Segment Tree Psuedo Code

Creation/Insertion:

```
create_segTree(node,startIndex,endIndex){

        if node > last node in a full & complete segment tree
                return;

        if startIndex == endIndex
                return;

        if (node == 1 (the root node))
                segmentTree[node] = sum of startIndex to endIndex;

                segmentTree[node LeftChild] = sum of left side of initial value array;
                segmentTree[node RightChild] = sum of right side of initial value array;

        else
                segmentTree[node LeftChild] = sum of left side of initial value array;
                segmentTree[node RightChild] = sum of right side of initial value array;

        //Two recursive calls that split initial array in half
        create_segTree(node LeftChild, startIndex, (startIndex+endIndex)/2);
        create_segTree(node RightChild, ((startIndex+endIndex)/2) + 1, endIndex);


}
```

Summing/Searching:

```
findSegSum(node, startIndex, endIndex, leftIndex (of range), righIndex (of range)) {

if leftIndex → rightIndex is contained in startIndex → endIndex
        return the value at node

if leftIndex → rightIndex is completely out of startIndex → endIndex
        return 0;

//Two recursive calls that split initial array in half
findSegSum(node LeftChild, startIndex, (startIndex+endIndex)/2, leftIndex, rightIndex);
findSegSum(node RightChild, ((startIndex+endIndex)/2) + 1, endIndex, leftIndex, rightIndex);

//returns the sum of both recursive calls when they all hit base cases
```