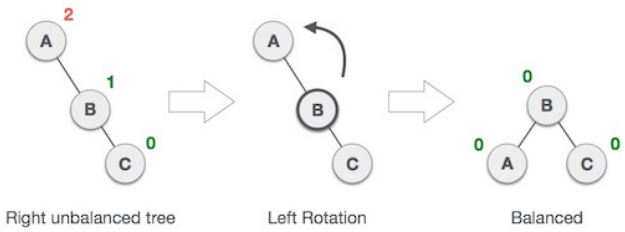
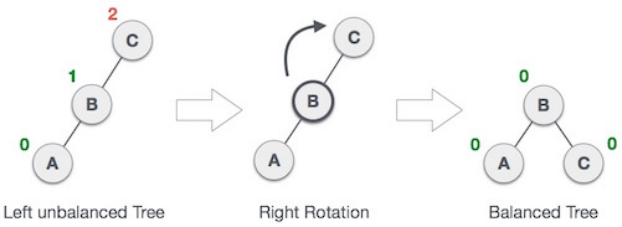
AVL TREE CHEATSHEET

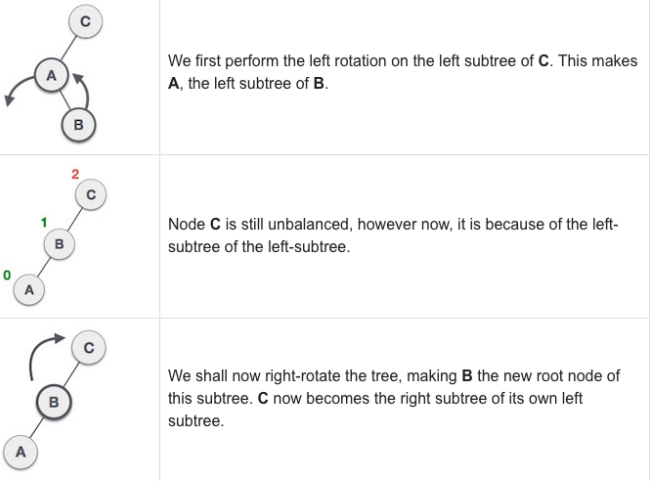
When an imbalance is caused by a node being inserted into the right subtree (C) of the right subtree (B), we use a left rotation.



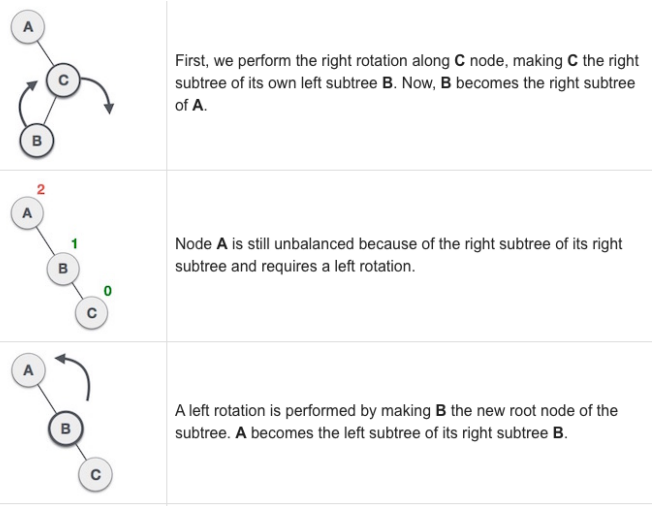
When an imbalance is caused by a node being inserted into the left subtree (C) of the left subtree (B), we use a right rotation.



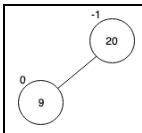
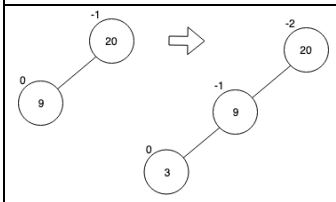
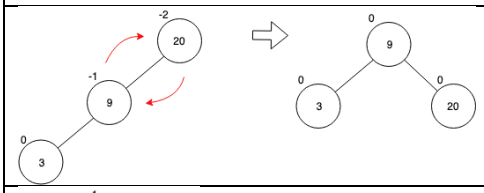
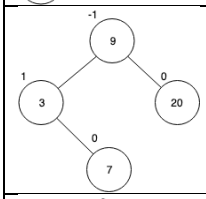
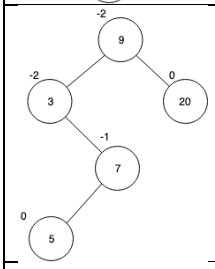
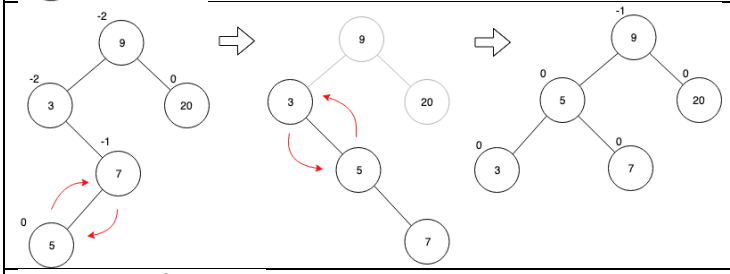
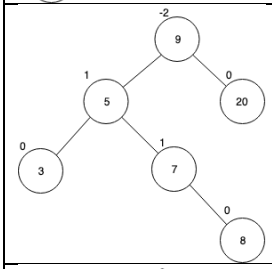
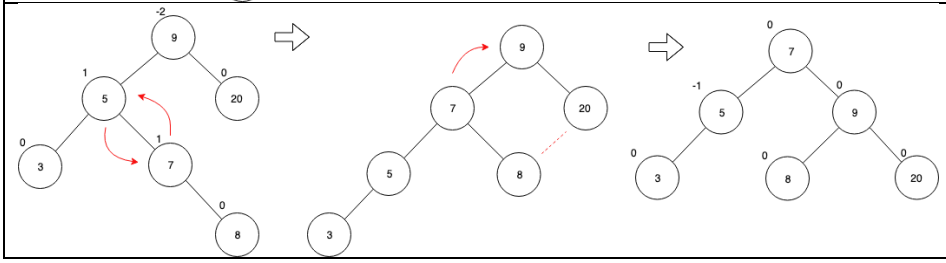
When an imbalance is caused by a node being inserted into the right subtree (B) of the left subtree (A), we use a left-right rotation.



When an imbalance is caused by a node being inserted into the left subtree (B) of the right subtree (A), we use a right-left rotation.



AVL TREE CHEATSHEET

	**Insert 20, 9** No rebalance required.
	**Insert 3** Node 20's subtrees are unbalanced. Imbalance caused by insertion into left subtree of left subtree. Right rotation required.
	**Right rotation on 20 and 9** Tree is now balanced.
	**Insert 7** No rebalance required.
	**Insert 5** Node 3's subtrees are unbalanced. Imbalance caused by insertion into left subtree of right subtree. Right-left rotation required.
	**Right-left rotation on 3, 7 and 5** - Right rotation on 7 and 6 - Left rotation on 3 and 5 Tree is now balanced
	**Insert 8** Node 9's subtrees are unbalanced. Imbalance caused by insertion into right subtree of left subtree. Left-right rotation required.
	**Left-right rotation on 9, 5 and 7** - Left rotation on 5 and 7 - Right rotation on 9 and 7 - Re-parent 8 (7's right child) to left-most of its new right child. Tree is now balanced.