AVL Tree: a binary search tree where no nodes have a balance factor greater than 1 or less than −1.

The only time the balance factor can change is when a node is added or removed. To re-balance a tree there are four cases:

Case 1: node’s balance factor is 2 and right child’s balance factor is 1 or 0.
- do a left rotation around the node.

Case 2: node’s balance factor is 2 and right child’s balance factor is −1.
- do a right rotation about the right child and then a left rotation about the node.
Case 3: node's balance factor is -2 and left child's balance factor is -1 or 0 - do a right rotation around node

Case 4: node's balance factor is -2 and left child's balance factor is 1 - do a left rotation around left child and a right rotation around node

Note: no nodes with duplicate keys are allowed, can solve this by defining nodes with equal keys are sorted by location in memory.