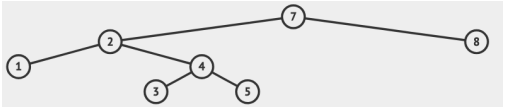
Computer Science 385
Design and Analysis of Algorithms
Siena College
Spring 2025

Balanced Trees Practice

Let's insert the values 1,2,3,4,5,6,7 into a BST and maintain a strict balance.

State the AVL condition:

Which of these trees satisfies the AVL condition and why?



Let's insert values and look carefully at the cases that require rotations.

Start by inserting 3, 2, then 1.

The root node now violates the AVL condition. We violated the condition with an insert into the left/right subtree of the left/right child of the offending node. A single rotation will correct for this.

Now insert 4 then 5.

The node containing ___ now violates the AVL condition. We violated the condition with an insert into the left/right subtree of the left/right child of the offending node. A single rotation will correct for this.

Insert 6.

Violation at ___, insert into left/right subtree of the left/right child. Also a single rotation.

Then insert 7.

These have all been single rotations, (left subtree of left child or right subtree of right child) which follow this general pattern (or its mirror image).

Picking up where we left off, let's insert 16 then 15.

Violation at ___, insert into left/right subtree of the left/right child. This requires a double rotation.

Now insert 14.

Violation at ___, insert into left/right subtree of the left/right child. This requires a double rotation.

The double rotations follow a bit more complicated pattern.

An analysis (that we won't do) can show that the height n of an AVL tree is guaranteed to satisfy the inequality:

Which means the worst case height of an AVL tree is $\Theta(\quad)$.

This will also be the worst case efficiency of important operations on AVL trees.

Let's insert the values 10, 20, 30, 40, 50, 60, and 70 into a 2-3 tree.

Label some of the nodes as 2-nodes, 3-nodes, or 4-nodes.

Now we insert 35, 25, 55, 75, 5, and 7.