AVL Trees

Module 7

Problems in BST

- Height can vary from $O(\log n)$ to $O(n)$ depending on the order of insertion
- So is search time
- Solution → Height balanced trees
- Examples → AVL Tree, Red-Black Tree etc.
- How to check whether a tree is height balanced?
- Compute balance factor for each node

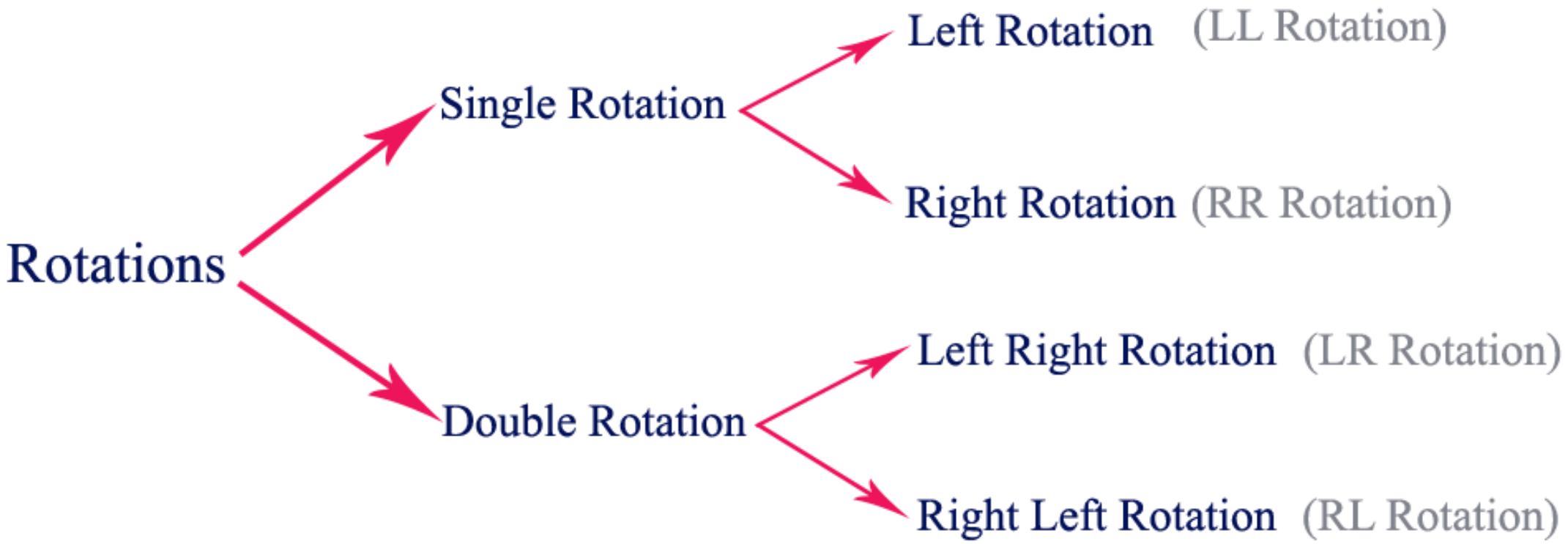
Balance Factor

- $bf = h_l - h_r$
- h_l → height of the node induced by the left subtree
- h_r → height of the node induced by the right subtree
- If $bf = \{-1,0,1\}$ for all nodes in the tree, then the tree is height balanced

