CMPT 225 - AVL Trees

AVL Trees are a kind of "self-balancing" BST.
Their height is always at most 2log(n), where n is number of keys.

An AVL tree is a BST that satisfies the following
height-balance invarient

For every node v
|height(left(v)) - height(right(v))| <= 1

We define height(left(v)) = -1 if left does not exist, and similarly for right(v).

We are concerned about the height of one side vs the other, not the number of nodes necessarily.

Implementation:
insert, then balance

Uses rotations to balance.

4 cases:

2 outside cases *(one rotation)*

- left left
- right right

2 inside cases *(two rotations)*

- left right
- right left

With insertion you can cause a bunch of nodes to become unbalanced but with a single fix you fix the whole tree.
With deletion you cause only one node to become unbalanced but fixing this node can cause other nodes to become unbalanced.