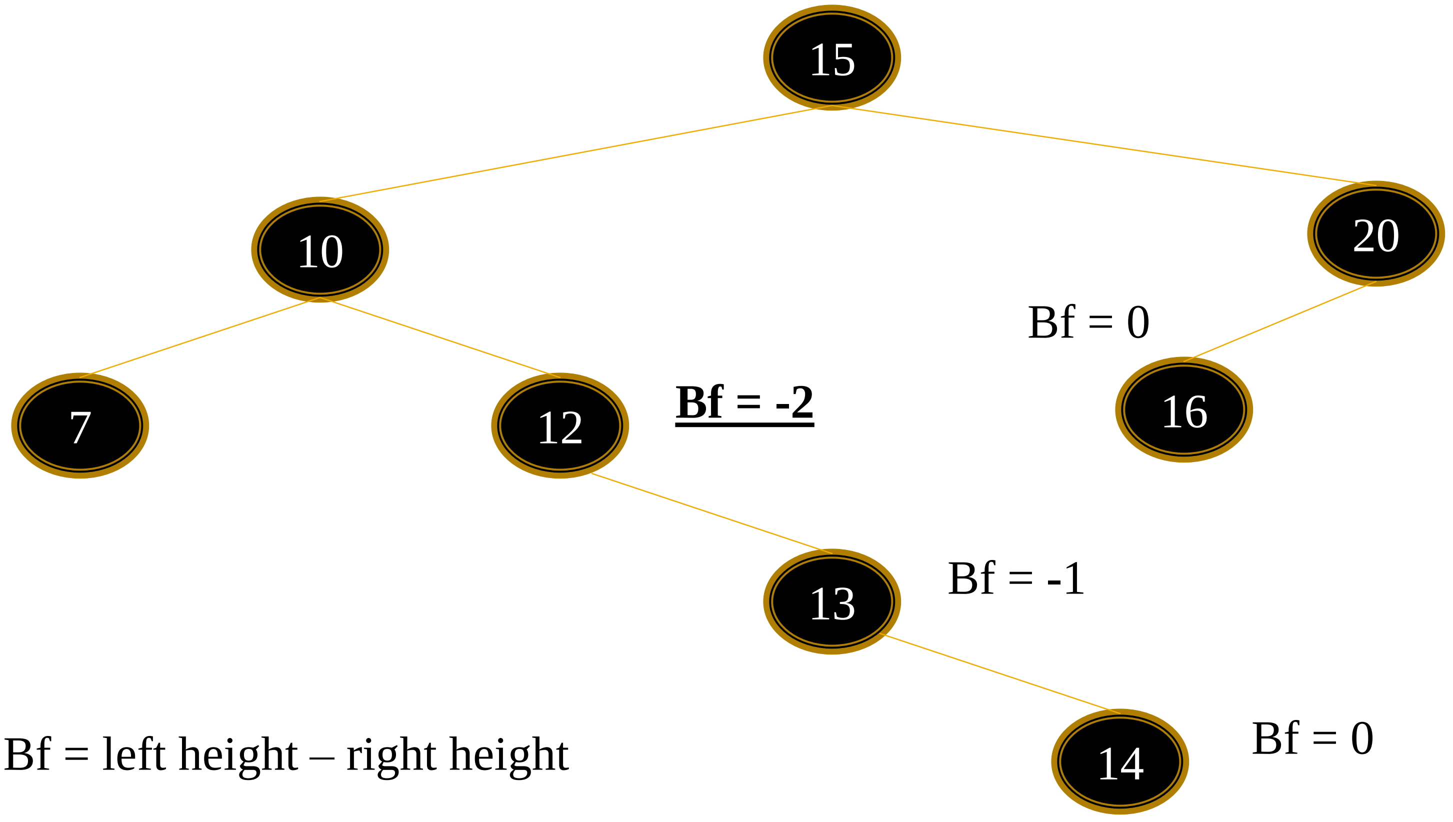
AVL Tree

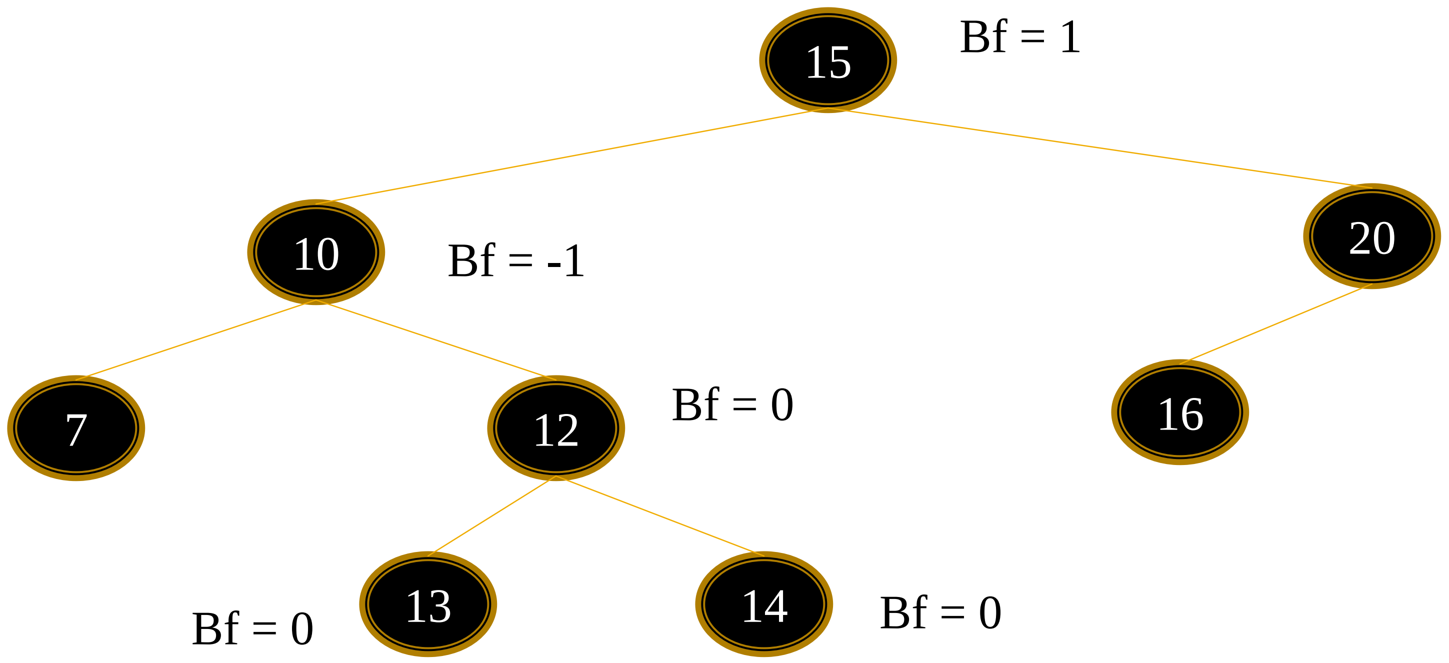
By Mostafa Saad

AVL Balance Factor

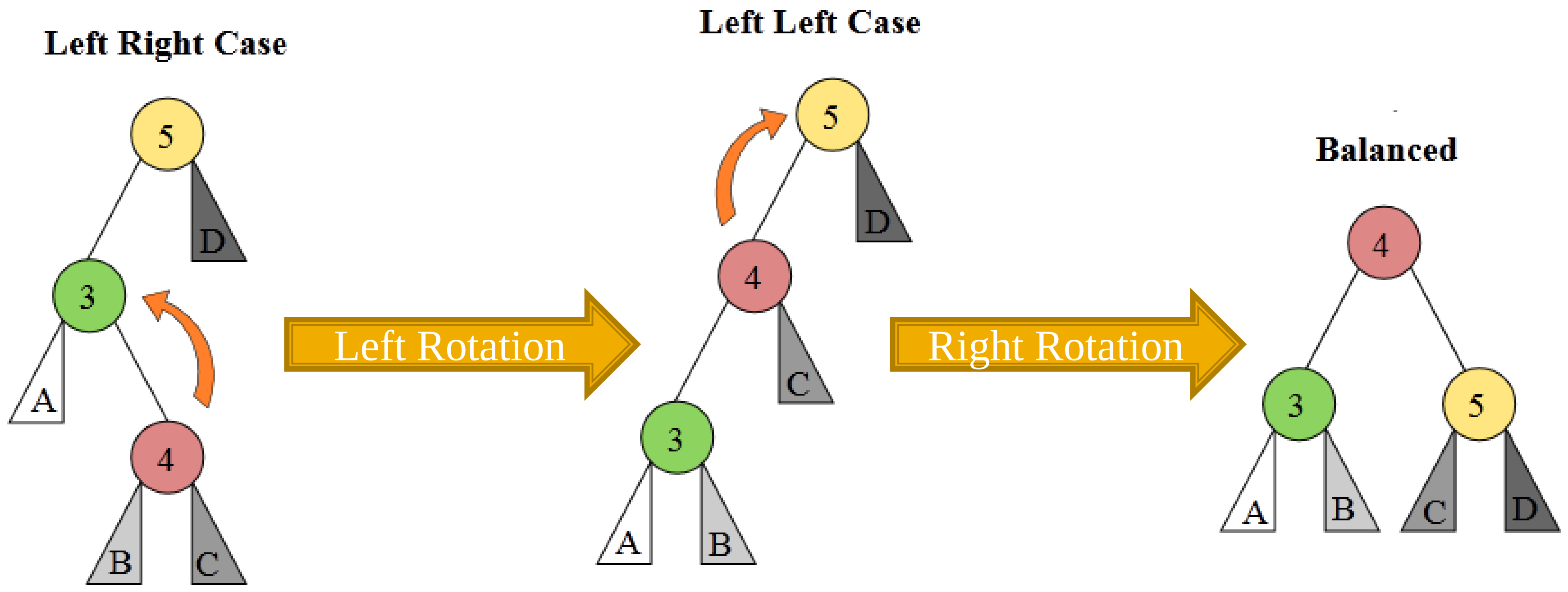


Bf = left height – right height

AVL Balance Factor



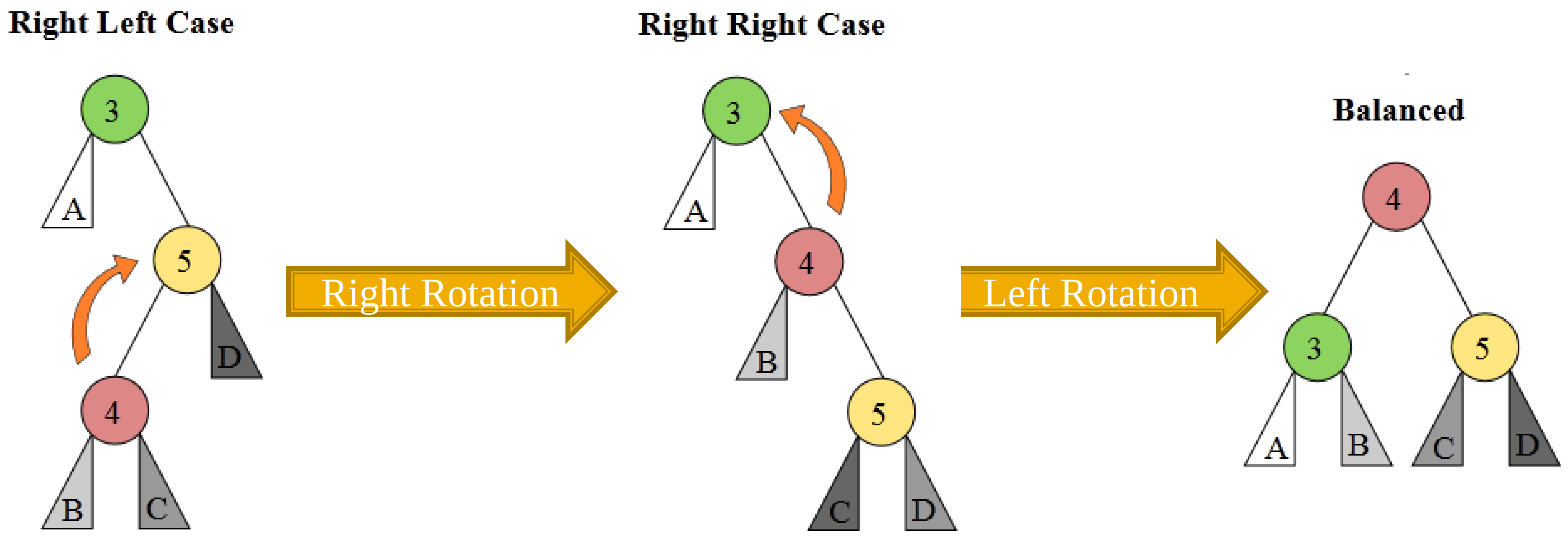
LR => LL => B



Imagine that: A, B, C, D are possible 4 sub-trees
We would like to do rotations to balance the 3 nodes causing a problem
