# AVL Tree

BY
RAJANIKANTH B

# Introduction

**AVL Tree is also a Binary Search Tree but it is balanced tree.**

**Here balanced means the difference between height of right**

**s**

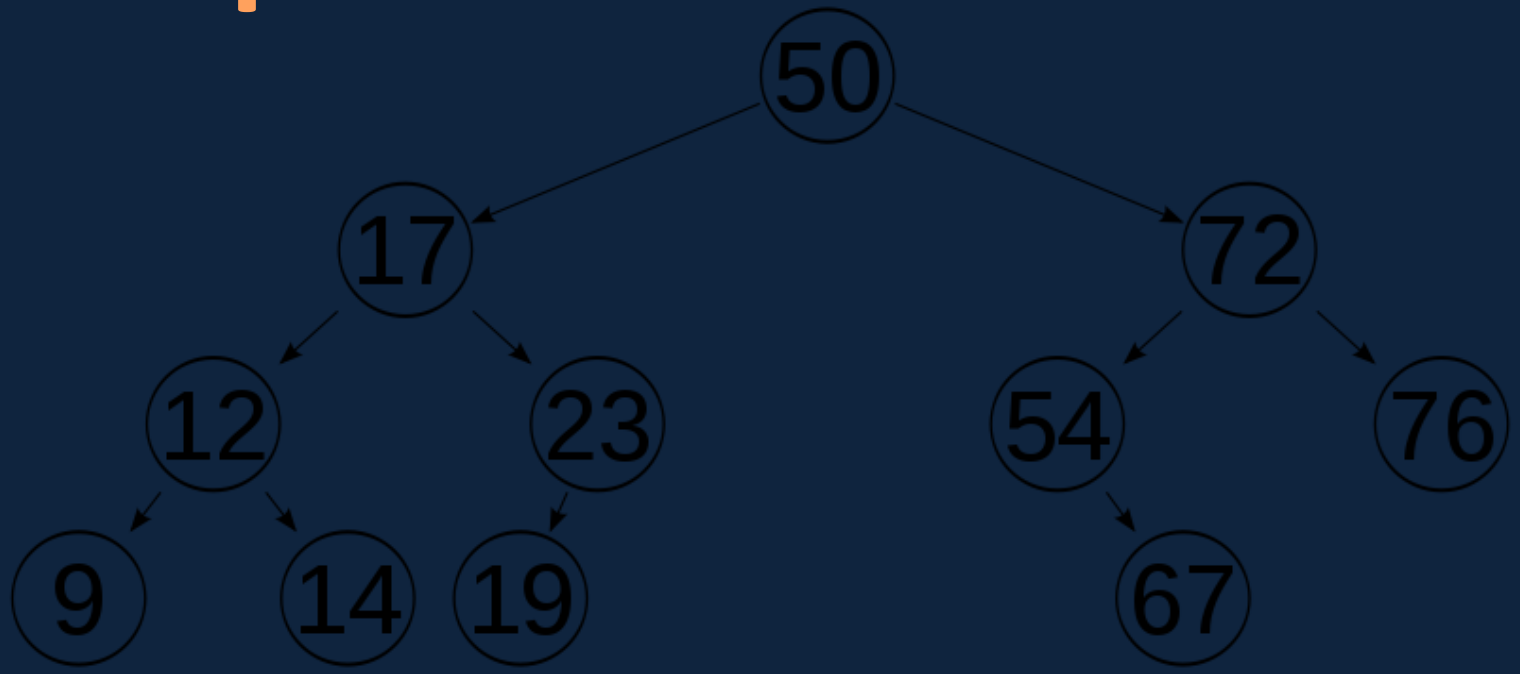
**In AVL Tree every node will have one extra information**

