AVL Trees

An **AVL Tree** is a binary tree that maintains balance, so that its operational complexity is closer to O(log(n)).

Motivation

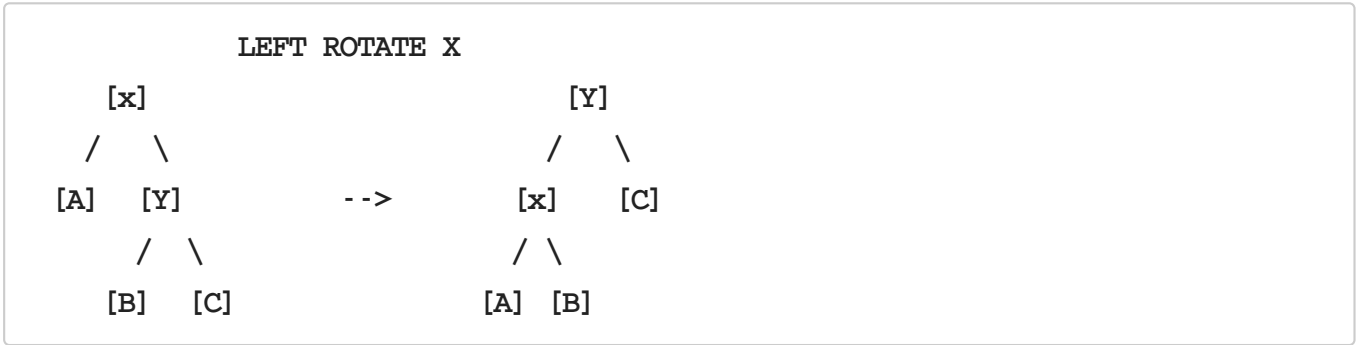
Binary search trees work in O(n), because in the worst case they behave like a linked list.

How Are the Properties Maintained?

Insert

We perform the usual insert for a binary search tree. If we find that three or more nodes sit in a line, do a rotate on the head of that sublist.

Rotations



The rotation maintains the in-order notation. It may sometimes take two rotations to achieve a balances tree.

Worst case for AVL

Based on the fundamental properties, the worst case for an AVL tree is that every left and right node have a height difference of one. So the motif below repeats for every level.

```
    /\
   / /\
  /\/  \
```

So the recursive folmula for height of AVL is
$N_h = 1 + N_{h-1} + N_{n-2}$

WHICH IS ALMOST FIBONACCI WTF??

$$
\begin{aligned}
N_n &> 1 + 2N_{n-2} \\
&> 2N_{n-2} \\
&= \theta(2^{h/2}) \\
h &< 2\log(n)
\end{aligned}
$$

And we know that fibonacci is exponential.
Height < 1.44log(n).