# Algorithms and Data Structures

Assessed Exercise

19/03/2018

Name – Martina Pironkova

Student ID – 2068174

**"Add element" algorithm :**

Set current node to root node

Instantiate parent node and int direction

Increment size

Loop :

If current node is null

Make a new node

Count

If the root is null

Set anoNode to the root

Else if the direction is less than 0

Set anoNode to be the parent's left node

Else

Set anoNode to be the parent's right node

Exit the Loop

If the current node was not null

Compare the element with current

Loop again

If the added element matches current

Increment the current and set its count

If the added element and current are not the same

Find the next position to move in the tree

### "Remove Element" algorithm:

Set current node to root node

Instantiate int direction

Decrement size

Loop:

If current node is null

The tree is empty, return

Else assign *direction* with the comparison between the current element and the element to be deleted

If the elements match

Decrement counter

Set the counter to the current

Return

If direction is less than 0

Set the current node to be the current's left node

Else

Set the current node to be the current's right node

### "Iterator" algorithm:

Create new stack

Set current node to root node

While current node is not null:

Add current node to stack

Set current node to current left node

If stack is not empty:

Set return node to stack.pop()

Set subtree node to return right node
While subtree node is not null :

    Add subtree node to stack
    Set subtree node to subtree right node

Return node