# Binary Search Trees (BST)

## Objective

The objective of this lab is to understand Binary Search Tree implementation using linked list. Examine how nonlinear data structure can be stored in linear memory.

## Task

Create a binary search tree as shown below and implement insert, print and find methods. Use constructor as needed.

Algorithm to Create a binary search tree

1. Compare VALUE with root node N of tree
   - If VALUE <N, proceed to the left child of N
   - If VALUE >N, proceed to the right child of N
2. Repeat step 1. until N=null and assign it a new node with value VALUE.

```
class node<T> {
  T data;
  node<T> left;
  node<T> right;

  node(T d){
  data=d;
  }
}

public class BST<T extends Comparable<T>> {

 node<T> root;

public void insert(T key){   .....       } // insert key in a tree public void

traverse(node n) { ....} // print tree using Breadth first traversal

public void TraverseLNR(node n){ ….} // print tree using Inorder traversal

public node find(T key){ ... } // find key in a tree
```