# 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

67

22

30

64

13

100

75

11

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?
   1. Height from 11 = 3
   2. Height from 22 = 2
   3. Depth of 22 = 1
2. 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?
   1. No, there is only node per “level” of the tree. The execution time is O(n)
2. Add a new value, 145, to the tree from #1

45

67

22

30

64

13

100

145

75

11

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

Replace 67 with 75. It is the leftmost node of the subtree rooted in the right branch node of 67.

45

75

22

30

64

13

100

145

11