

# Contents

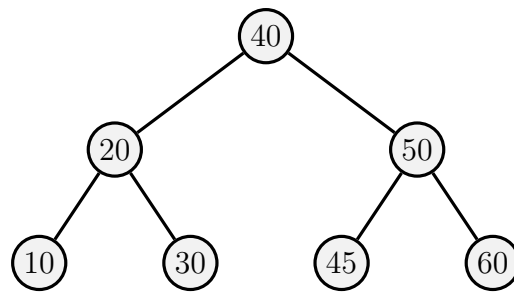
<b>1</b>	<b>Introduction to AVL Trees</b>	<b>3</b>
<b>2</b>	<b>Binary Search Tree</b>	<b>3</b>
<b>3</b>	<b>Example of Creating Binary Search Tree</b>	<b>4</b>
<b>4</b>	<b>Can We Improve Binary Search Tree?</b>	<b>4</b>
4.1	Order-1 . . . . .	5
4.2	Order-2 . . . . .	5
4.3	Order-3 . . . . .	5
4.4	Order-4 . . . . .	5
4.5	Order-5 . . . . .	6
4.6	summary . . . . .	6
4.7	LL-Rotation . . . . .	7
4.8	RR-Rotation . . . . .	7
4.9	RL-Rotation . . . . .	7
4.10	LR-Rotation . . . . .	7
<b>5</b>	<b>What is AVL Tree?</b>	<b>8</b>
5.1	Examples of finding balance factor of a node . . . . .	8
<b>6</b>	<b>Types Of Rotations</b>	<b>9</b>
6.1	LL-Rotation . . . . .	9
6.2	RR-Rotation . . . . .	10
6.3	LR-Rotation . . . . .	10
6.4	RL-Rotation . . . . .	11
<b>7</b>	<b>Rotations With Sub-Trees</b>	<b>11</b>
7.1	LL-Rotation With Sub-Trees . . . . .	11
7.2	RR-Rotation With Sub-Trees . . . . .	12
7.3	RL-Rotation With Sub-Trees . . . . .	12
7.4	LR-Rotation With Sub-Trees . . . . .	12
<b>8</b>	<b>How AVL Trees Generated?</b>	<b>13</b>



# 1 Introduction to AVL Trees

1. what is a Binary Search Tree (BST)
2. Drawback of BST
3. How BST can be improved
4. what is an AVL Tree
5. Rotation in AVL Tree
6. How to create AVL Tree

## 2 Binary Search Tree



- All the elements left hand side of any node is smaller than that
- All the elements right hand side of any node is bigger than that

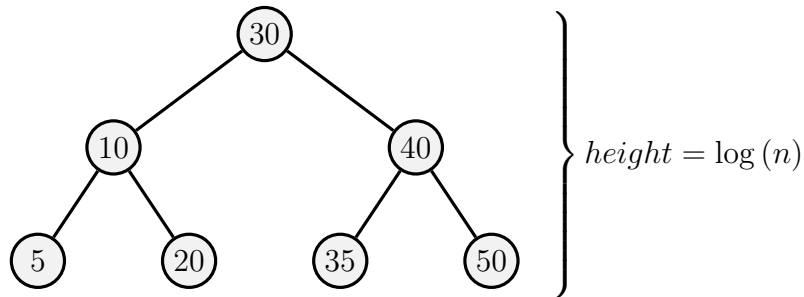
we use Binary Search Tree for optimize searching