**known as BALANCE FACTOR.**

**BalanceFactor = HeightOfRightSubtree – HeightOfLeftSubtree**

**The AVL tree is named after its two inventors, G.M. Adelson-Velsky and E.M. Landis, who published it in their 1962.**

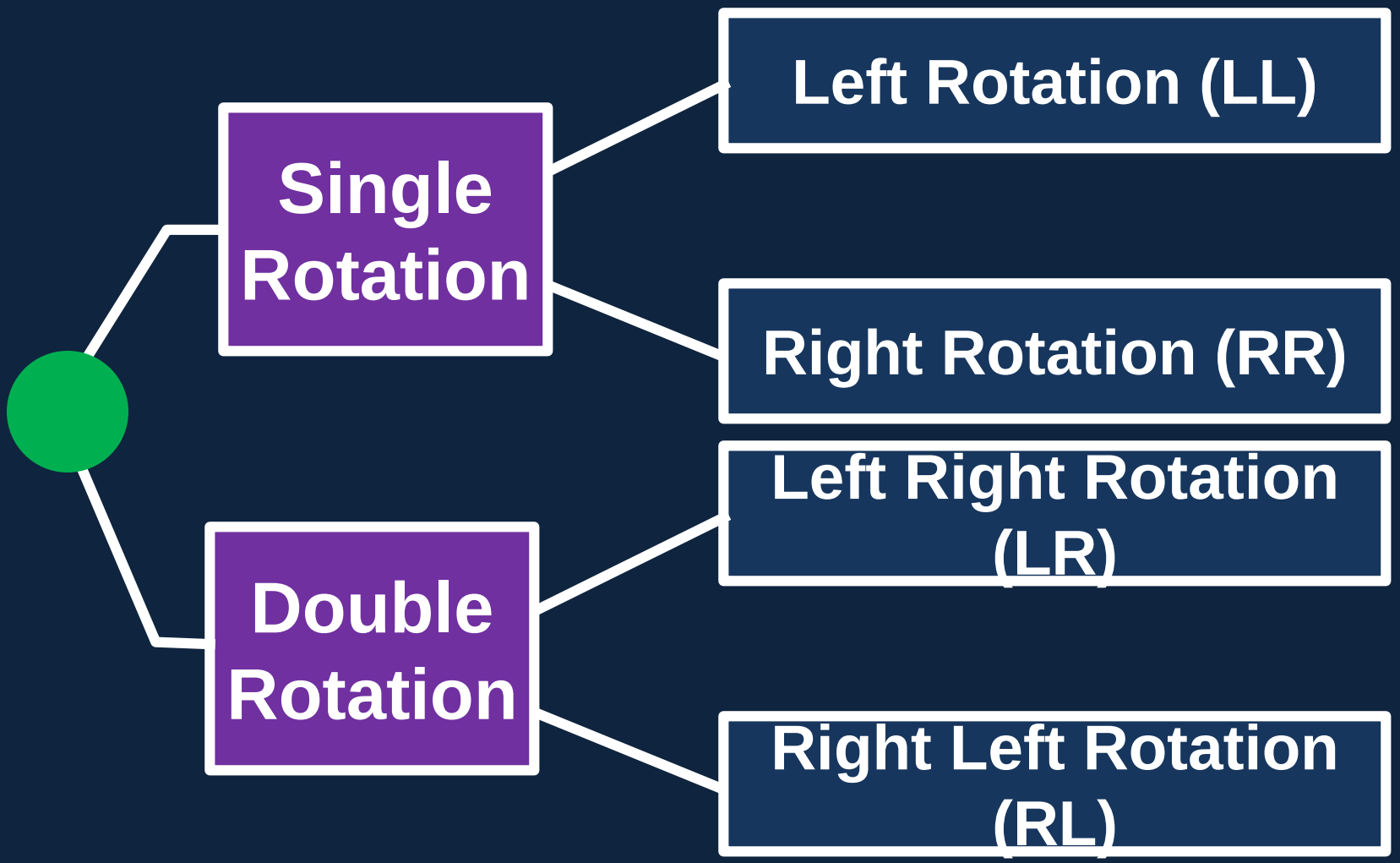
# Example



**After every insertion / deletion we need to check the BALANCE FACTOR**
**If it is other than -1 or 0 or +1 then we perform ROTATION to make the Tree Balance.**

