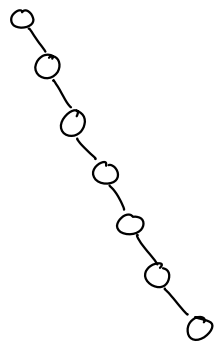
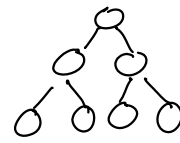


AVL trees (a kind of BST)

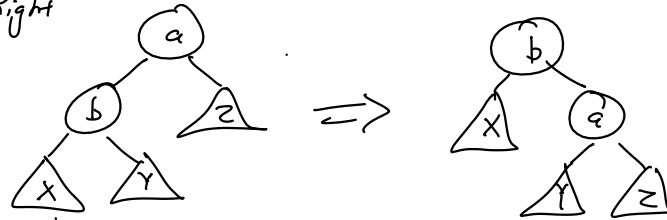
- BST algorithms all have $O(\text{height})$
- AVL trees: for all nodes, height of left and right subtrees differ by at most one

Full (and complete)



Tool: rotation

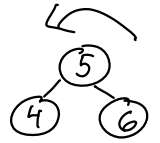
Rotate Right



```

a ← node
b ← node.left
Y ← b.right
b.right ← a
a.left ← Y
    
```

$O(1)$



insertInSubtree(key, value, node):

If node is empty tree:

Return new node w/ key, value

Else If key < node.key:

node.left ← insertInSubtree(key, value, node.left)

node ← rebalance(node)

recalc Height(node)

Return node

Else ...

Method rebalance(node):

If node.left.height > node.right.height + 1:
node ← rotateRight(node)

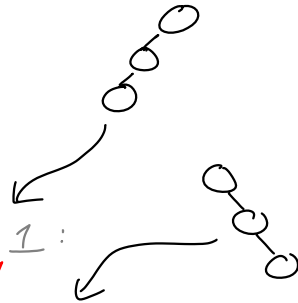
Else If node.right.height > node.left.height + 1:
node ← rotateLeft(node)