AVL Tree insertion/creation

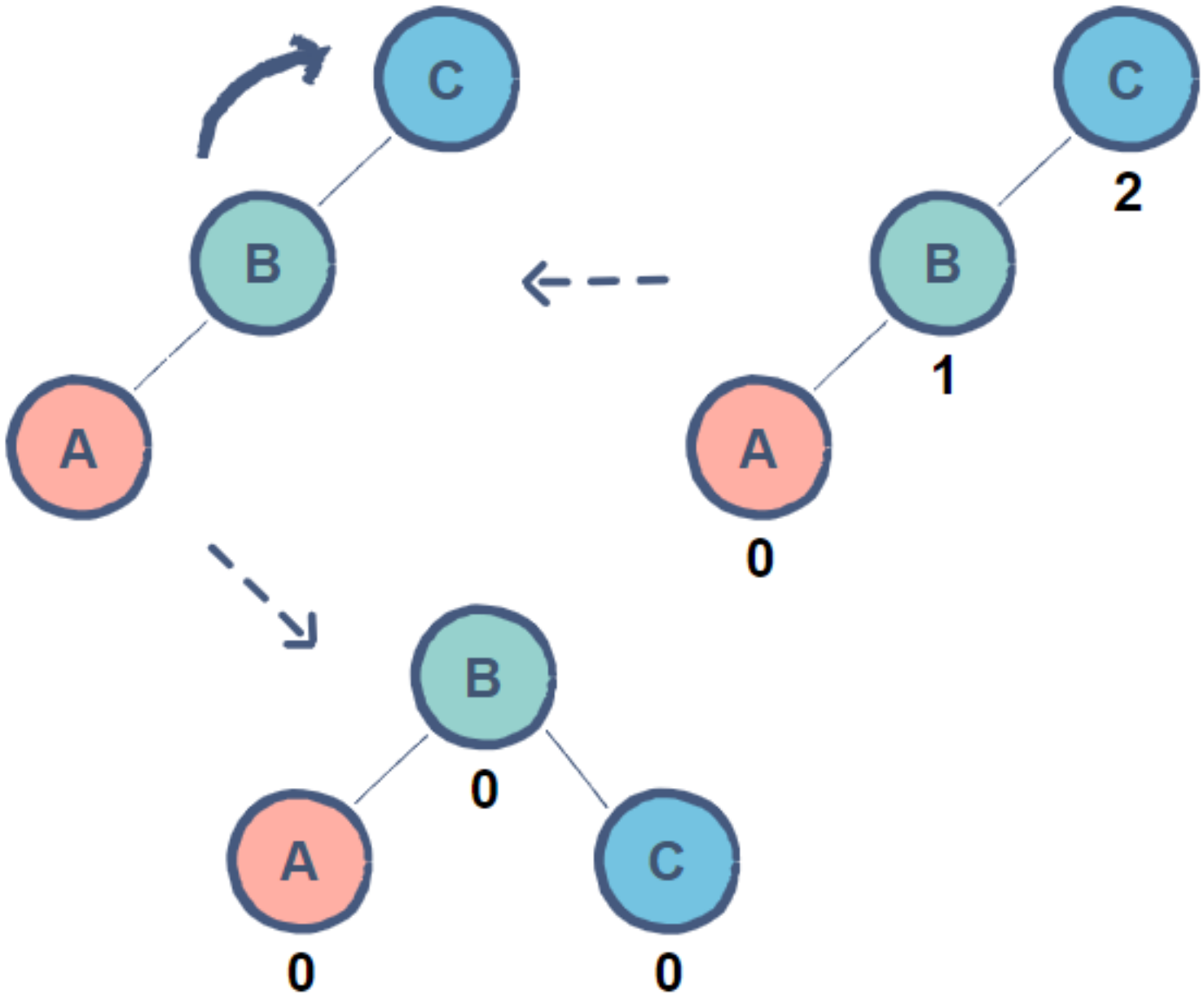
- AVL ➔ Georgy Adelson Velsky and Evgenii Landis
- Insert nodes in the incoming order
- After each insertion, check balance factor of each node
- If $bf > 1$ or < -1 perform any one of the rotations depending upon how the imbalance is created

AVL Tree Rotations

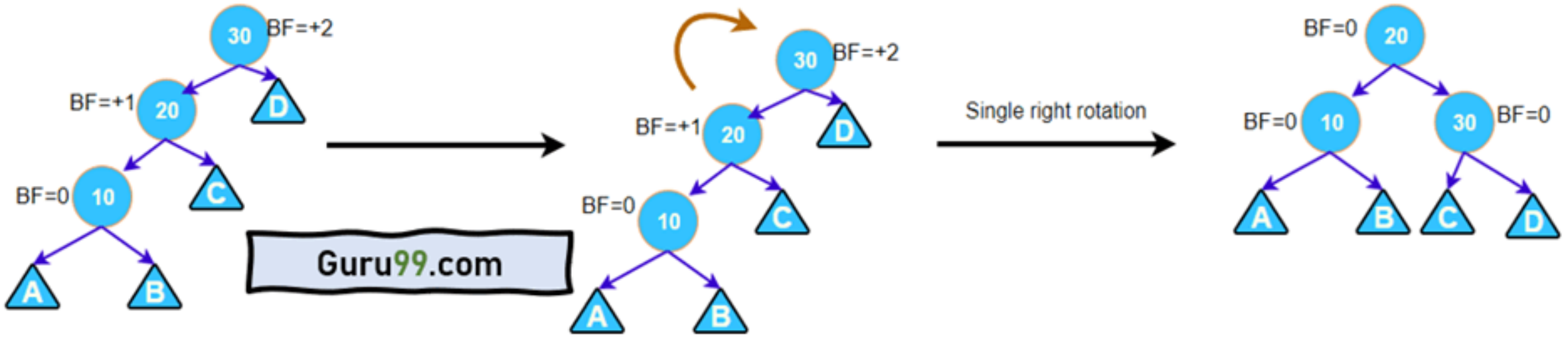


LL Imbalance/Right Rotation (Single)

- This rotation is performed when a new node is inserted at the left child of the left subtree.
- Creates a Left-Left Imbalance
- Perform Right rotation (Clockwise) on the critical node
- It will be a single rotation to the right
- Critical node is the node that has a balance factor $\neq \{-1, 0, 1\}$