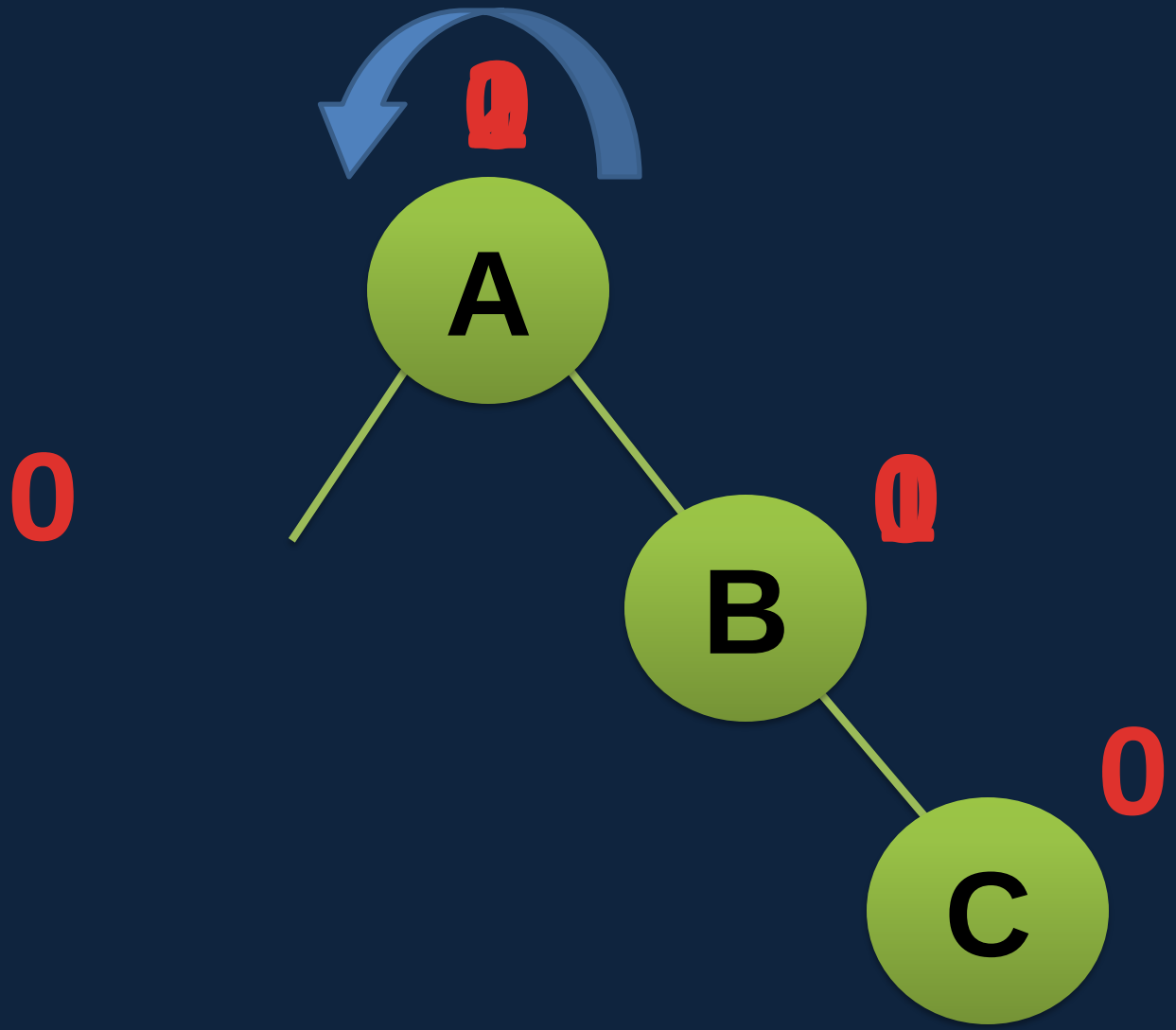

- **All AVL Trees are Binary Search Trees but All Binary Search Trees need not be AVL Tree**
- **All Binary Search Trees are Binary Trees but All Binary Trees need not be Binary Search Tree**

# Rotations


Single Rotation
Double Rotation
Left Rotation (LL)
Right Rotation (RR)
Left Right Rotation (LR)
Right Left Rotation (RL)

