COMP 282

Lecture 18

AVL Trees
AVL Trees

- AVL Trees are Binary Search Trees with the additional notion of “height” for subtrees.

- Height information can be represented as a member field of the Node class just like color for Red-Black trees.
- Except that at least a byte is required compared to a bit.
- Height of any node can be computed by adding 1 to the height of the taller of the left or right subtrees.
Restrictions

• The only rule then is:
  – No node is allowed to have children who differ in height by more than one.

An example of an unbalanced tree:
Operations for Rebalancing

- Balance is restored through the use of Left and Right rotations. (similar to Red-Black tree operations)

- A simple Left Rotation restores the balance.
Cases:

- What operations are required depends on the relationship of the left and right subtrees in the too-tall tree.

- If the problem is caused by the outside subtree then a single rotation is all that is needed to restore balance.
Pathological case

- Like Red-Black Tree rotations its not always that simple:

- If the inner subtree is taller a single rotation leaves it just as unbalanced.
Inner subtree is the culprit

- Inner subtrees can be the culprit in two ways:

- Either case can be corrected by the same two rotations.
First culprit: Two rotations

- The first rotation is designed to move the tallest subtree to the outside...

- The second balances the tree.
Second culprit: two rotations