C
B
A
C
2
B
1
A
0
B
0
A
0
C
0

Right Rotation with subtrees



Tree is imbalance as BF(30) = +2

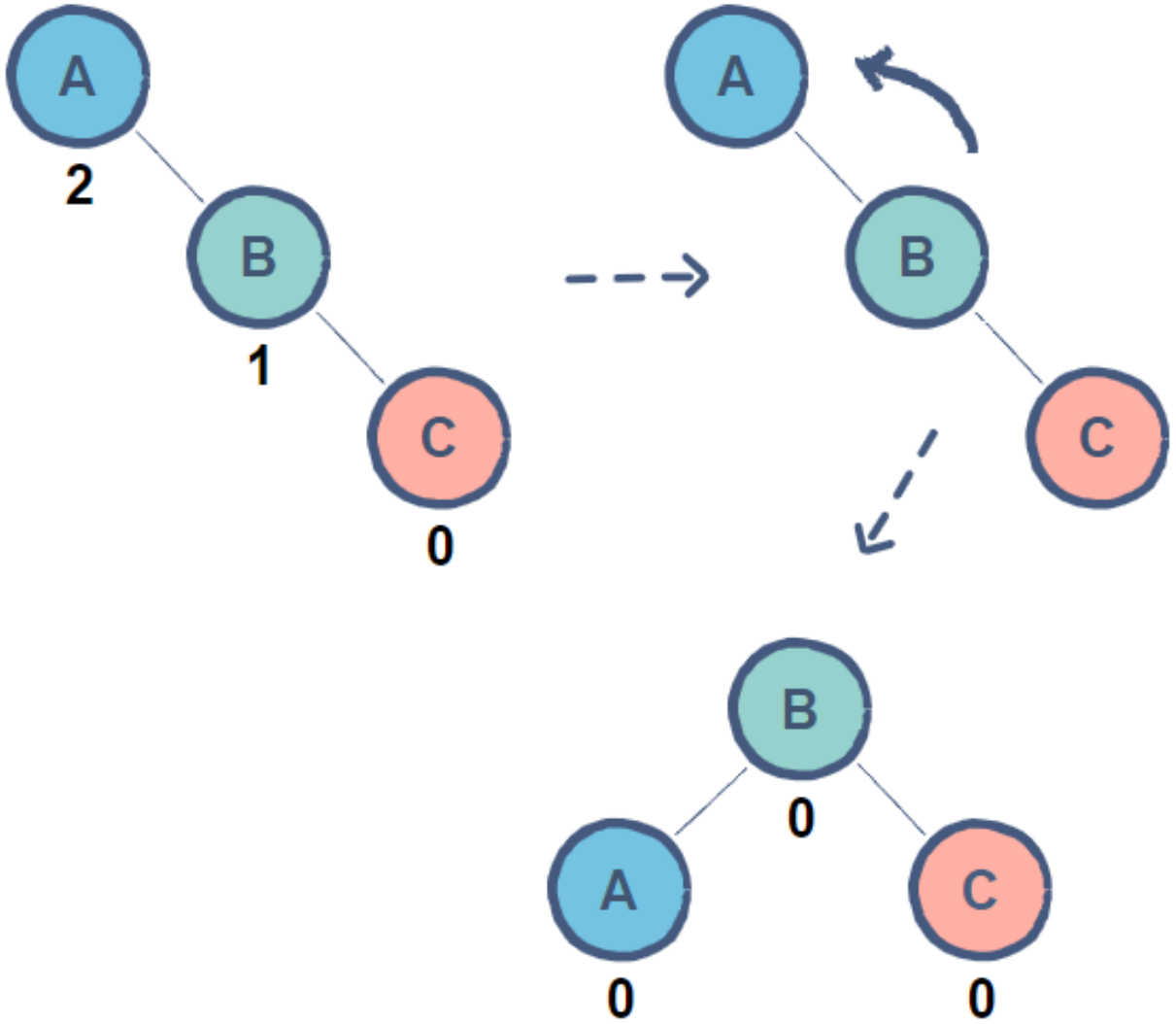
Perform a right rotation at node 20

Now the tree is balanced.