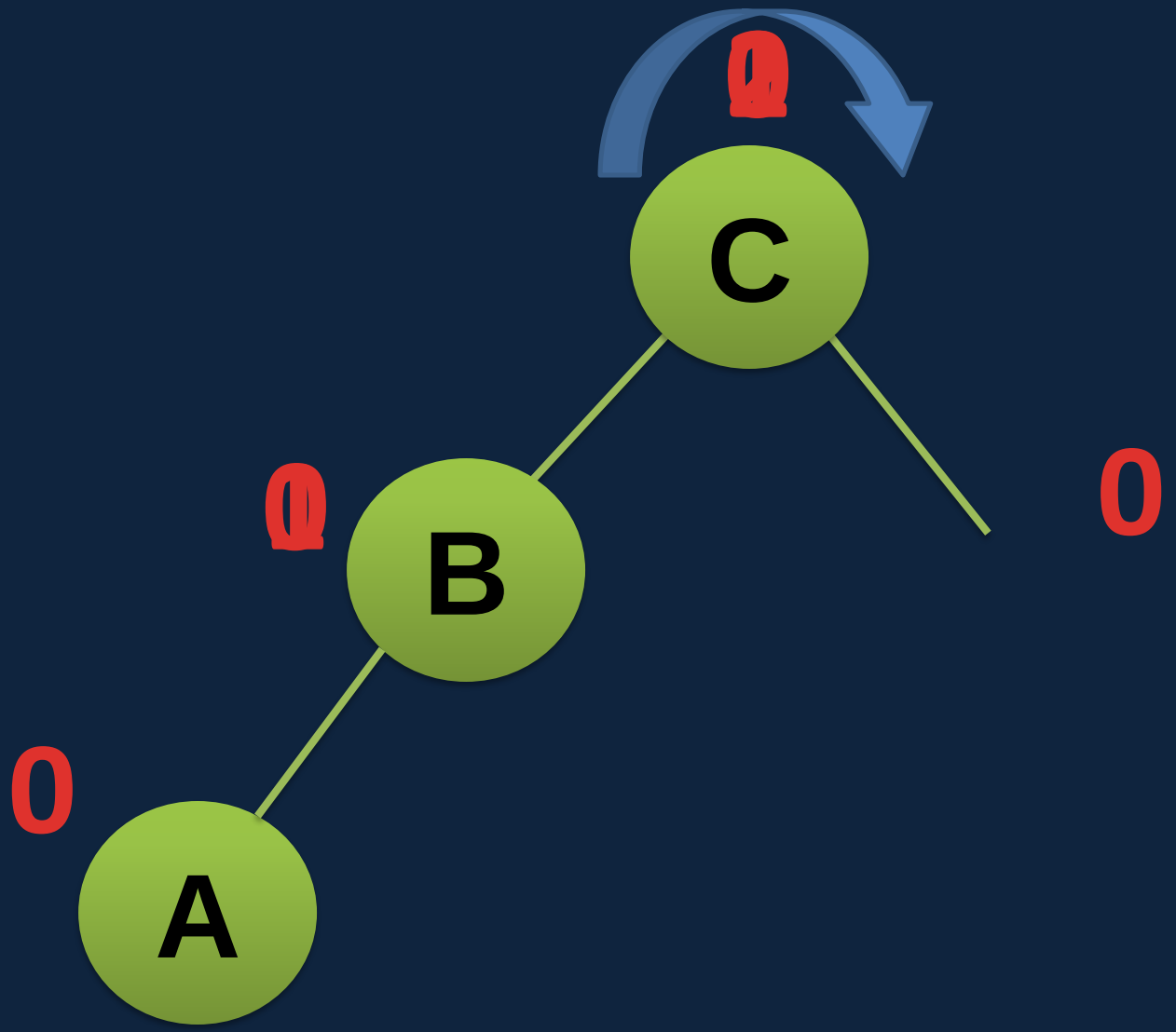
# LL Rotation

Insert: A , B , C



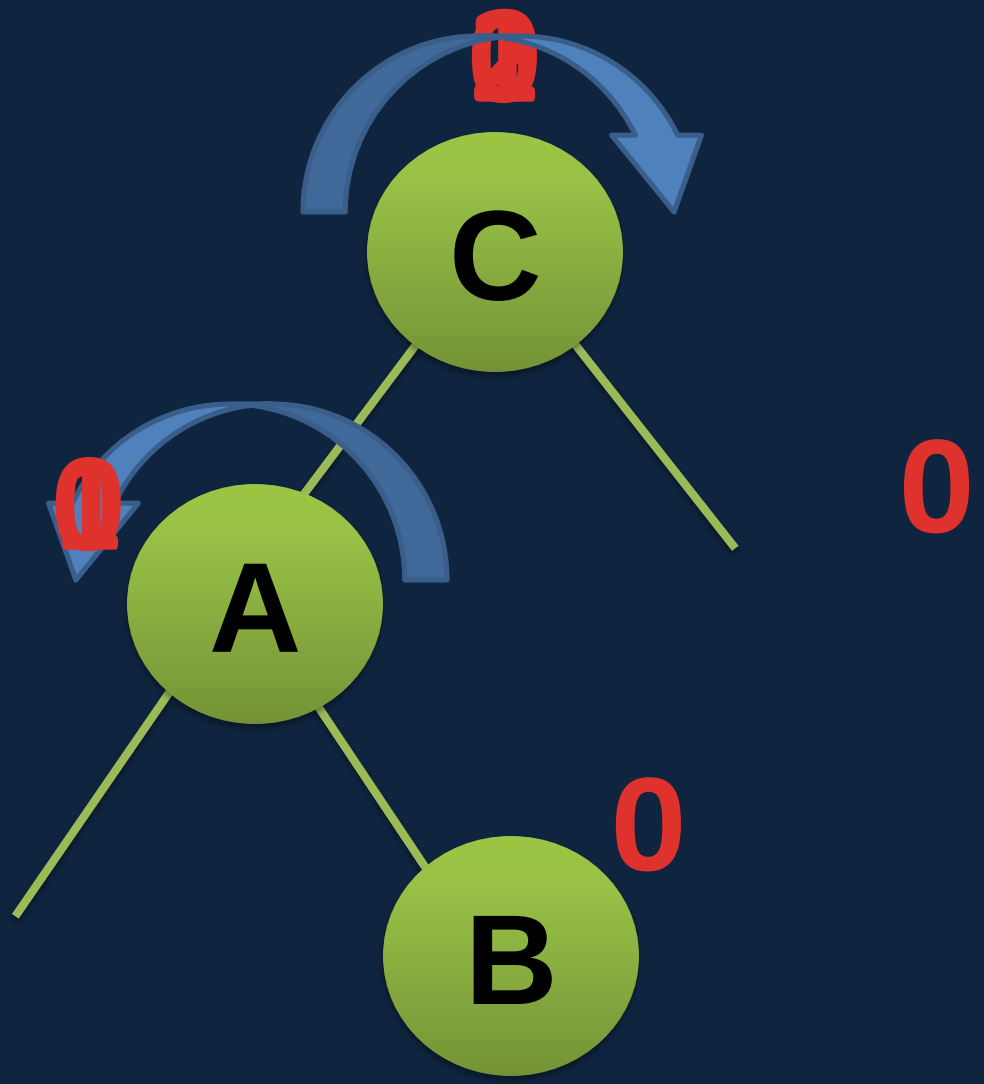
# RR Rotation

Insert: C , B , A



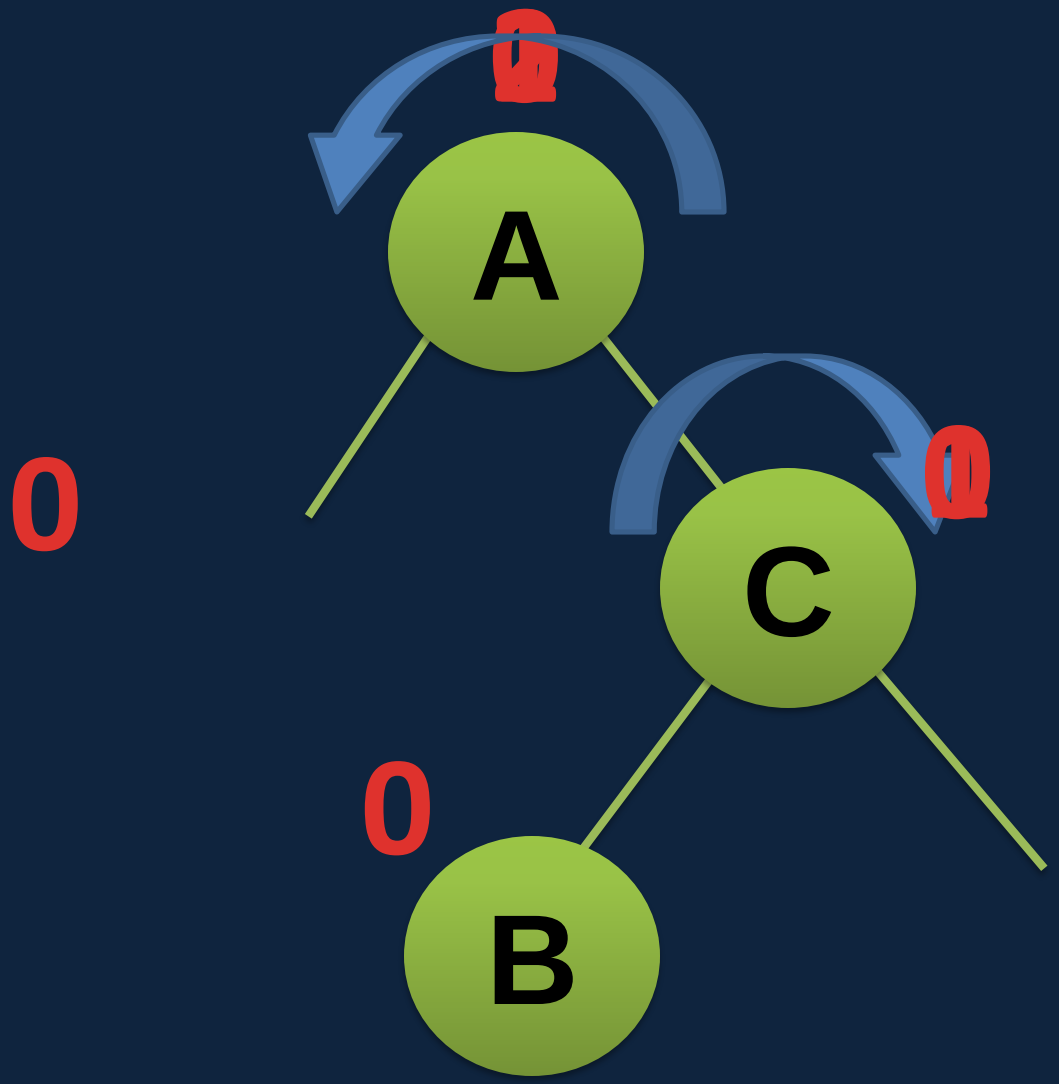
# LR Rotation

Insert: C , A , B



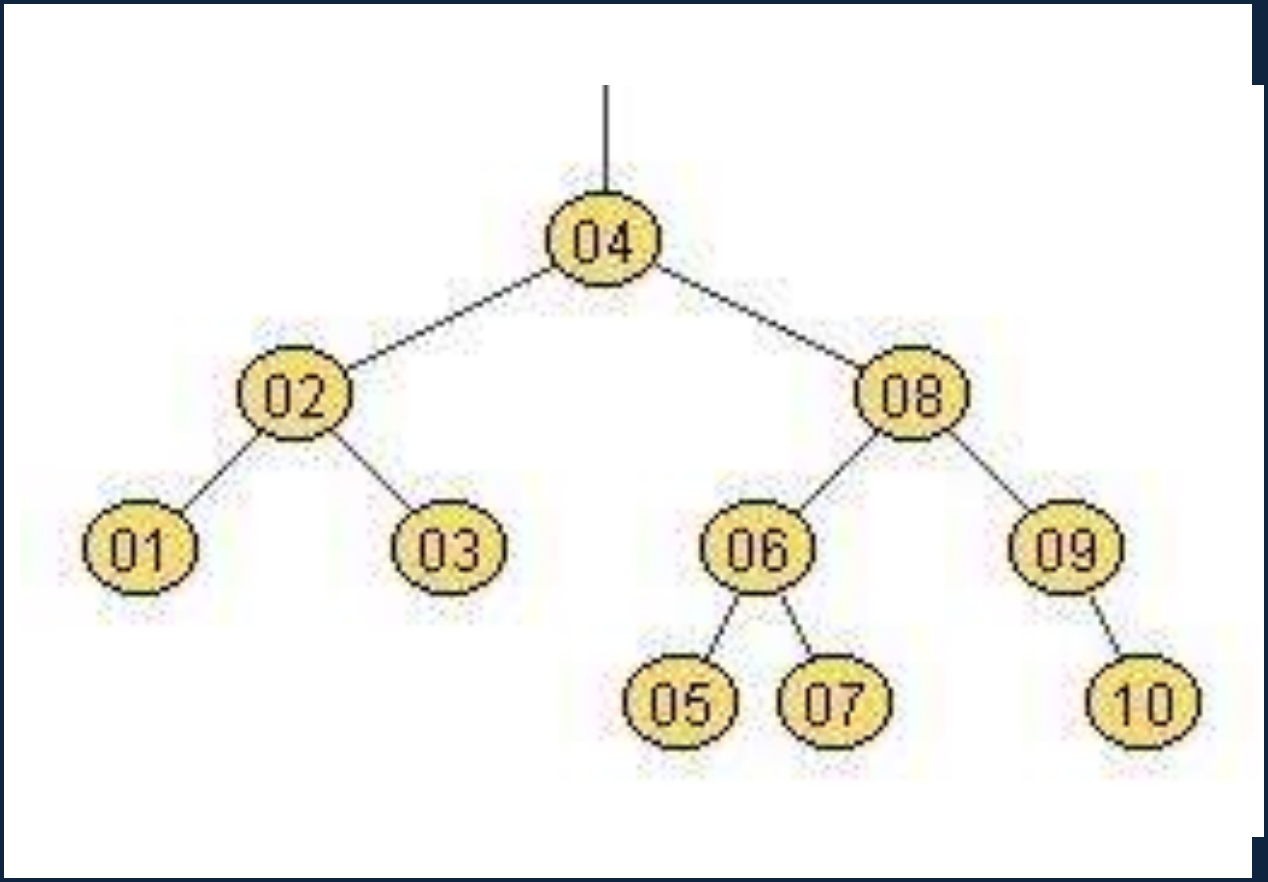
# RL Rotation

Insert: A , C , B



