**AVL Trees**

What is the performance of a binary search tree?

<table>
<thead>
<tr>
<th>Operation</th>
<th>Best Case</th>
<th>Worst Cast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Retrieve</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Traverse</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Can we do better?

**AVL Tree** –

Define the height of each node

**$p$-tree** –

AVL is a 1-tree, therefore

The height of an AVL Tree is roughly $1.44 \log(N+2) - .328$, which is

**Example AVL Trees**

Is it an AVL Tree?

**1 node tree**

**2 node trees**

**3 node trees**
More Examples – AVL Tree yes or no?

What is the effect on the height of inserting 1 node?

Insertion into an AVL Tree
1. Insert node using the standard technique
2. Check the heights of the subtrees along the path of insertion and fix any unbalances by performing rotation

4 insert positions
1. An insertion into the left subtree of the left child of $\alpha$
2. An insertion into the right subtree of the left child of $\alpha$
3. An insertion into the left subtree of the right child of $\alpha$
4. An insertion into the right subtree of the left child of $\alpha$

Rebalance process similar for 1 and 4. Process similar for 2 and 3.

Single Rotation
Cases 1 and 4 – the outside of the tree is heavy

Insert 33 into the tree (verify it is currently an AVL tree)
Insert 70 followed by 80

Single rotation maintains Binary Search Tree characteristic
How long does it take?

Create an AVL Tree
Beginning with an initially empty AVL Tree insert the keys 3, 2, 1, 4, 5, 6, 7.
Double Rotation
For cases 2 and 3 – where the middle of the tree is heavy

First, try a single rotation on the following tree

Double rotation process
1. Make the outside heavy with a single rotation
2. Rebalance the tree with another single rotation

Insert 53 followed by 58
Beginning with the following tree insert 16, 15, 14, 13, 12, 11, 10, 8, 9