But keep the BST property correct

You may think: A = 2, B = 3.5, C = 4.5, D = 6

RL => RR => B



If you checked to the A, B, C, D order in the 3 shapes, you will find them sorted

The general flow

- 1- Add the element to the BST in normal way
- 2- Let current node = just added one
- 3- Calculate BF
- 4- if |BF| > 1, we have an AVL unbalance
  - If we are in case (3 or 4), convert to case (1 or 2)
  - If new in case 1 or 2, handle them
- Let current node = parent
- Go to 3

Delete

- Same steps as in BST
- Start from where the node deleted (according to the 3 cases) and behave as the insertion in AVL(calc BF and do rotations if |BF| > 1)

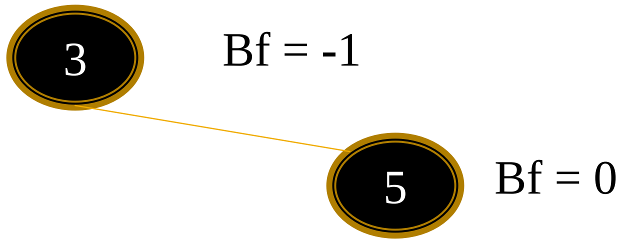
Let’s simulate a big example

- Build following AVL tree for input
- 3 5 9 1 0 2 6 10 7 4 8

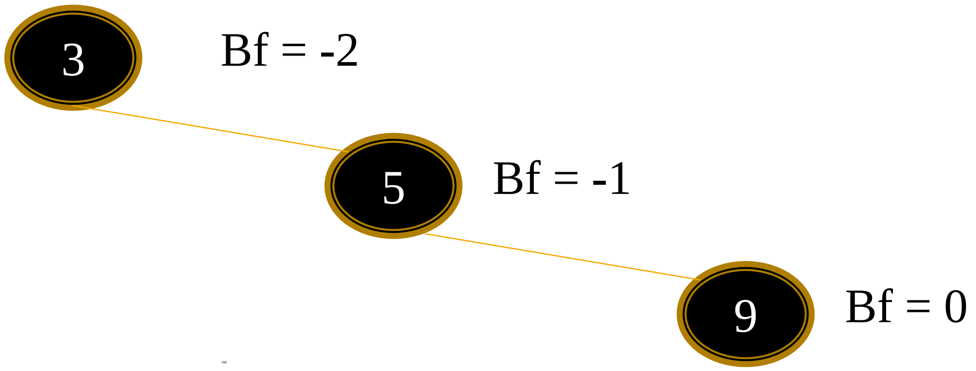
3 [5 9 1 0 2 6 10 7 4 8]