# Example

## Construct AVL Tree by inserting 1 to 10



# Deletion

## Deletion in AVL Tree have 3 cases

1. Deleting node without any child
2. Deleting node with one child
3. Deleting node with two children

# 1. Deleting node without any child

**Step – 1:**

Find given node in AVL Tree by performing search operation

**Step – 2:**

Remove given node in AVL Tree by using ‘delete’ operator

**Step – 3:**

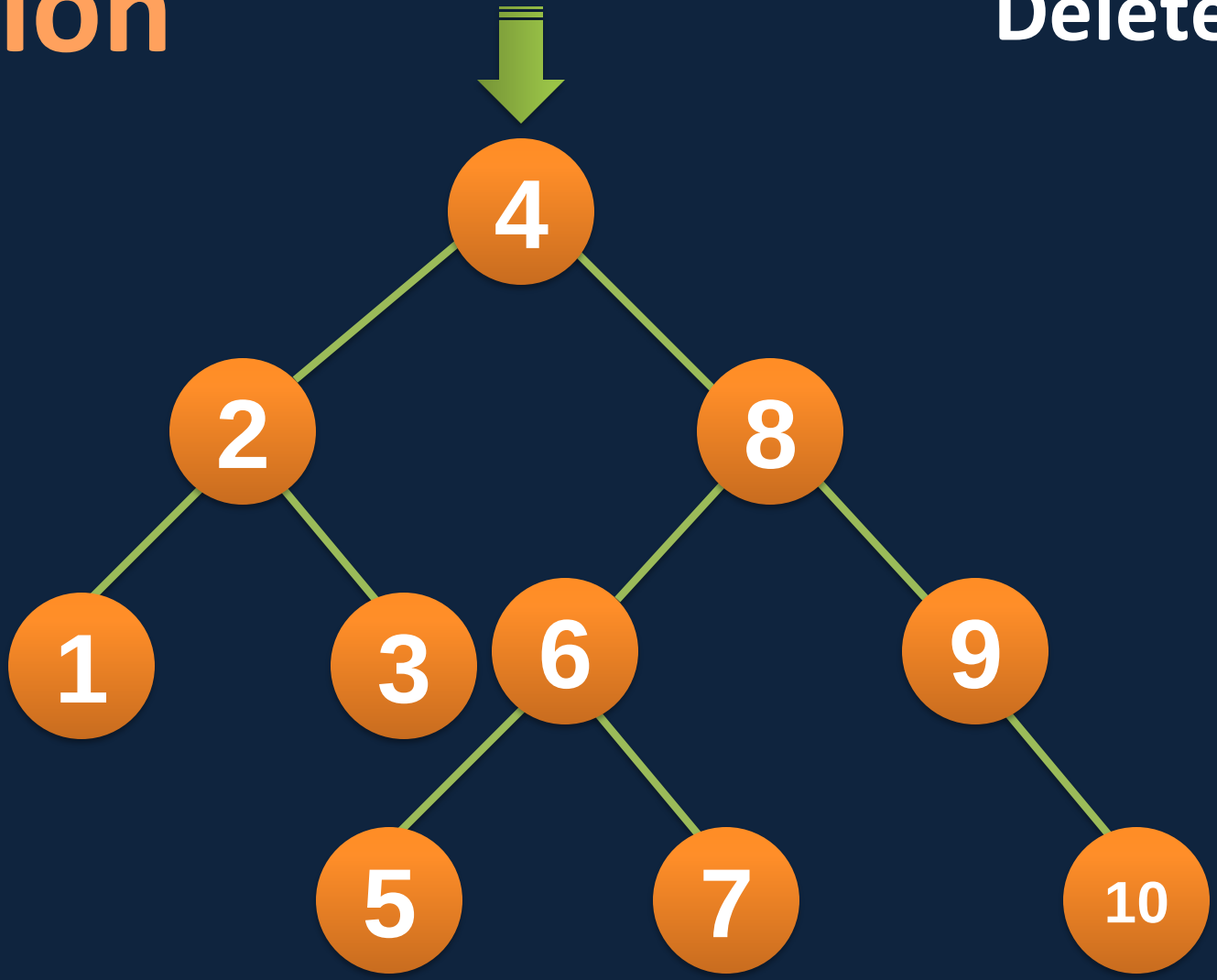
Check balance factor of each node in AVL tree

**Step – 4:**

If tree is not balanced perform suitable Rotation operation to make tree balanced

