# 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

45

22 67

13 30 64 100

11 75

1. 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?

height: 4

height of 22: 3

depth of 22: 1

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

3

14

15

20

25

30

33

62

200

1. Is the tree from #3 balanced? Why not? What is the execution time required for searching for a value in this tree?

It is not balanced. All of the nodes are on the right side of the tree. The time for searching will be O(n)

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

45

22 67

13 30 64 100

11 75 145

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

It is replaced with 75 since it is the left most value of the right child.

45

22 75

13 30 64 100

11 145