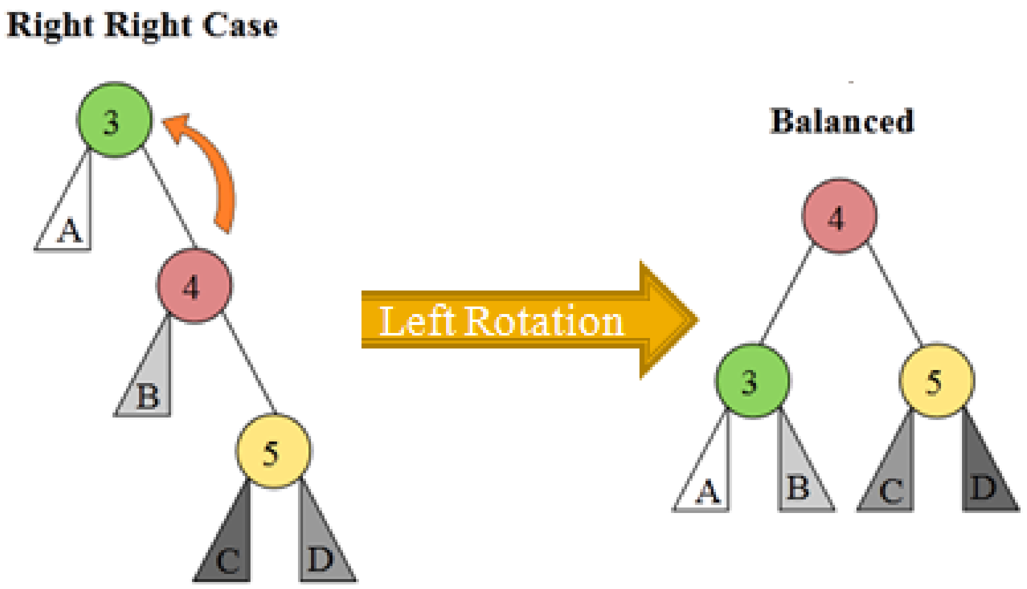
Bf = 0

3 5 [9 1 0 2 6 10 7 4 8]

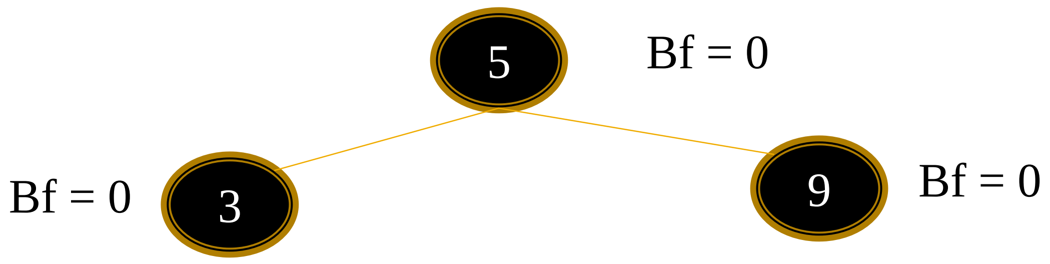


3 5 9 [1 0 2 6 10 7 4 8]

