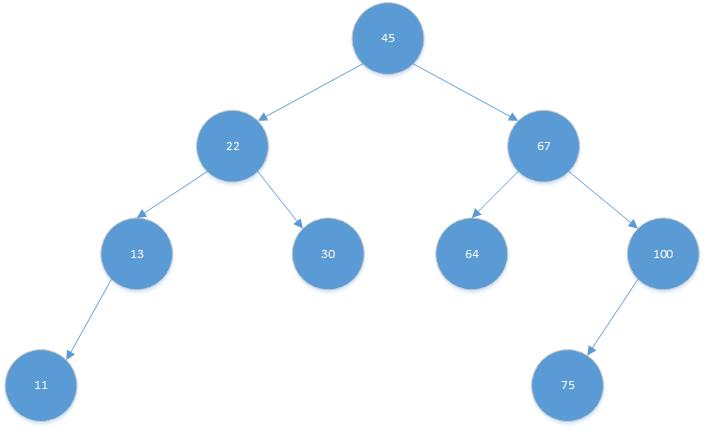
# Worksheet 28: Binary Search Trees

**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.

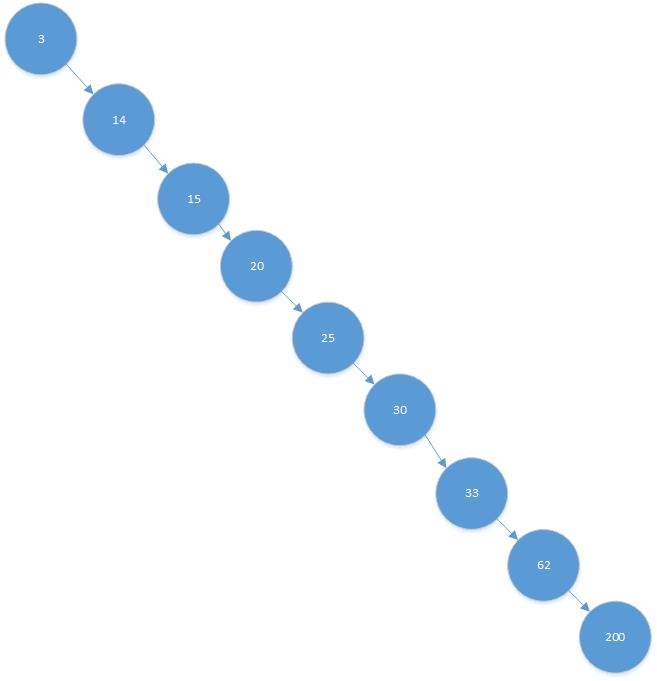
**1.**Add the following numbers, in the order given to a binary search tree.  45, 67, 22, 100, 75, 13, 11, 64, 30



**2.**What is the height of the tree from #1?  What is the height of the subtree rooted at the node holding the value 22?  What is the depth of the node holding the value 22?

1. Height is 3.
2. Height for the node holding the value 22 is 2.
3. The depth is 1.

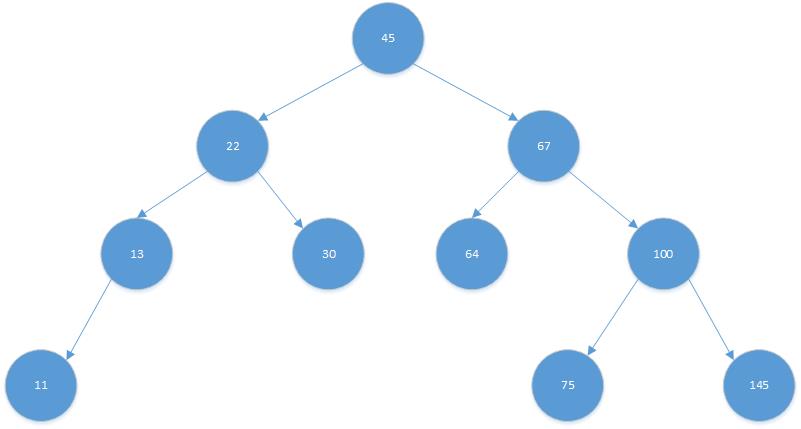
**3.**Add the following numbers, in the order given to a binary search tree. 3, 14, 15, 20, 25, 30, 33, 62, 200.



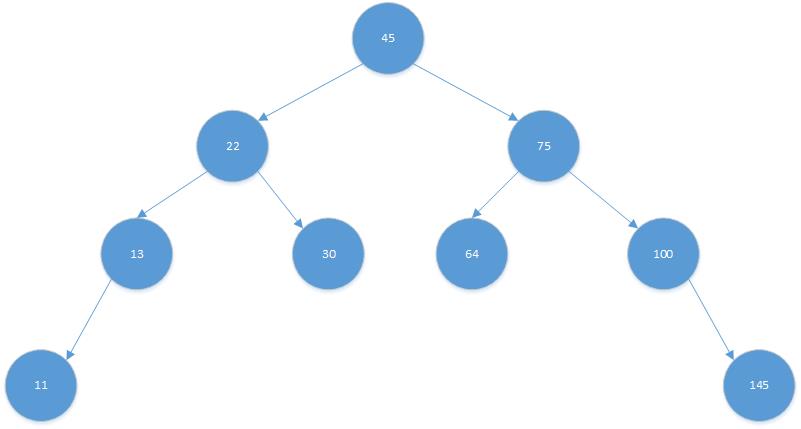
**4.**Is the tree from #3 balanced?  Why not?  No, the tree is not balanced because there is not an equal distribution of nodes on each side of the root node. This acts more like a linked list due to all the elements being added in order and having to keep branching to the right.

What is the execution time required for searching for a value in this tree? O(n)

**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 67 was replaced with 75. The left most child of the right most sub tree will be less than all of the nodes in the right most sub tree but less than the nodes

but greater than the nodes in the leftmost subtree?