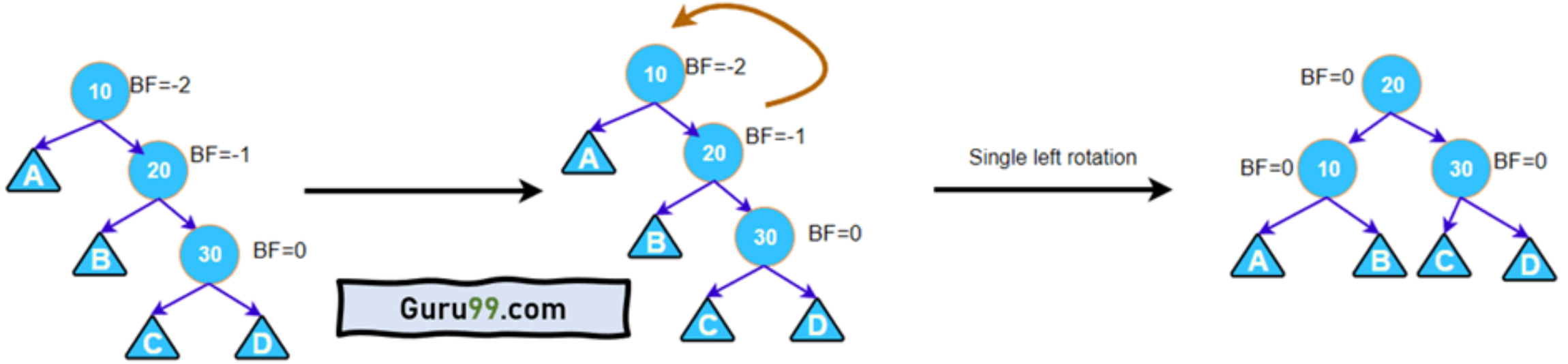
RR Imbalance/Left Rotation (Single)

- This rotation is performed when a new node is inserted at the right child of the right subtree.
- Creates a Right-Right Imbalance
- It will be a single rotation to the left
- Perform Left Rotation (Counter-clockwise) on the critical node

