Assignment 6

Avijit Ghosh

February 2016

1 INTERVAL TREES

Interval Trees have been made using a Binary Tree structure where each structure has two nodes (left and right child) and a data point, which is a sorted link list. The tree is always balanced and sorted (BST).

2 CREATE INTERVAL TREE

1. The l and u values are passed to the function, along with n-the no. of intervals.

2. I find the subrange length ass size/n

3. I begin from the first index (l) and keep incrementing by the subrange size to get my ranges. The remainder goes to the last range.

4. Now, these ranges are passed to another function called allot.

5. The ranges are already sorted, so no sorting is applied.

6. We find the media by taking the middle element of the lower range index. In case of even number, the higher one is taken.

7. The median range is chosen and created as a root. All the lower ranges go to the left child and all the higher ones go to the right child and the process is applied recursively.

8. By this method, each element in the n ranges is accessed once. There is no explicit sorting involved and thus the complexity is O(N)

3 MERGE

1. The node is picked recursively until the given l and u lie entirely inside itself or its left and right children.

2. An inorder traversal of the tree is done from that point onwards to collect all the values in a linked list.

3. The range values are collected in another array. The given [l, u] values generated randomly are inserted. Now the array is sorted. (O(N) as I use a list and insert in-place)

4. A loop is now run over the range values to merge the ranges. If the incoming range has a lower start value than the already existing end value, the maximum of the two end values are taken (merge). Otherwise a new range is created and appended to the list.

5. Finally, the new ranges are used by the allot function (as written above) to allot nodes based on the median strategy to balance, and then insert is repeatedly called with the values in our linkedlist to populate that tree.

6. Finally, this new tree is attached to the same place where its root used to be.

7. The elements are accessed in O(n) time and the height of the tree is log(n), so the complexity is O(n log n).