Report for assignment 6

Vasudha Todi (14EC10059)

February 12, 2016

Interval Trees

1. **To create a mutually exclusive and exhaustive equal sized interval tree** This is a recursive function where the median node is taken as the root node at each step. The function is then called recursively for the left and the right sub-trees.

Algorithm 1 Creating an interval tree

```
createIntervalTree($l, u, n$)
**if** $l >= u$ **then**
    return NULL
**end if**
$step = (u - l + 1)/n$
store the median node as the root node
$root.left = createIntervalTree(l, tree.l - 1, n/2)$
$root.right = createIntervalTree(tree.u + 1, u, n/2)$
return tree
```

The complexity of the function is given by :
$T(n) = 2 * T(n/2) + O(1)$
By the master method,
$a = 1, b = 1$
The solution is given by :
$T(n) = n^{log_b a}$
hence,
$T(n) = O(n)$

2. **To merge an interval with the given interval tree** First, a node is created for the new interval. Then the first overlapping interval from the root is found out and its elements are inserted in the new node. Then the elements of the subsequents overlapping intervals are include and the tree is edited accordingly.

Algorithm 2 Merge an interval with the tree

```
merge$(tree, l, u)$
Create a new node for the interval to be merged
Traverse the tree to find the first overlapping node (b)
if l lies before b.l then
    Traverse the left sub-tree
    if l is greater than the interval then
        Move to the next right node
    else if l is smaller than the interval then
        Include all elements of the right sub-tree in the new node and move to the next left node
    else
        Insert the elements after l and the elements of the right sub-tree in the new node
        break
    end if
end if
if u lies after b.u then
    Repeat the same thing for the right side
end if
if l and/or u lies within the interval then
    Create a new node for the remaining elements of b on both the sides
    Link it to the new node
end if
Remove the existing node and insert the new node
return root
```

The complexity of the functions is given by : O(n+k)
where, n is the number of intervals and k the number of elements
As, in the worst case, the whole tree is overlapping and all the elements need to be included in the new interval and also all the nodes have to be traversed.