Left Rotation



Left Rotation with Subtrees



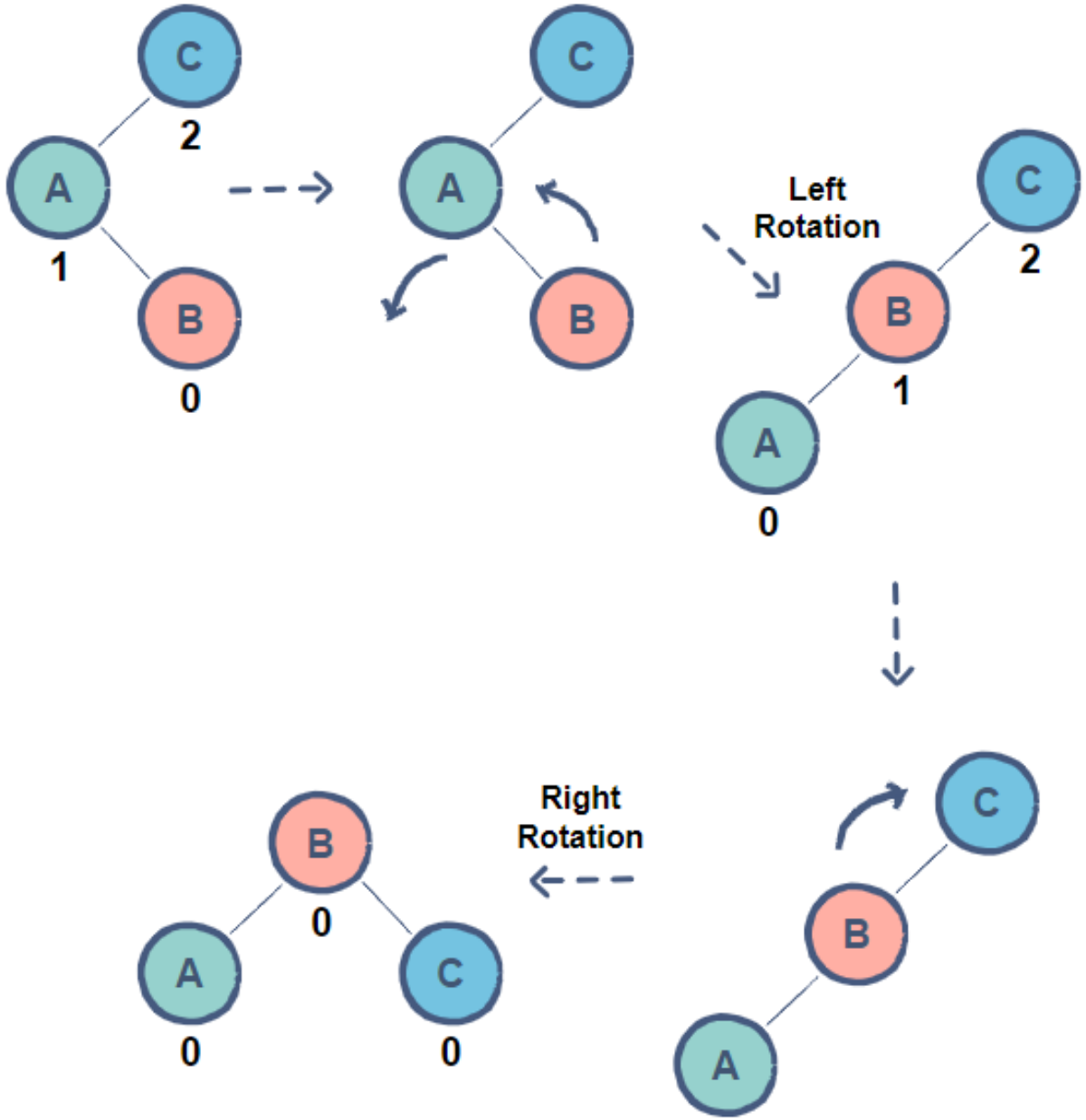
Tree is imbalance as BF(10) = -2

Perform a left rotation at node 20

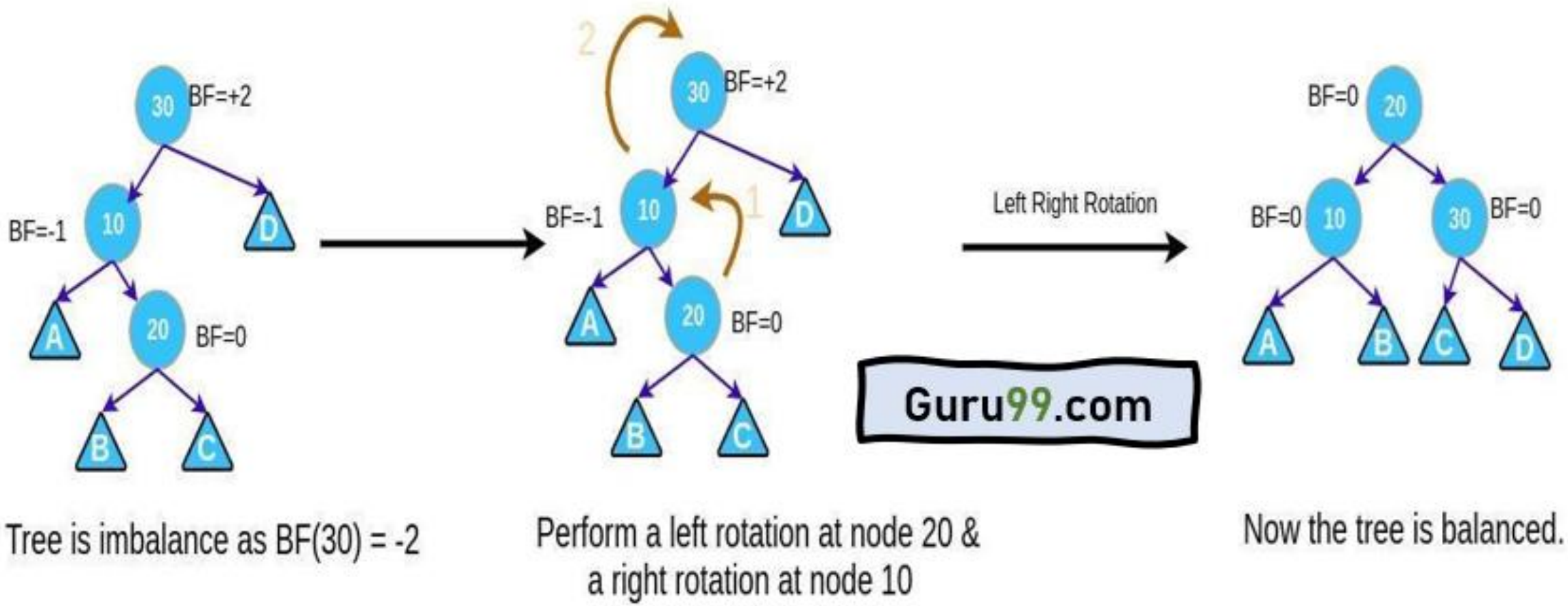
Now the tree is balanced.

LR Imbalance (Left and Right → Double)

- This rotation is performed when a new node is inserted at the left child of the right subtree.
- Creates a Left-Right Imbalance
- Left-Right Rotation is the combination of a left rotation and a right rotation
- Perform Left Rotation first and Right Rotation next