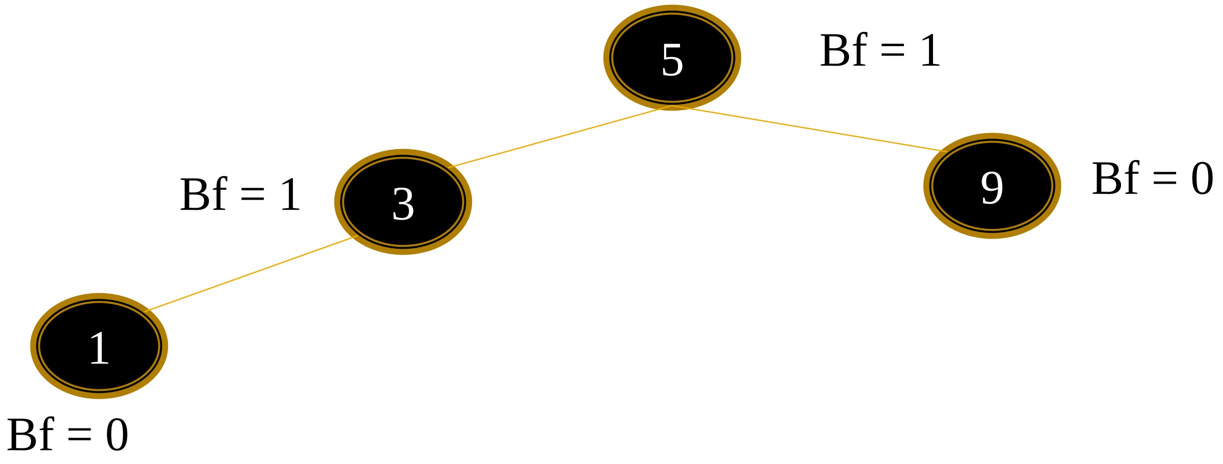




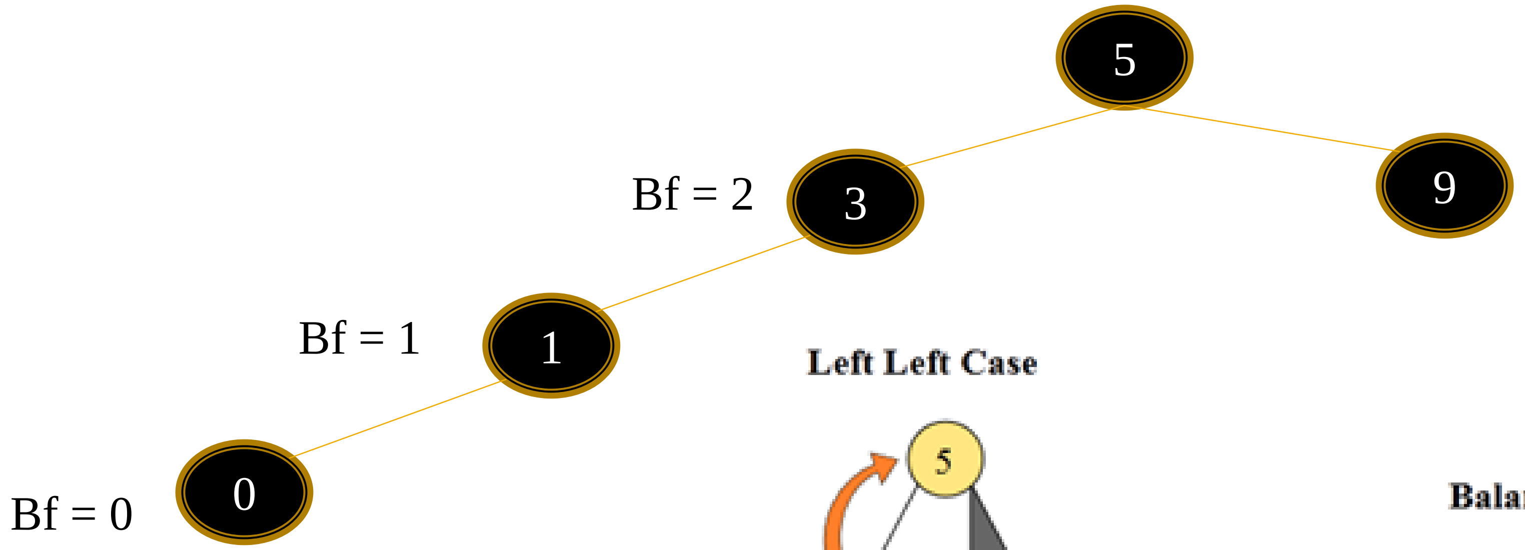
3 5 9 [1 0 2 6 10 7 4 8]

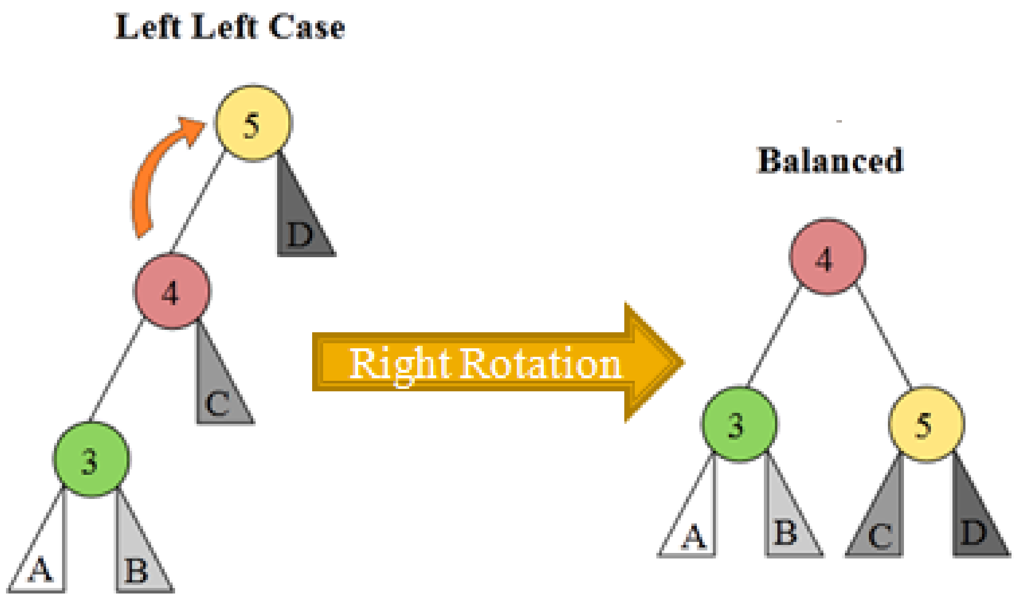


3 5 9 1 [0 2 6 10 7 4 8]

