Java implementation of algorithm to find a value in a binary search tree.

```
private BinaryTreeNode<T> find (T element, BinaryTreeNode<T> r) {
      if (r == null) return null;
      else {
            Comparable<T> comparableElement = (Comparable<T>)element;
             if (comparableElement.compareTo(r.element) == 0)
                  return r;
             else if (comparableElement.compareTo(r.element) > 0)
                        return find(element,r.right);
                    else return find(element,r.left);
       }
}
```

In the next page is an algorithm in pseudocode to insert a value into a binary search tree.

Algorithm insert(k, r)
Input: value k, node r of a binary search tree
Output: true if k was successfully added and false if not

```
newNode = new node storing k
if tree is empty then {
    set newNode as the root of the tree
    return true
}
if k == value at r then return false // no duplicates allowed
else if k < value at r then
            if r has no left child then {
                    set newNode as left child of r
                    return true
              }
            else return insert (k, left child of r)
if k > value at r then
            if r has no right child then {
                    set newNode as right child of r
                     return true
              }
            else return insert (k, right child of r)
```