Maximum Number of Comparision depends on the Height of the Binary Search Tree

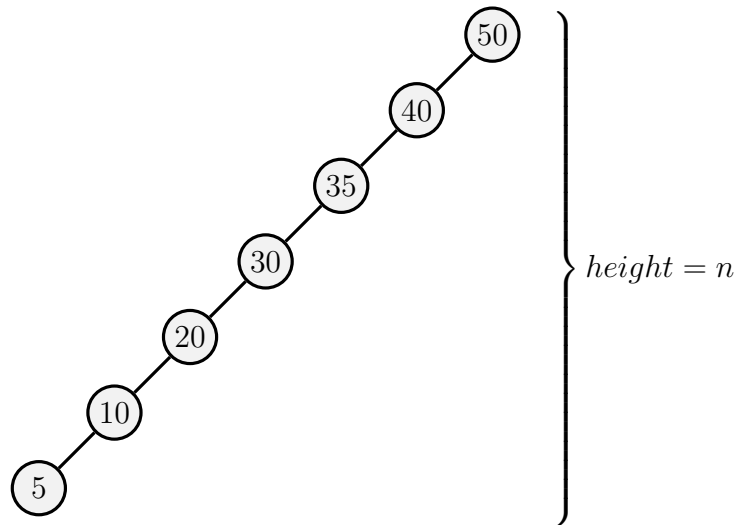
$$\text{Height of the Binary Search Tree} \Rightarrow \begin{cases} \text{Minimum} \Rightarrow \log(n) \\ \text{Maximum} \Rightarrow n \end{cases}$$

### 3 Example of Creating Binary Search Tree

keys : 30, 40, 10, 50, 20, 5, 35



keys : 50, 40, 35, 30, 20, 10, 5



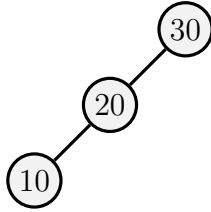
Height of Binary Search Tree Depends on how keys are inserted

### 4 Can We Improve Binary Search Tree?

Suppose we have 3 keys like : 30, 20, 10 Then We Have These 5 following Shapes for making Binary Search Tree

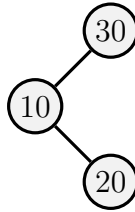
#### 4.1 Order-1

order : 30, 20, 10



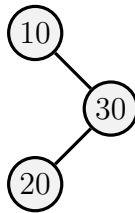
#### 4.2 Order-2

order : 30, 10, 20



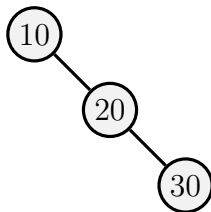
#### 4.3 Order-3

order : 10, 30, 20



#### 4.4 Order-4

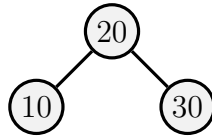
order : 10, 20, 30



## 4.5 Order-5

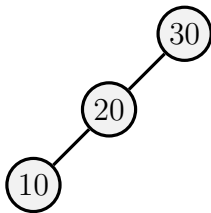
order : 20, 10, 30

order : 20, 30, 10

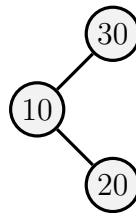


## 4.6 summary

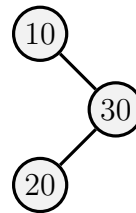
30, 20, 10



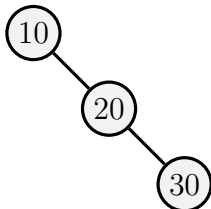
30, 10, 20



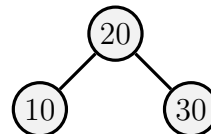
10, 30, 20



10, 20, 30



20, 10, 30



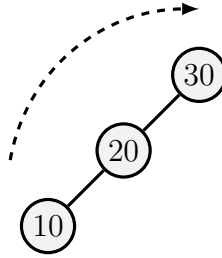
20, 30, 10

Can We Improve Binay Search Tree?

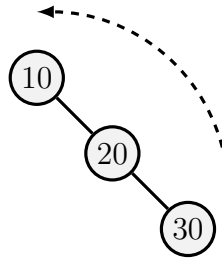
Is There Any Way to Convert The Order-[1-4] to Order-5 ?

Answer : Yes , We define Rotations

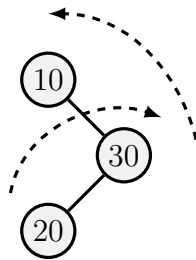
#### 4.7 LL-Rotation



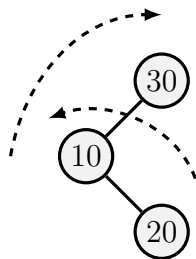
#### 4.8 RR-Rotation



#### 4.9 RL-Rotation



#### 4.10 LR-Rotation



There was only 3 Nodes What if we have Many Nodes ?