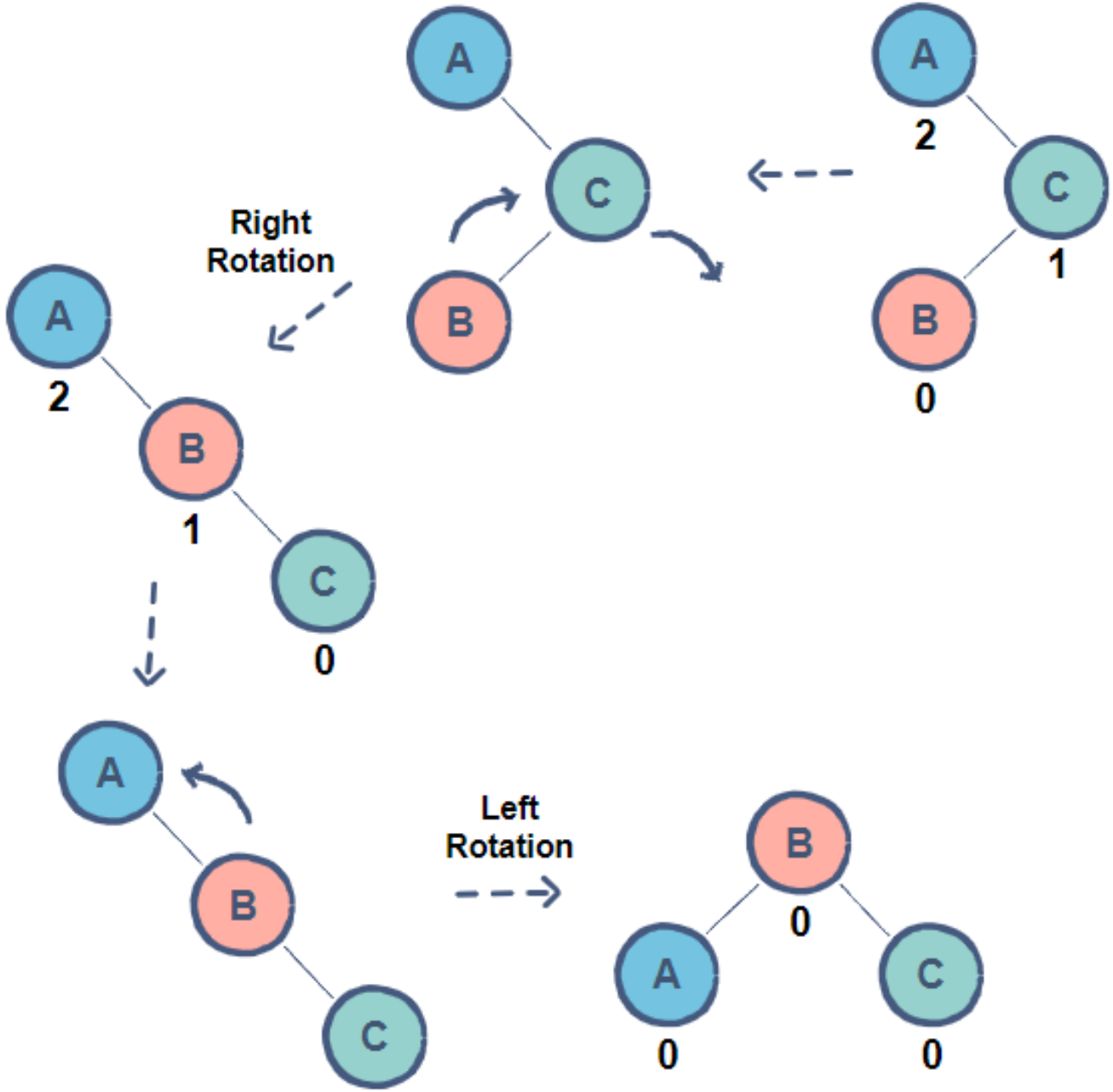

C
2
A
1
B
0
C
A
B
Left
Rotation
C
2
B
1
A
0
C
B
A
Right
Rotation
B
0
A
0
C
0

LR Rotation with subtrees

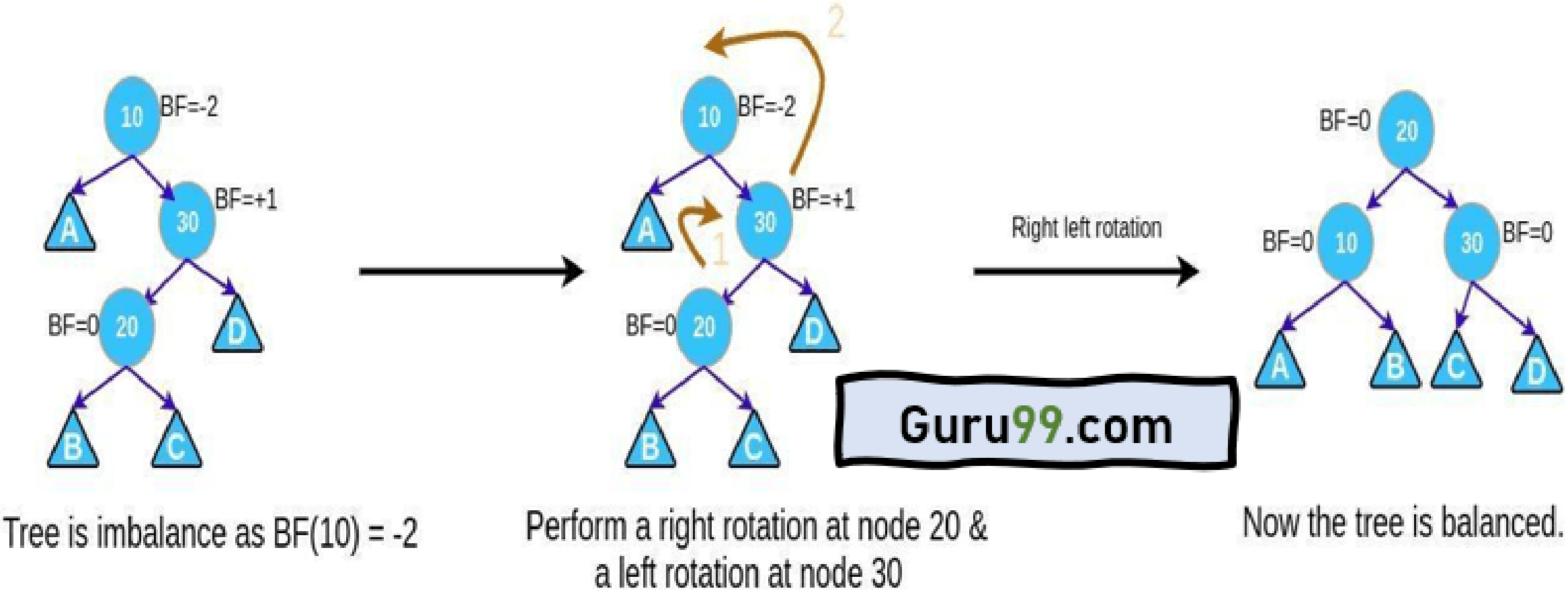


RL Imbalance (Right and Left → Double)

- This rotation is performed when a new node is inserted at the right child of the left subtree.
- Creates a Right-Left Imbalance
- Right-Left Rotation is the combination of a right rotation and a left rotation
- Perform Right Rotation first and Left Rotation next