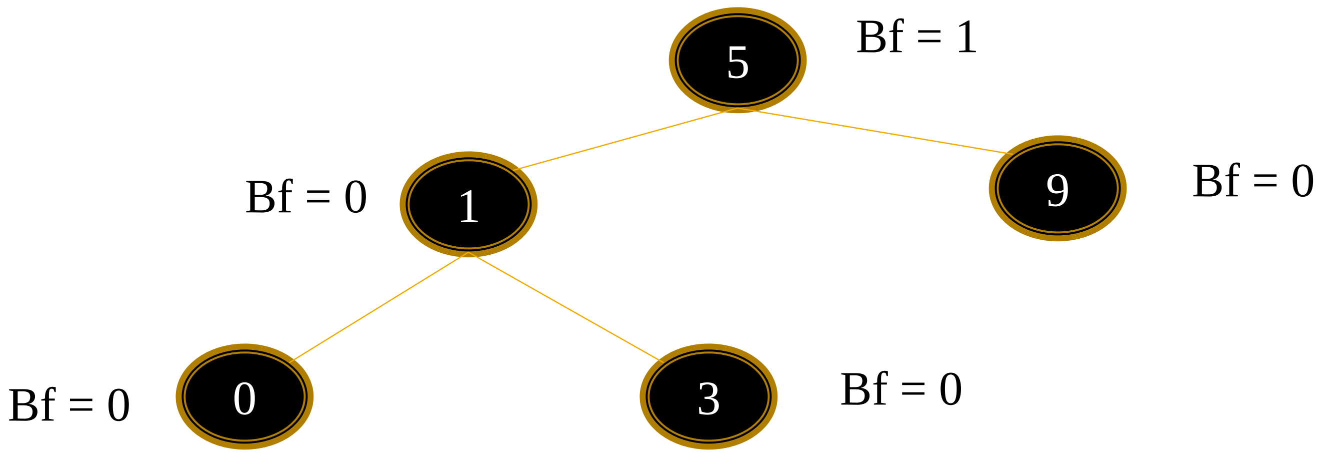


3 5 9 1 0 [2 6 10 7 4 8]