## 5 What is AVL Tree?

AVL Tree is Height Balanced Binary Search Tree

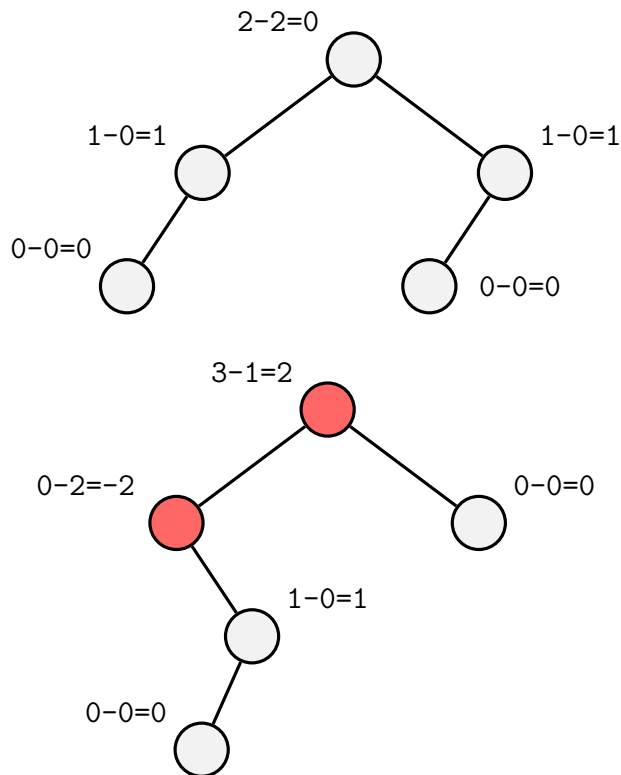
for balancing the height of Binary Search Tree we define One factor that is balance factor

