public boolean isBST(Node root, int min, int max)

{

if(root == null)

{

return true;

}

if(root.iData < min || root.iData > max)

{

return false;

}

return isBST(root.leftChild, min, root.iData) && isBST(root.rightChild, root.iData, max);

}

public boolean ISBST(BST tree)

{

return isBST(tree.root, Integer.***MIN\_VALUE***, Integer.***MAX\_VALUE***);

}

The time complexity of this algorithm is O(N). This algorithm visits every node in the tree once through recursion and compares its iData with the max and min, and takes the value with the recursive process to compare until the base case which is null. The number of operations is just N depending on the number of nodes.