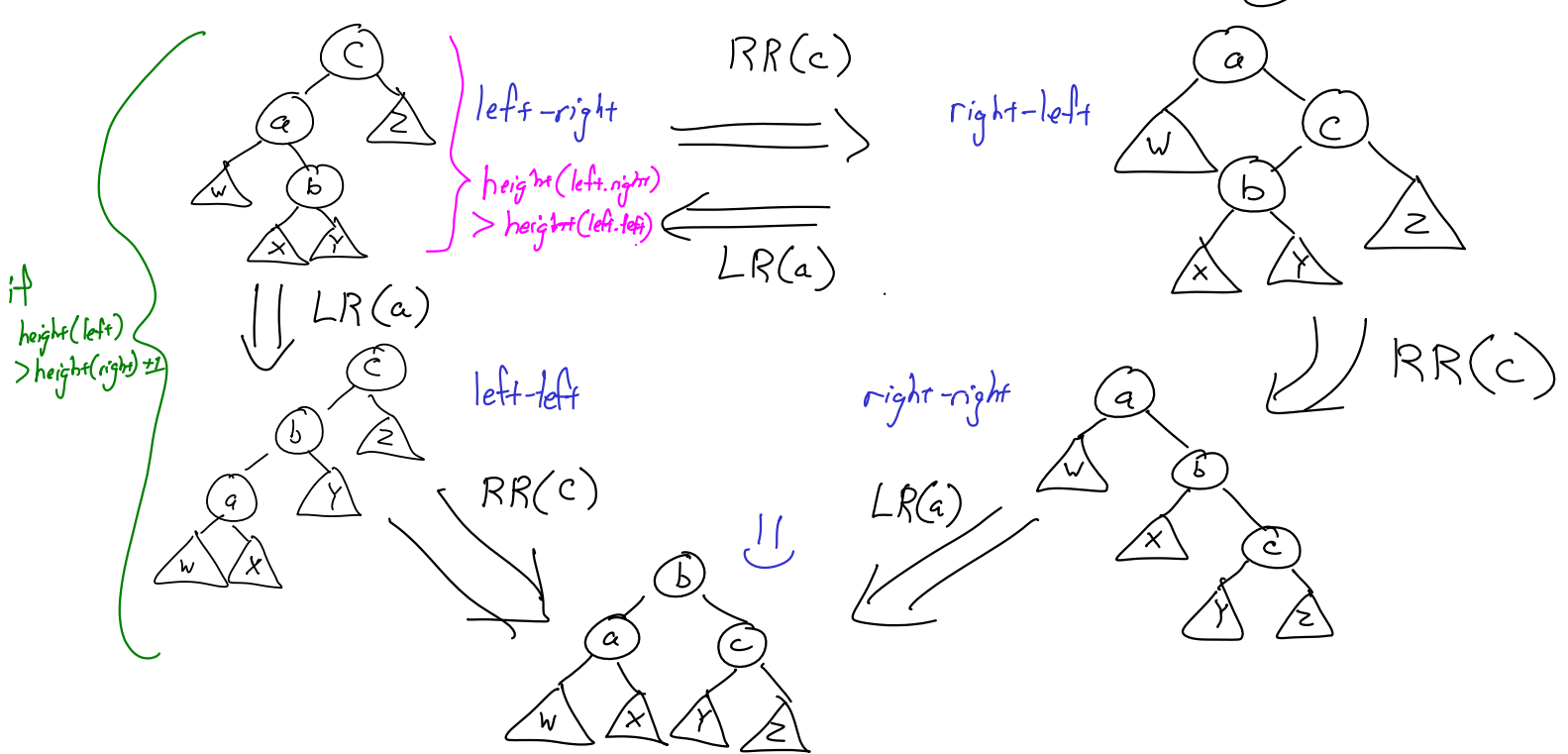
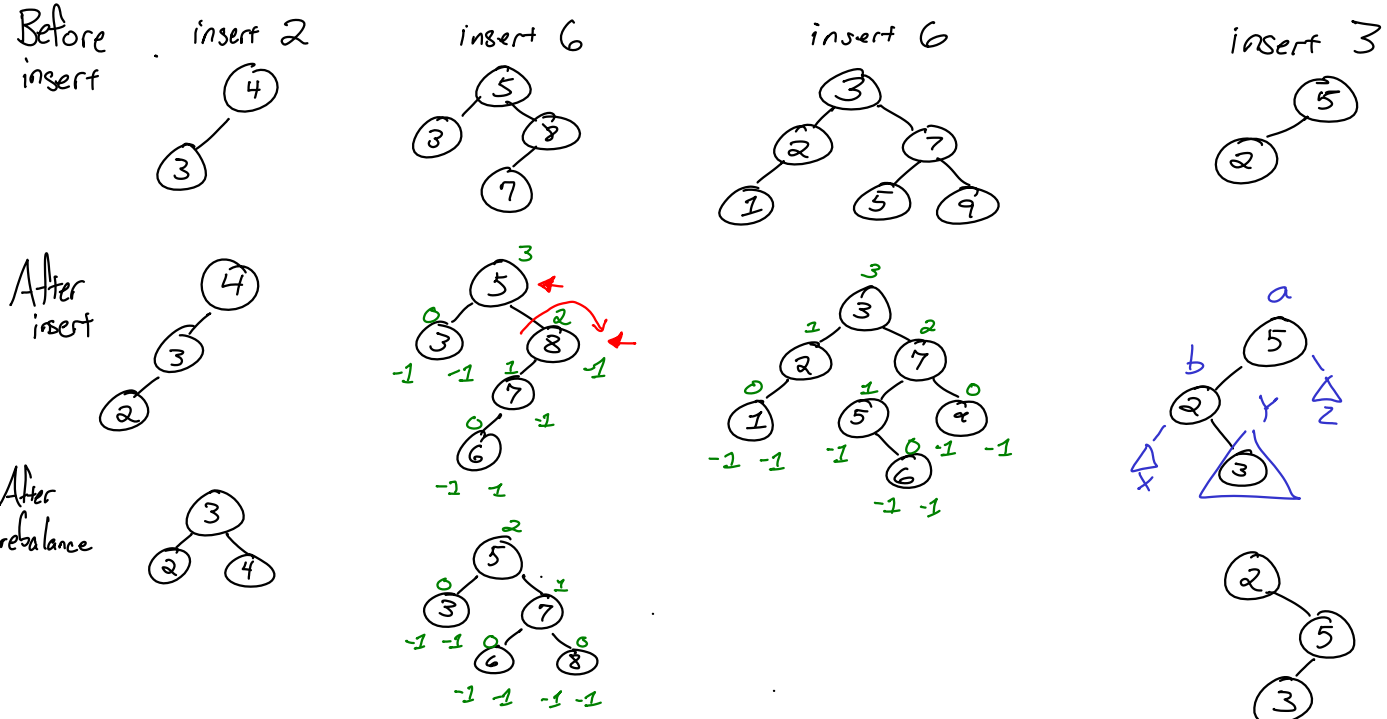
End If

Return node

End Method

Does not work





Method rebalance(node)

If $node.left.height > node.right.height + 1$: \leftarrow are we left-heavy?

If $node.left.left.height < node.left.right.height$: \leftarrow are we left-right case?

$node.left \leftarrow rotateLeft(node.left)$

End If \leftarrow now in left-left

$node \leftarrow rotateRight(node)$

Else- If $node.right.height > node.left.height + 1$:

End If