balance factor = height of left subtree - height of right subtree

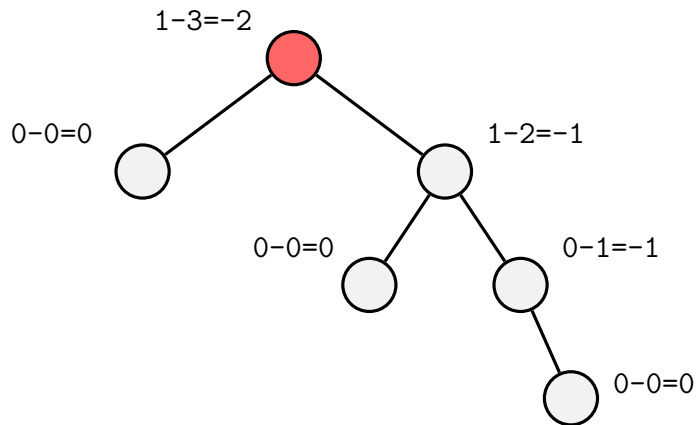
$$b_f = h_l - h_r = \{-1, 0, 1\}$$

$$|b_f| = |h_l - h_r| \leq 1$$

### 5.1 Examples of finding balance factor of a node

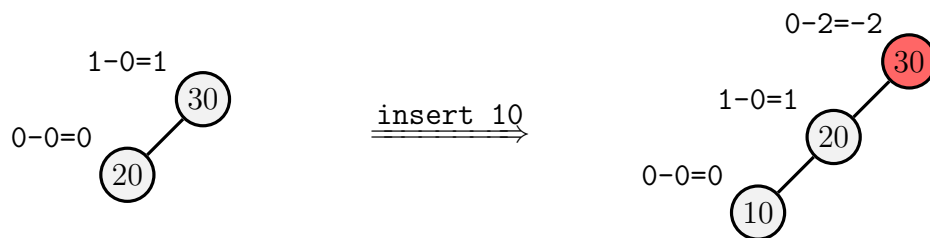




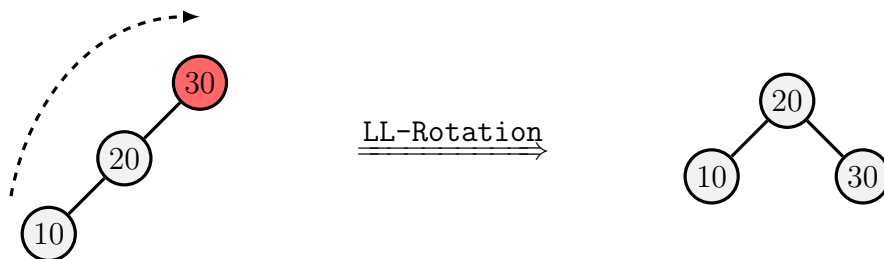


## 6 Types Of Rotations

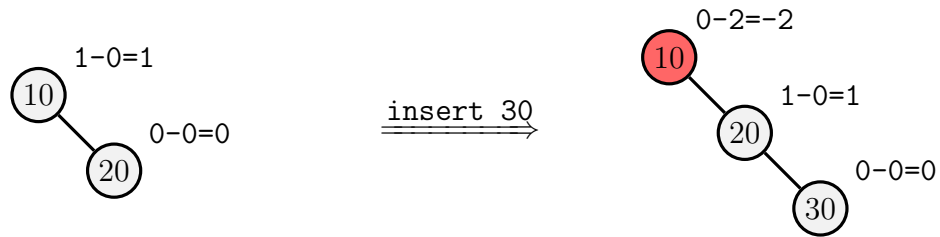
### 6.1 LL-Rotation



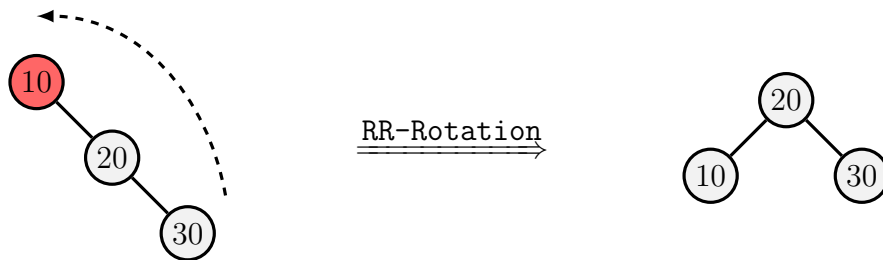
30 became unbalanced because we insert 10 at left and left so we called it LL-Rotation .



## 6.2 RR-Rotation



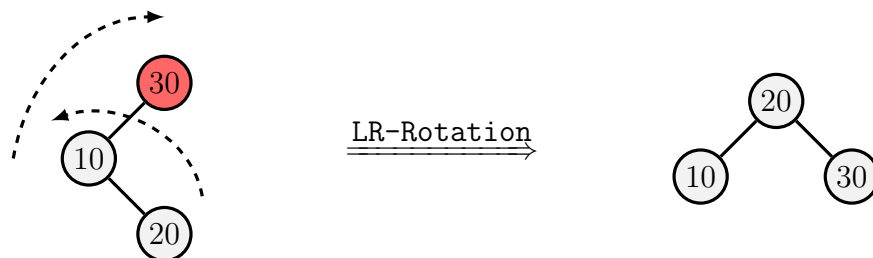
Node 10 became unbalanced because we insert 30 at right and right so we called it RR-Rotation .



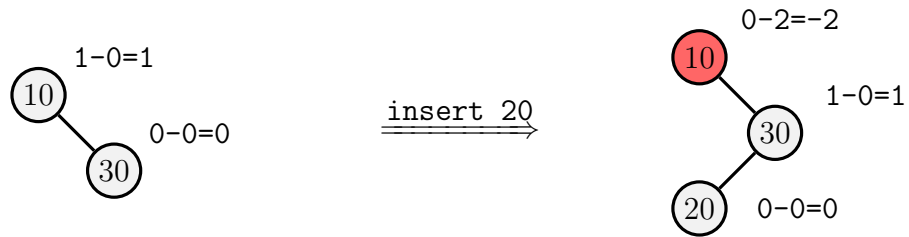
## 6.3 LR-Rotation



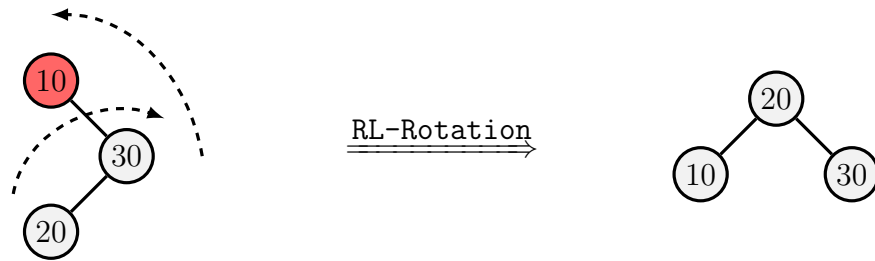
Node 30 became imbalanced because we insert 20 at left then right so we called it LR-Rotation .



