//Thi Thanh Tra Kieu #26106651

Big O Analysis

1. Array search:

Input: array type int, value type int.

Output: true or false;

```
public static boolean arraySearch(int[]arr, int value) {
        boolean found = false;                          ===>O(1)
        for(int i = 0 ; i<arr.length; i++) {            ===>O(n)
                if(arr[i] == value) {
                        found =  true;                   ===>O(1)
                        break;
                }
        }
        return found;                                    ===>O(1)
}
```

T(n) = O(1) + O(n) + (O(n) + O(1) + O(1) = 3*O(n) + 3*O(1) = O(n).

2. AVL tree search

Input: data type int, node type Node.

Output: true or false;

```
public boolean search(int data, Node node) {
        if(node == null)
                return false;                            ===>O(1)
        else {
                if(data < node.data)
                        return search(data, node.left);     ===>O(logn)
                else if(data > node.data)
                        return search(data, node.right);    ===>O(logn)
                else {
                        return true;                     ===>O(1)
                }
        }
}
```

T(n) = O(1) + O(logn) + O(logn) + O(1) = 2*O(logn) + 2*O(1) = O(logn).