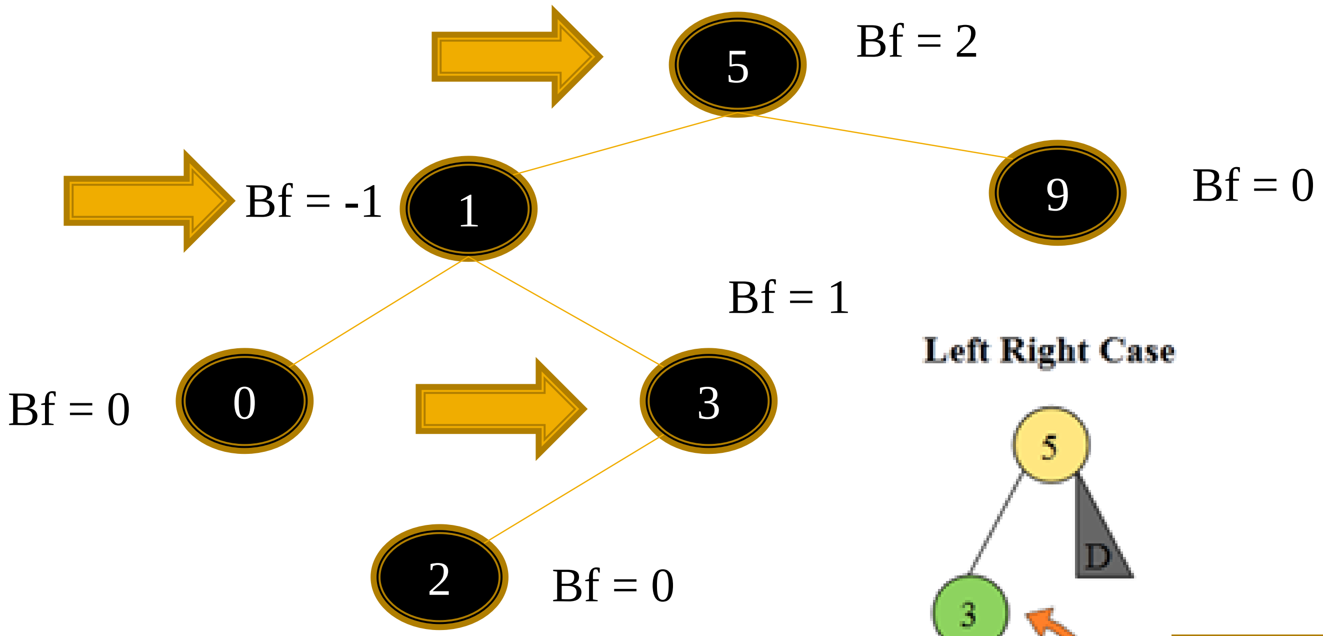




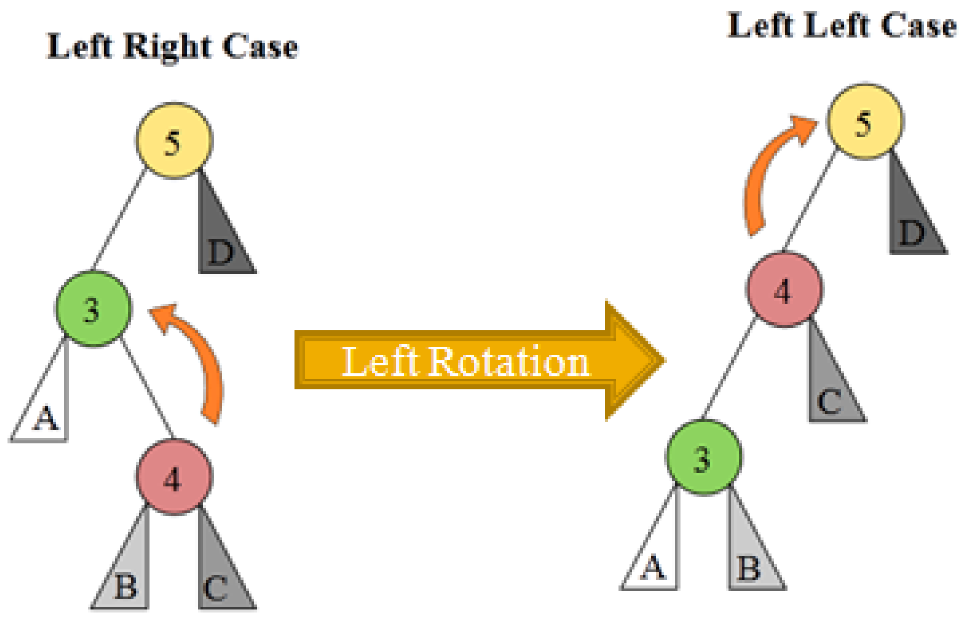
3 5 9 1 0 [2 6 10 7 4 8]



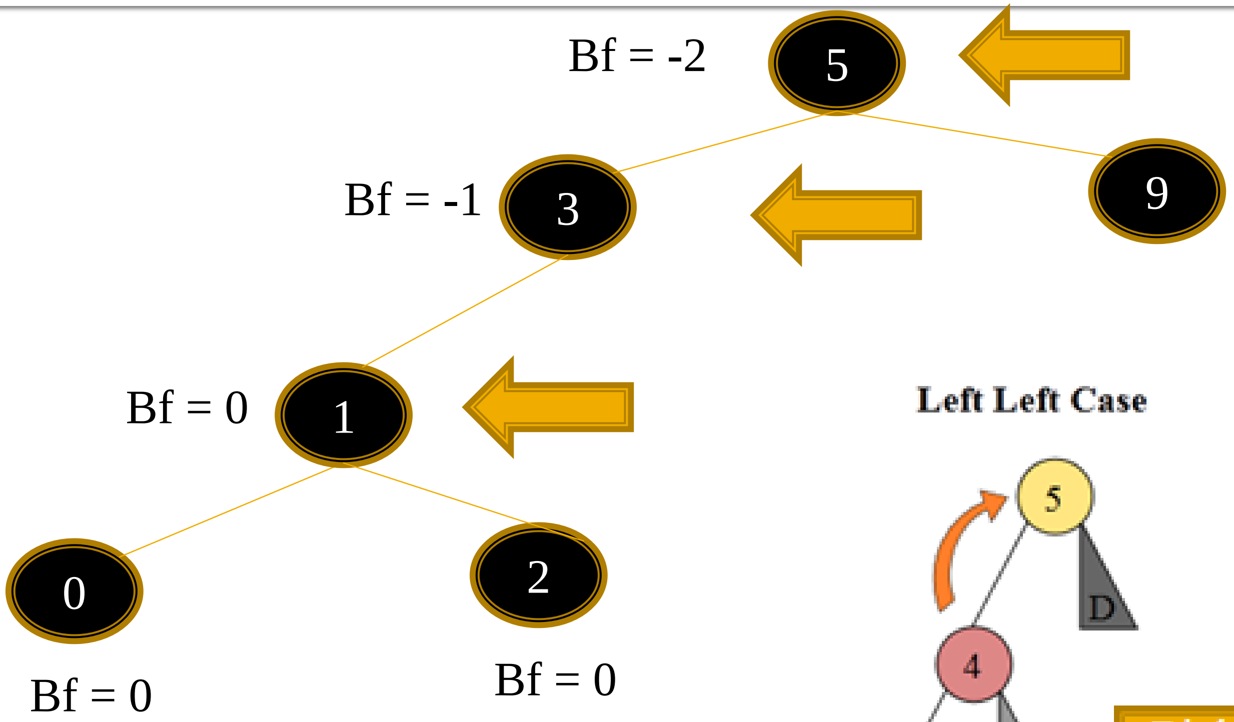
3 5 9 1 0 2 [6 10 7 4 8]