## 6.4 RL-Rotation

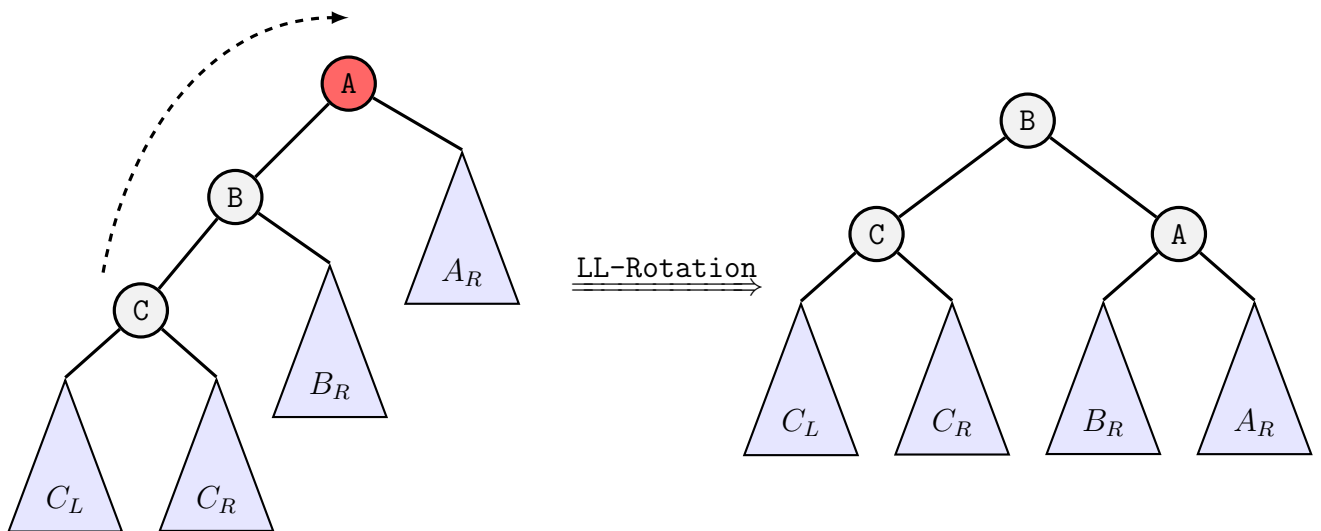


Node 10 became unbalanced because we insert 20 at right and right so we called it RR-Rotation .

