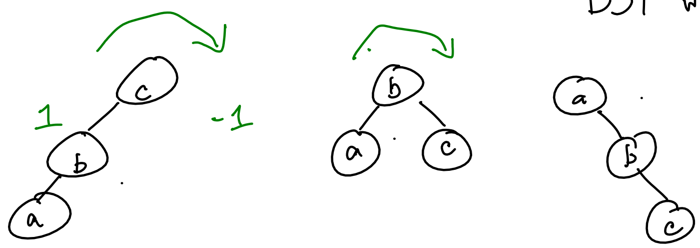
BST is a kind of Dictionary

Just the BST algorithms (w/o rebalancing) : slow (O(height))

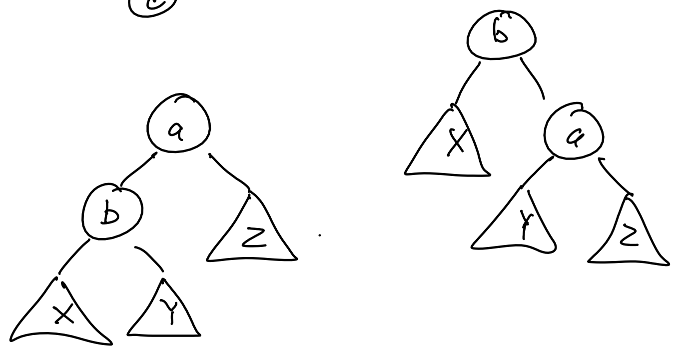
"Balanced" binary search tree : $\underbrace{\text{AVL tree}}$

BST where all nodes have children whose heights differ by no more than one




```
Function rotateRight(node):
    If node.left == null: !!
    a ← node
    b ← node.left
    Y ← b.right
    a.left ← Y
    b.right ← a
    Return b
EndFunction
```

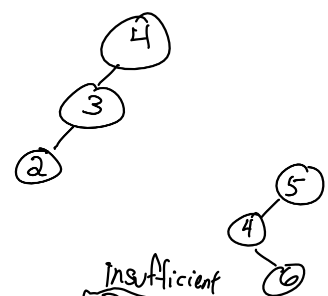
O(1)