A
C
B
Right
Rotation
A
2
C
1
B
0
A
2
B
1
C
0
A
B
C
Left
Rotation
B
0
A
0
C
0

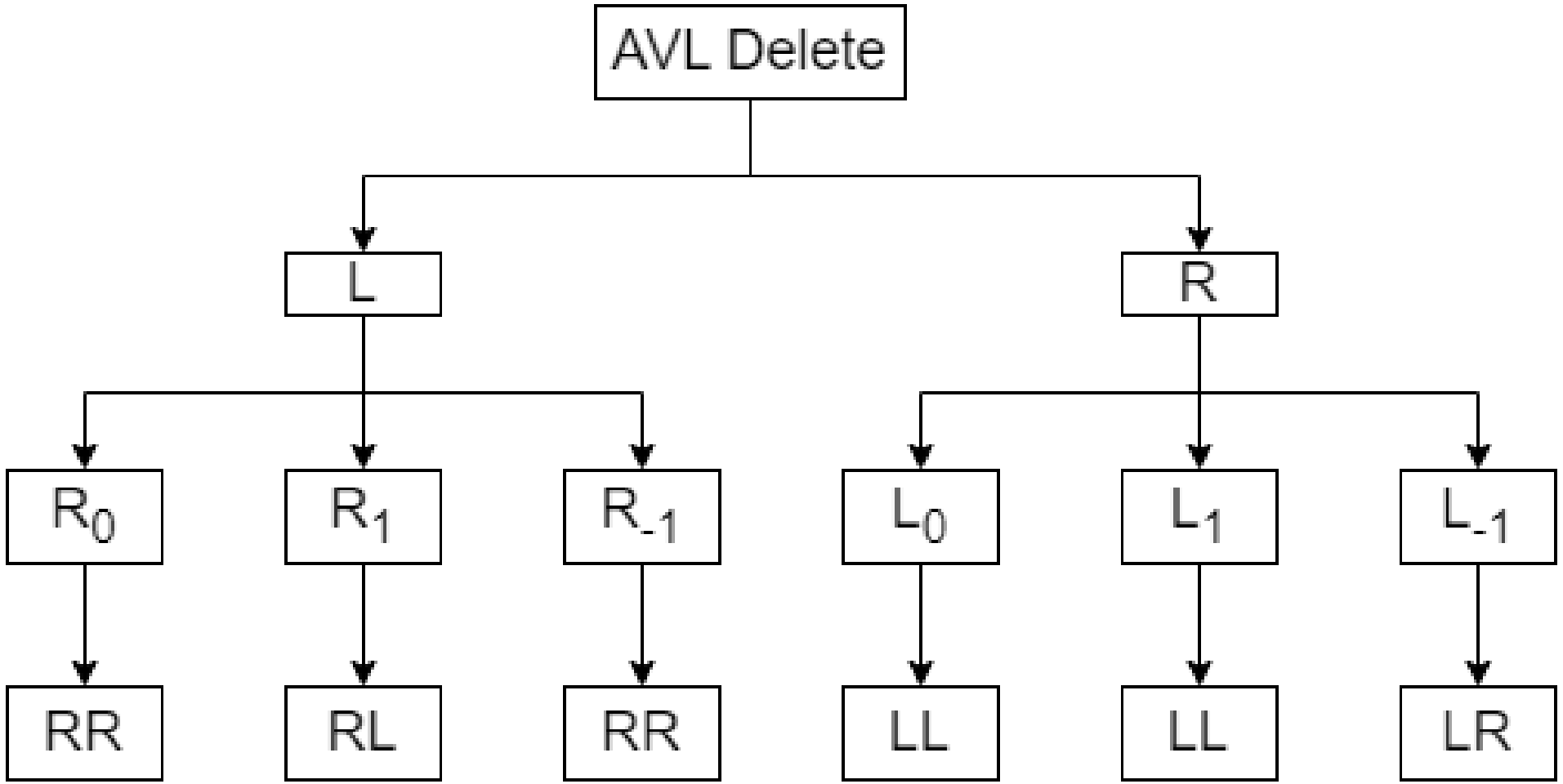
RL Rotation with subtrees



AVL Tree Insertion Example

- Eg1: Construct an AVL Tree with the following sequence
- 14,17,11,7,53,4,13,12,8,60,19,16,20

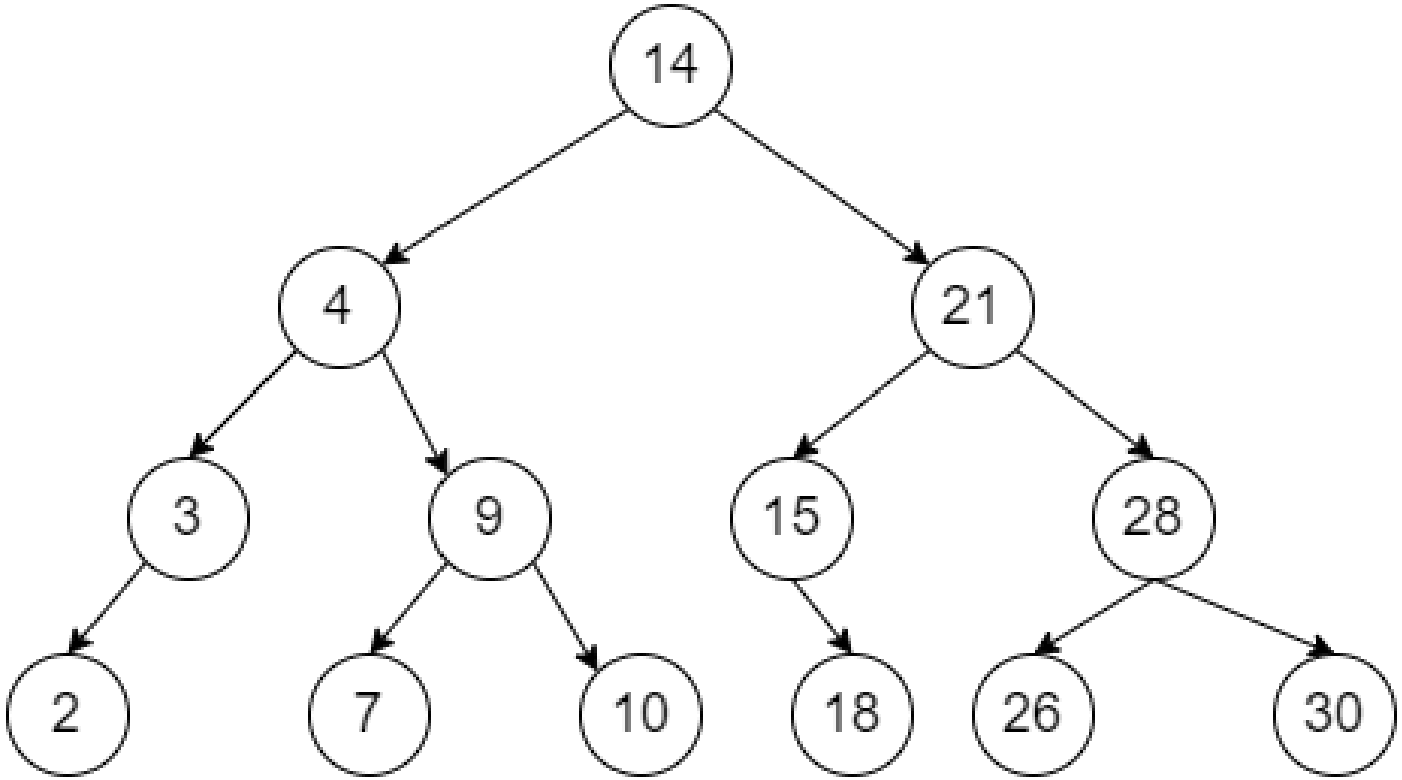
AVL Tree Deletion



AVL Tree Deletion Examples

- Delete 14,13,17,10 in sequence from the AVL tree created in the previous Eg 1
- Construct an AVL tree with the sequence 21,26,30,9,4,14,28,18,15,10,2,3,7 and delete 2,3,10,18,4,9,14,7,15 in sequence from it

AVL Tree Deletion Eg-2



- Delete 2,3,10,18,4,9,14,7,15 in sequence

Pseudocodes for AVL Tree Rotations

- Will be included in DA1
- Expect for FAT

Thank you..