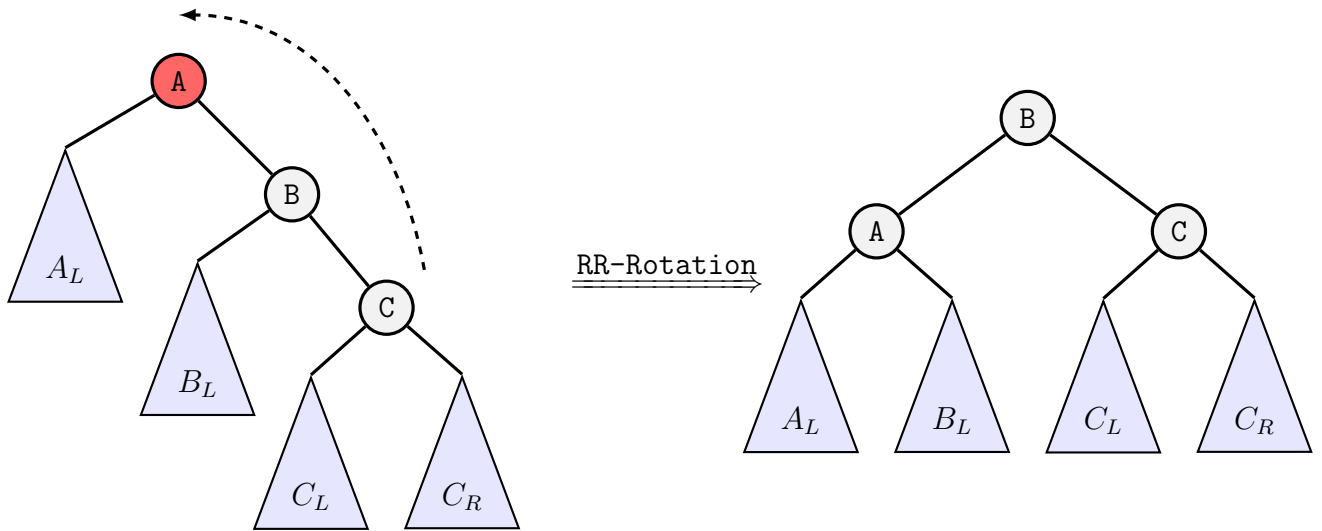


## 7 Rotations With Sub-Trees

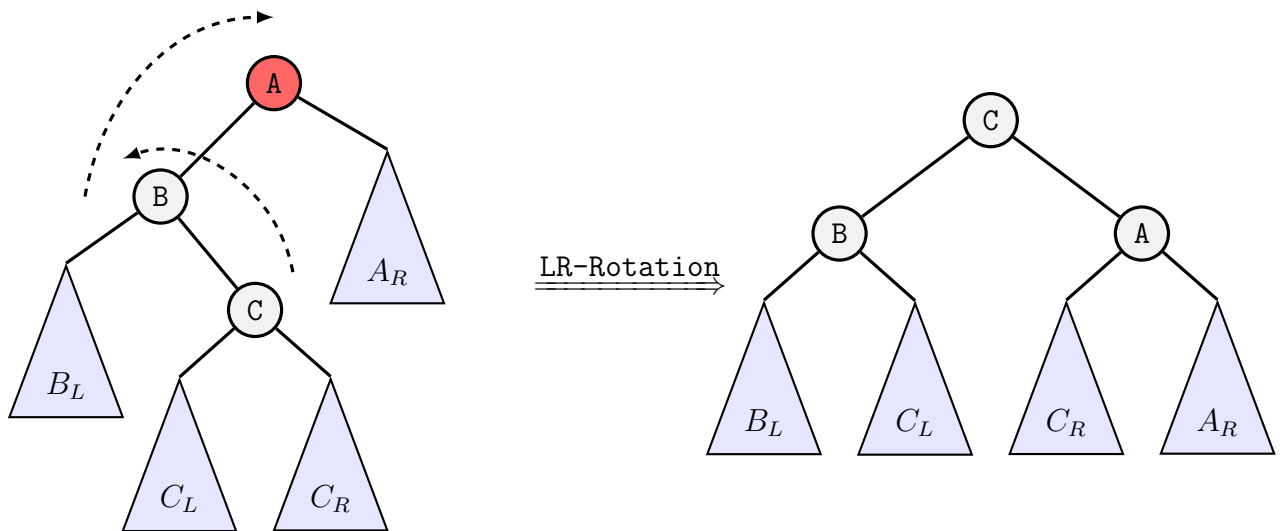
### 7.1 LL-Rotation With Sub-Trees



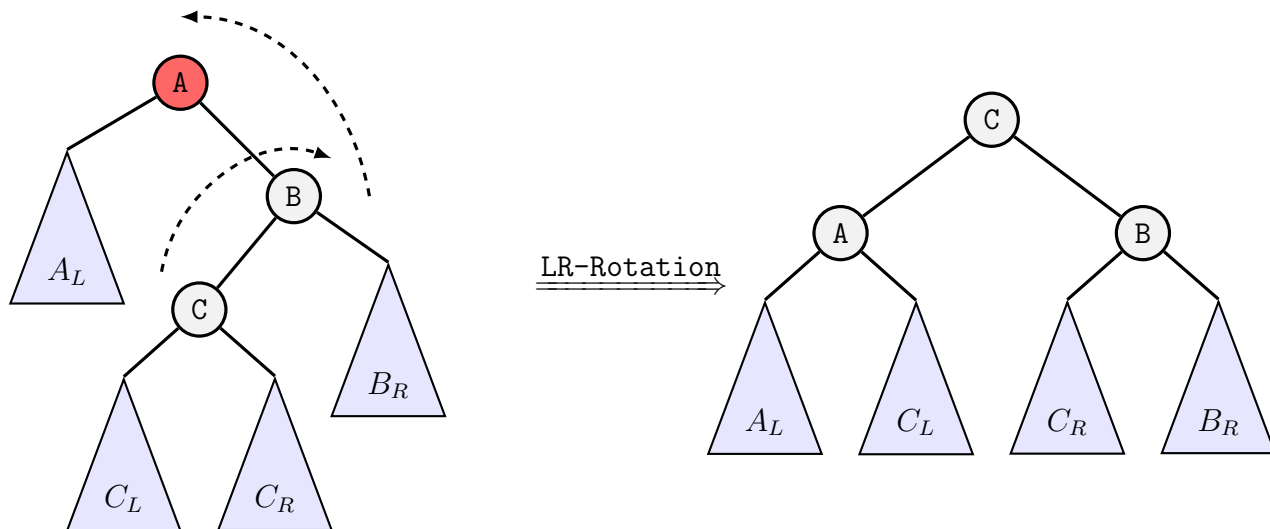