# Deletion

Delete: 5



# 2. Deleting node with one child

**Step – 1:**

Find given node in AVL Tree by performing search operation

**Step – 2:**

Make a link between its Parent and its Child

**Step – 3:**

Remove given node in AVL Tree by using ‘delete’ operator

**Step – 4:**

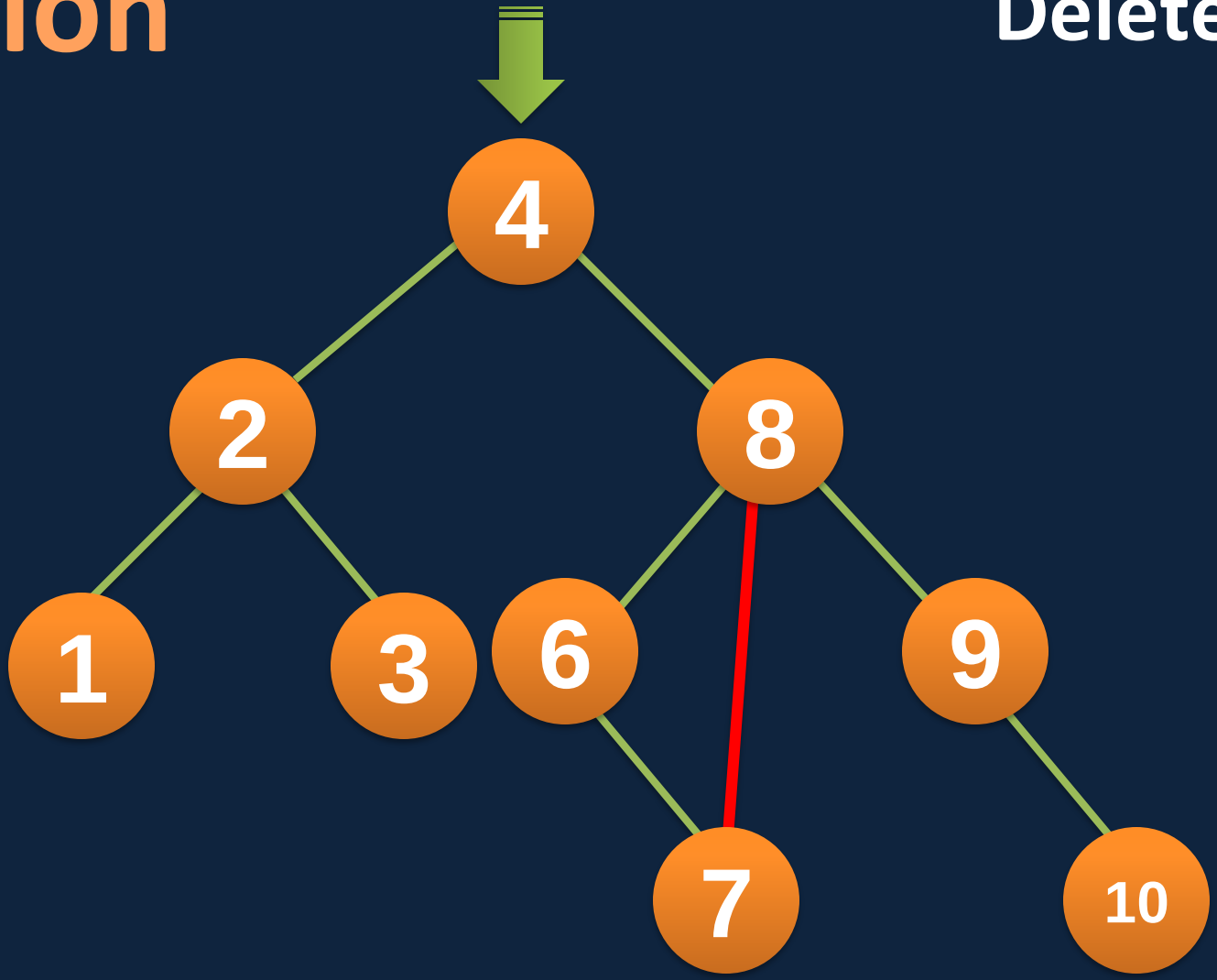
Check balance factor of each node in AVL tree

**Step – 5:**

If tree is not balanced perform suitable Rotation operation to make tree balanced

