**BST and Min-Heap insertion (6 Points)**

Determine if the following tree is BST or Min-Heap then apply the insert() method

1. Insert 50
2. Insert 35

1. Insert 25

**Tree Implementation (4 Points)**

The following Java code shows the BST class that has the insert operation.

* In the test class, that has main method, ask the user to enter 10 elements then insert these elements into BST using insert operation.

public class **BST** <T> {

 BSTNode<T> root, current;

 /\*\* Creates a new instance of BST \*/

 public **BST**() {

 root = current = null;

 }

public boolean **insert** (int k, T val){

 BSTNode<T> p, q = current;

 if (findkey(k)){

 current = q; /\* findkey() has modified current \*/

 return false; /\* key already in the BST \*/ }

 p = new BSTNode<T>(k, val);

 if (empty()) {

 root = current = p; return true;}

 else {

 /\* current is pointing to parent of the new key. \*/

 if (k < current.key)

 current.left = p;

 else

 current.right = p;

 current = p; return true;}

}}