## 7.2 RR-Rotation With Sub-Trees



## 7.3 RL-Rotation With Sub-Trees

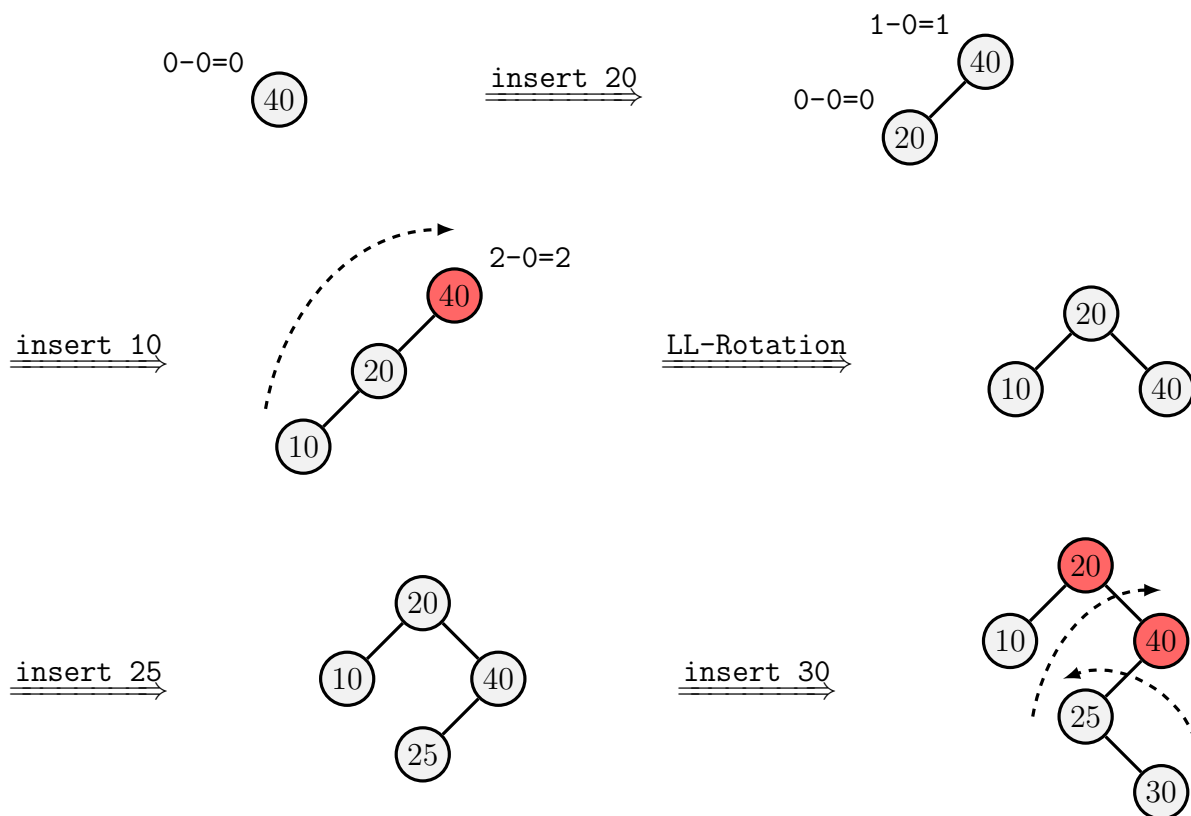


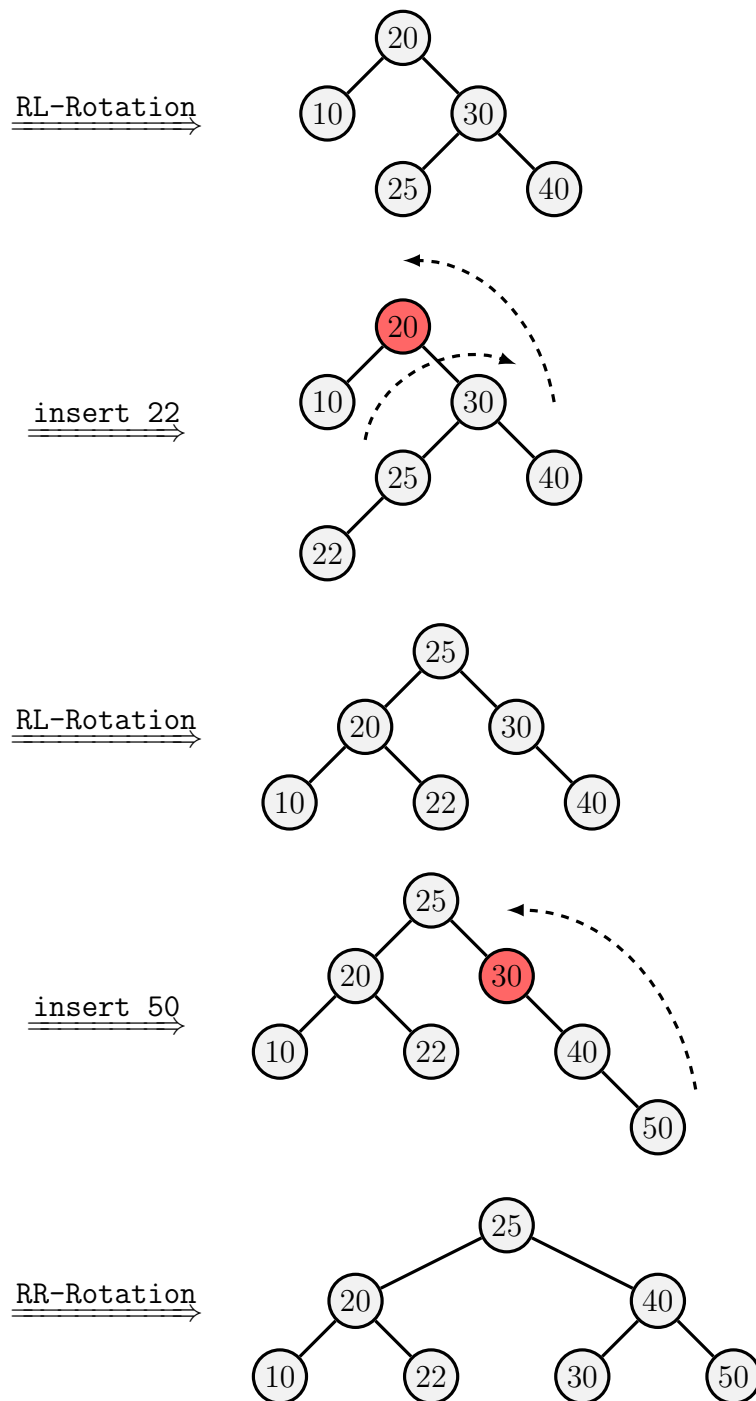
## 7.4 LR-Rotation With Sub-Trees



## 8 How AVL Trees Generated?

keys : 40, 20, 10, 25, 30, 22, 50





Don't Allow Any Node To Exceed the balance Factor From -2 or +2

You Should Never get -3 or 3

If the Node Become imbalance then perform Rotation and Balance the Tree

## 9 Example

