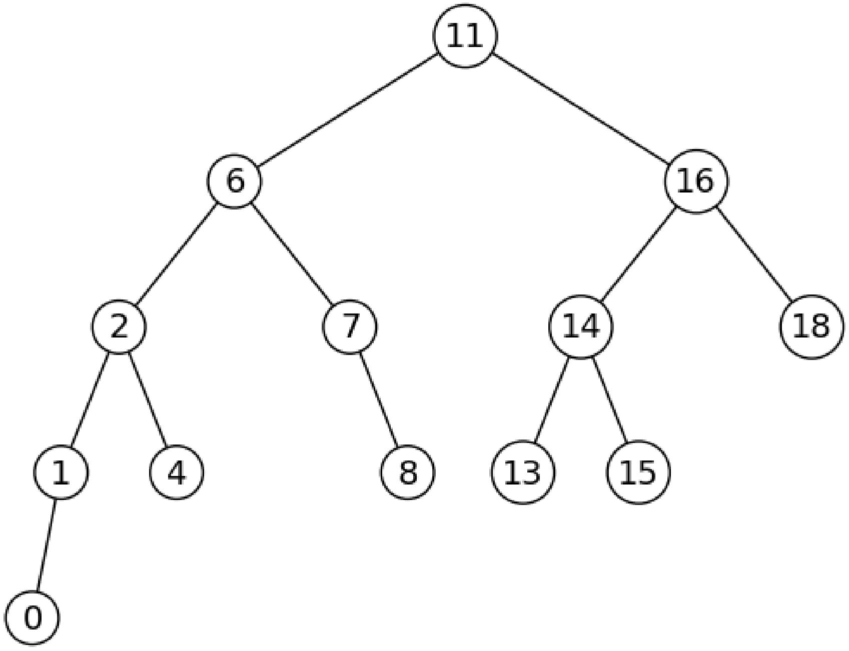
CS2302 - Data Structures

Fall 2020

Exercise - Binary Search Trees

1. The class BST included in the file *bst.py* implements basic binary search tree operations. Looking at the code, answer the following questions:
   (a) What are the data attributes of an object of class BST?
   (b) What are the data attributes of an object of class BSTNode?
   (c) Can we determine the number of nodes in a BST object in $O(1)$ time (that is, without traversing the tree)?
   (d) Why do we need separate BST and BSTNode classes?

2. Suppose T is the BST object that appears in the figure.
   (a) What is the value of *T.root*?
   (b) What is the value of *T.size*?
   (c) What is the value of *T.root.left.data*?
   (d) What is the value of *T.root.right.right.left*?
   (e) What is the value of *T.root.right.right.data*?



3. Write the function *smallest(t)* that receives a reference to a binary search tree and returns the smallest item in the tree.

4. Write the function *largest(t)* that receives a reference to a binary search tree and returns the largest item in the tree.

5. Write the function *sum_bst(t)* that receives a reference to a binary search tree and returns the sum of the elements in the tree.

6. Extra credit: Write the function *print_by_level(t)* that receives a reference to a binary search tree and prints the data in the tree ordered by depth. Thus if T is the BST object in the figure, *print_by_level(T)* would print 11 6 16 2 7 14 18 1 4 8 13 15 0. Hint: use a queue.