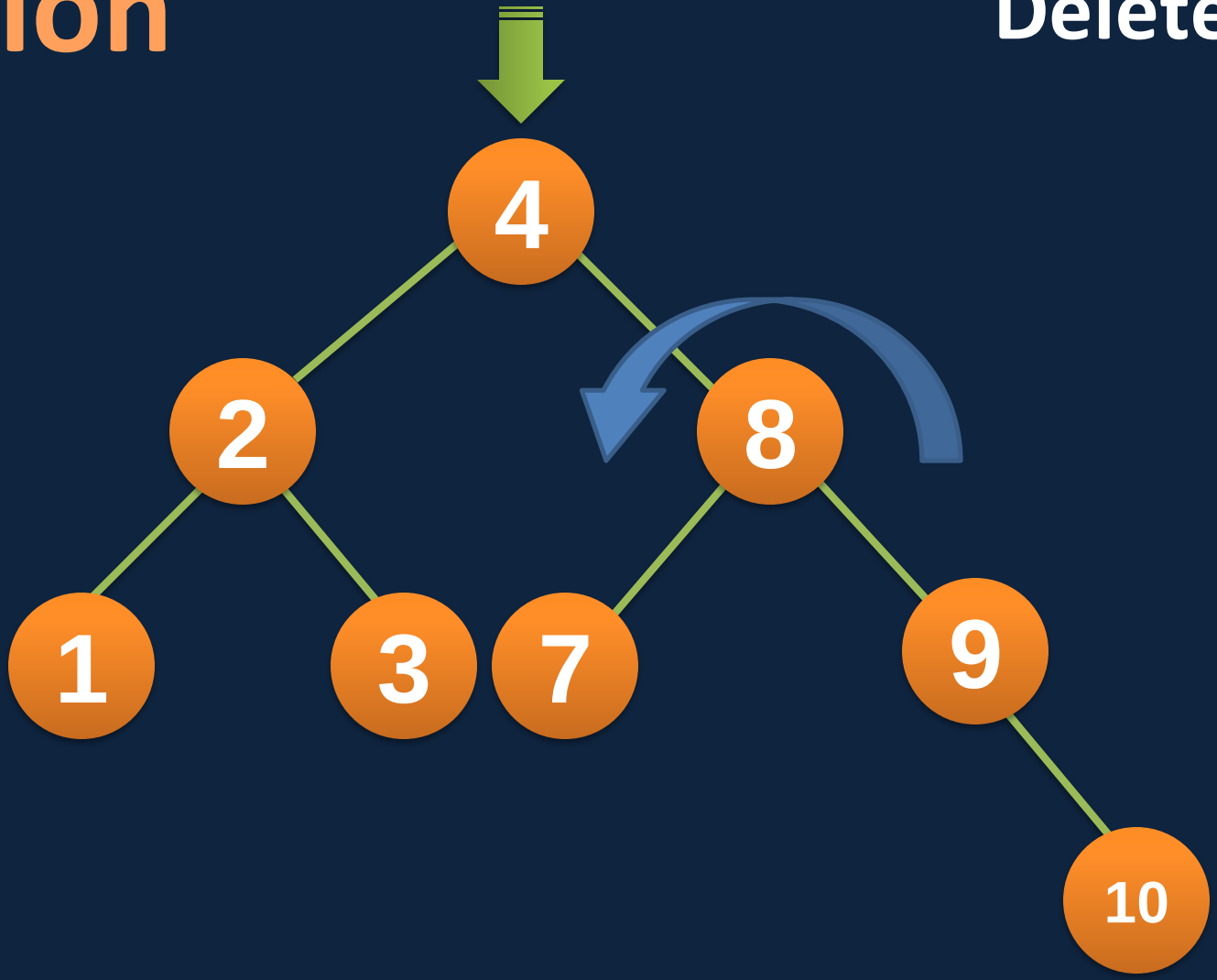
# Deletion

Delete: 6



# Deletion

Delete: 7



# 3. Deleting node with two children

**Step – 1:**

Find given node in AVL Tree by performing search operation

**Step – 2:**

Find the Smallest node in its Right Subtree

**Step – 3:**

Swap both given node and the Smallest node in its right subtree

**Step – 4:**

Remove given node in AVL Tree by using ‘delete’ operator

**Step – 5:**

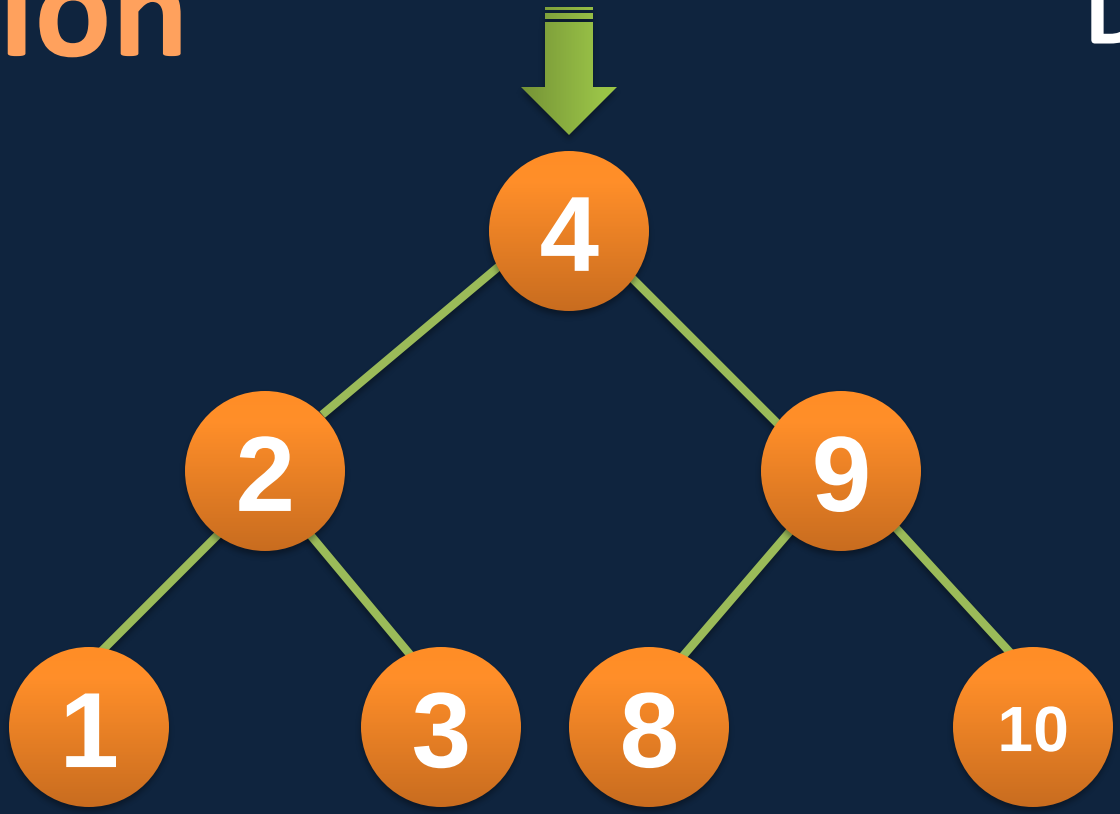
Check balance factor of each node in AVL tree

**Step – 6:**

If tree is not balanced perform suitable Rotation operation to make tree balanced

# Deletion

Delete: 4



# Assignment

Construct AVL Tree by inserting 10 to 1

Write applications of AVL Trees

Explain Deletion in AVL Tree in detail