

## Lab 3 – Binary Search Tree

### Aim

Understand Binary Search Tree (BST).

### Resources

All Java files you need are found on Learning Mall.

### BST Implementation

At this Lab session, we are going to implement Binary Search Tree in java. Please download the three classes from Learning Mall.

#### Task : Complete the BST

Please complete below methods in *IntBST* class. If you have already done *search()* method and *delete()* method at your own time, please ignore them and work on the other two methods.

*search()*:

- function to search for a given key.
- return boolean.

*delete()*:

- function to delete a given key in the tree.
- No return.

*printRange()*:

- function to print all the keys which in the given range [k1..k2]. Given two values k1 and k2 (where k1 < k2). Print all the keys of tree in range k1 to k2. i.e. print all x such that  $k1 \leq x \leq k2$  and x is a key of given BST. Print all the keys in increasing order.
- parameters: int k1, int k2.
- No return.
- Algorithm:
  - 1) If value of node's key is greater than k1, then recursively call in left subtree.
  - 2) If value of node's key is in range, then print the node's key.
  - 3) If value of node's key is smaller than k2, then recursively call in right subtree.

Test:

Use *IntBSTTest* class to test your methods.

Also, try to call *tree.root.print(1)* to see what is this method doing.

Extra:

*KthElement()*:

- function to print the k-th element in a BST.
- parameter: int k.
- No return.
- Algorithm: The idea is to do reverse inorder traversal of BST. The reverse inorder traversal traverses all nodes in decreasing order. While doing the traversal, we keep track of count of

nodes visited so far. When the count becomes equal to  $k$ , we stop the traversal and print the key.

- For example: When  $k=2$ , we need to print the 2nd largest key, which is second last element in inorder traversal and second element in reverse inorder traversal. We traverse given Binary Search Tree in reverse inorder and keep track of counts of nodes visited. Once the count becomes 2, we print the node.

Note: If you cannot finish all the tasks please do them as homework.