AVL tree

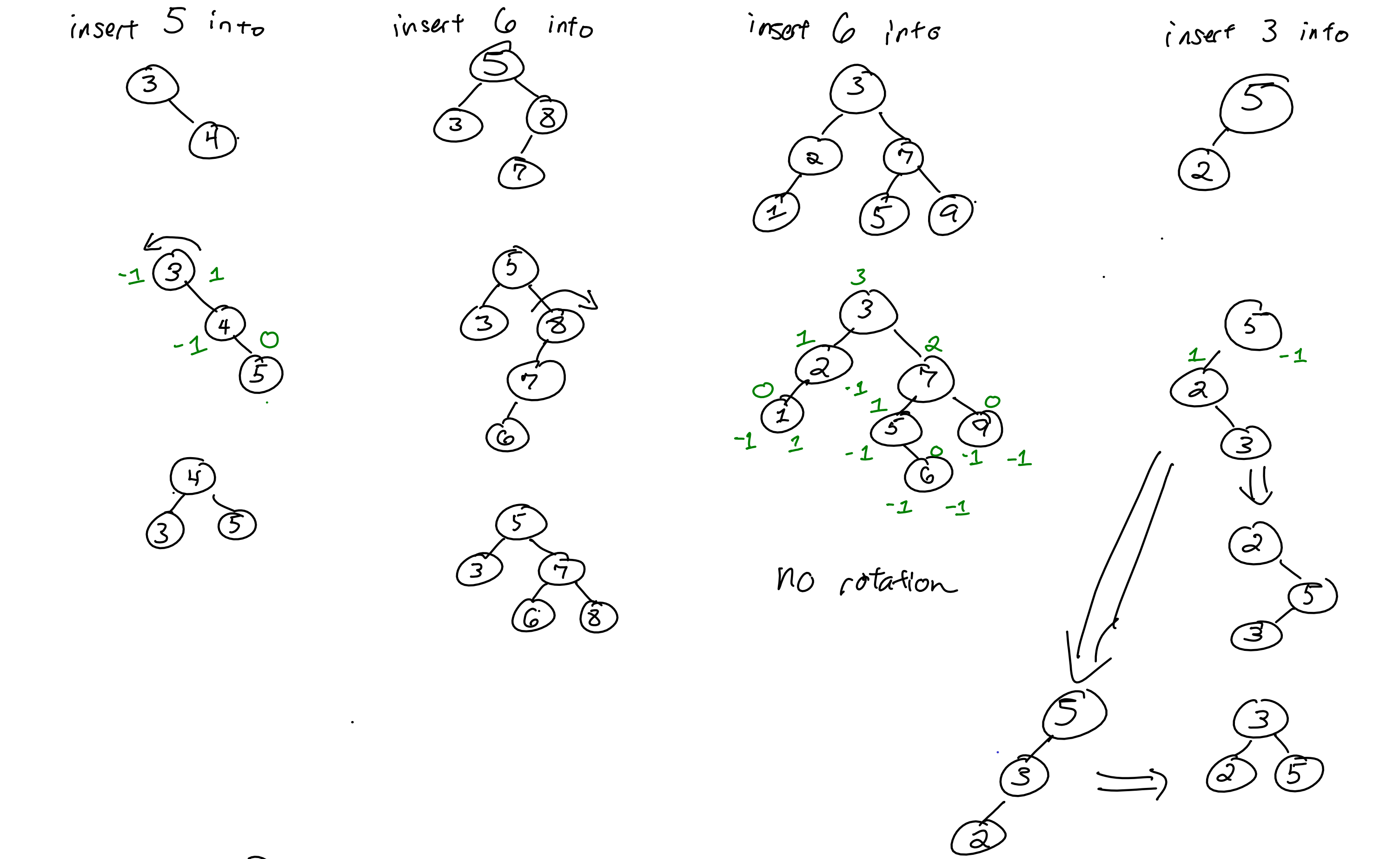
AVL tree

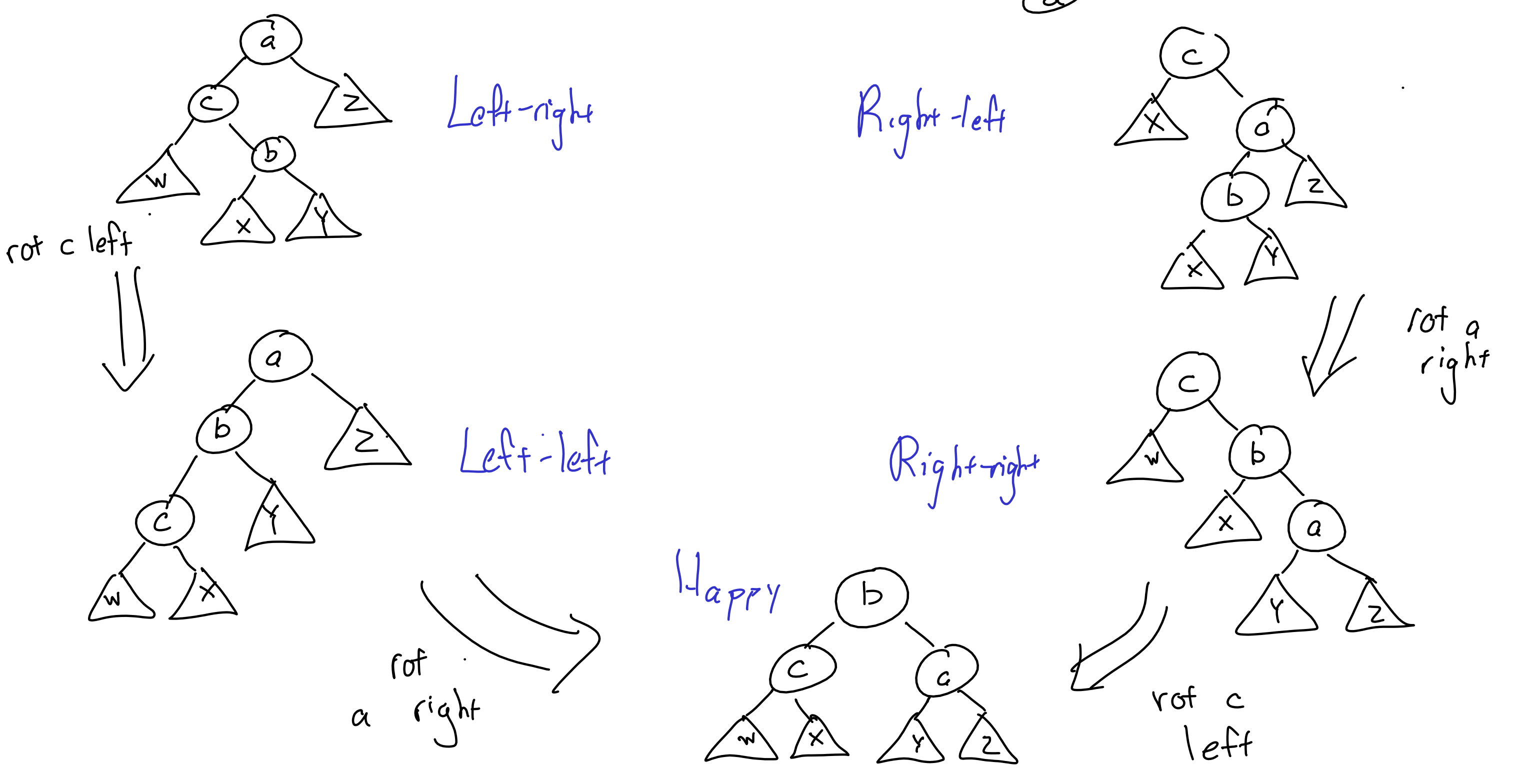
```
Function insertInSubtree(node, key, value):
    If node == null:
        Return new Node(key, value)
    Else If key < node.key:
        node.left ← insertInSubtree(node.left, key, value)
        node ← rebalance(node)
        recalcHeight(node)
        Return node
    Else ...
```



Insufficient

BST is a BT where left children are smaller and right children are larger


insert 5 into
insert 6 into
insert 6 into
insert 3 into
no rotation


Left-right
Right-left
rot c left
rot a right
Left-left
Right-right
Happy
rot a right
rot c left

```
Function rebalance(node):
    delta <- node.right.height - node.left.height
    If delta > 1 :
        If node.right.right.height < node.right.left.height :
            node.right <- rotateRight(node.right)
        EndIf
        node <- rotateLeft(node)
    Else If delta < -1 :
        If node.left.left.height < node.left.right.height :
            node.left <- rotateLeft(node.left)
        EndIf
        node <- rotateRight(node)
    EndIf
    Return node
```