

## CS 111 Lab 13: Binary Search Trees

20 points

Goal: Test your understanding of Binary Search Trees

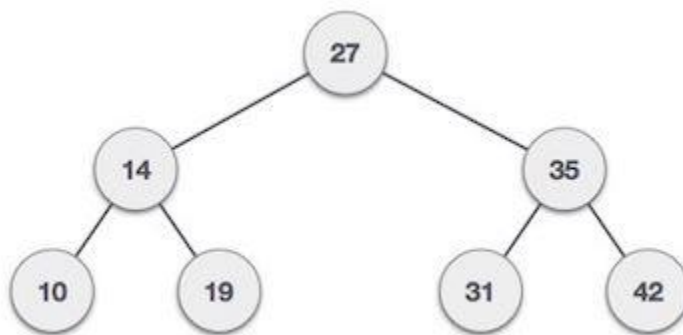
### In this lab you will be implementing Binary Search Tree

Binary Search Tree is a node-based binary tree data structure which has the following properties:

The left subtree of a node contains only nodes with keys lesser than the node's key.

The right subtree of a node contains only nodes with keys greater than the node's key.

The left and right subtree each must also be a binary search tree.



Methods to implement:

**public boolean contains(int x)** : Checks if a int x is in the BinarySearchTree. Return true if x is in the tree, false otherwise.

**private boolean contains(BinaryTree<Integer> t, int x)**: Recursive private method to check for an Integer in a BST. Here, t is the BinaryTree to search and x is the Integer to search. Return true if x is in the tree, false otherwise.

**public void add(Integer x)** : Add a item in a BST. Add must enforce the BinaryTree to obey the laws of a BST. Here, x is the integer to add.

**private BinaryTree<Integer> add(BinaryTree<Integer> t, Integer x)** : Private method to add a item in a BST, recursively. Add must enforce the BinaryTree to obey the laws of a BST. Here, t is the BinaryTree to add to and x is the integer to add.

**public void addAll(Collection<Integer> collection)** : Adds a collection to the BinaryTree. This must still enforce the rules of the BST. Here, "collection" is the parameter.

**Additional requirements:**

- Name included at the top of your code
- Clean readable and commented code