Grant Mercer

Assignment 5

CS 302

Assignment 5 Part B write-up:

Summary of the AVL tree data structure:

The AVL tree is a very unique and interesting data structure. The tree structure itself is very similar to that of the binary search tree. The AVL node holds data as well as a left and right pointer, and now a height variable as well. The height variable will be used to self-balance the tree during insertions and deletions in order to maintain balance. The base functions are exactly those of the binary search tree offering simple insertion deletion, but the important part is that AVL tree lookup’s are very fast.

AVL Tree compared to Binary Search Tree:

The AVL tree holds a number of benefits of the Binary Search Tree. One thing to note is that BST without balancing has a very bad worst case, that being said a Binary Search Tree can have a very bad worst case when dealing with large data. On the other hand the AVL tree assures that the lookup time of any node will be O(log N) apart from the root. One disadvantage of the AVL tree however is it consumes much more memory as it must keep track of a balance factor in every node, and can be weighed down with self-balancing execution costs(though only constant). The AVL seems a much more superior data structure when lookup’s are used often.

Big O for AVL tree options

LookUp – O(log N)

Insert O(log N)

Delete O(log N)