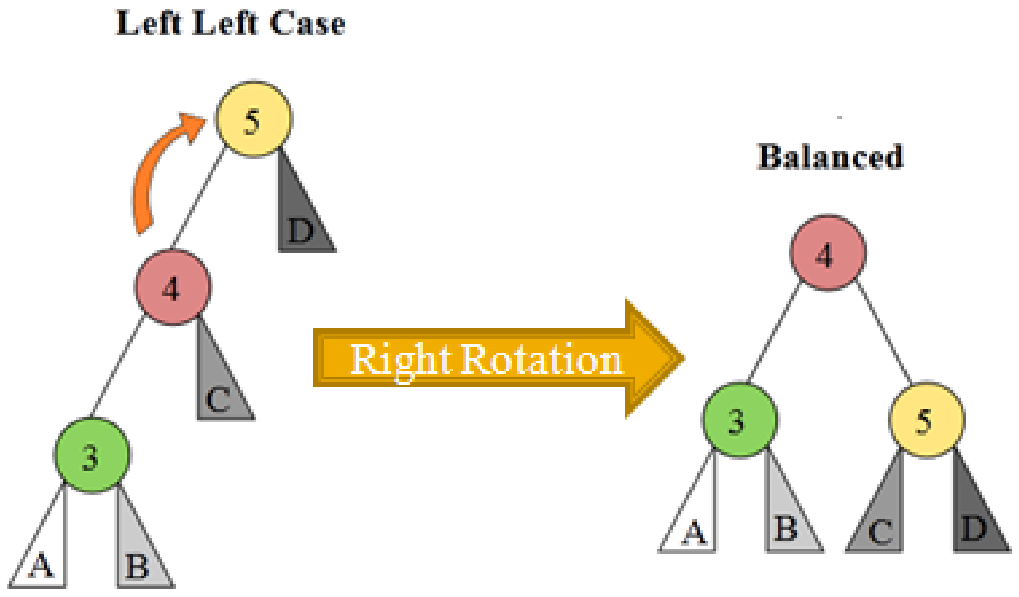


A = 0
B = 2
C = NULL
D = 9



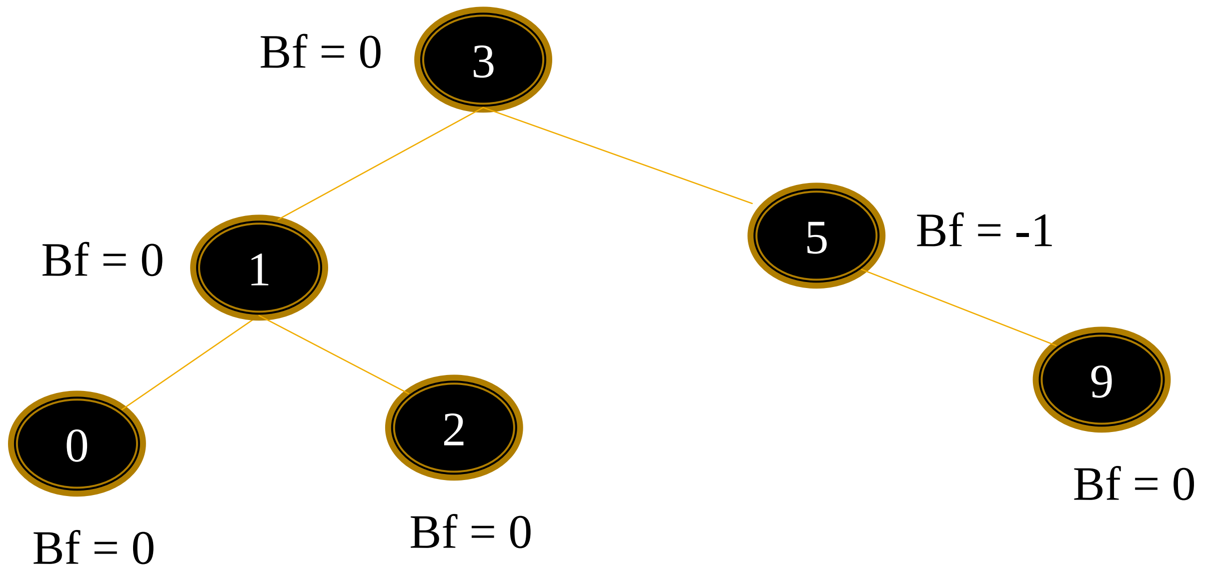
3 5 9 1 0 2 [6 10 7 4 8]



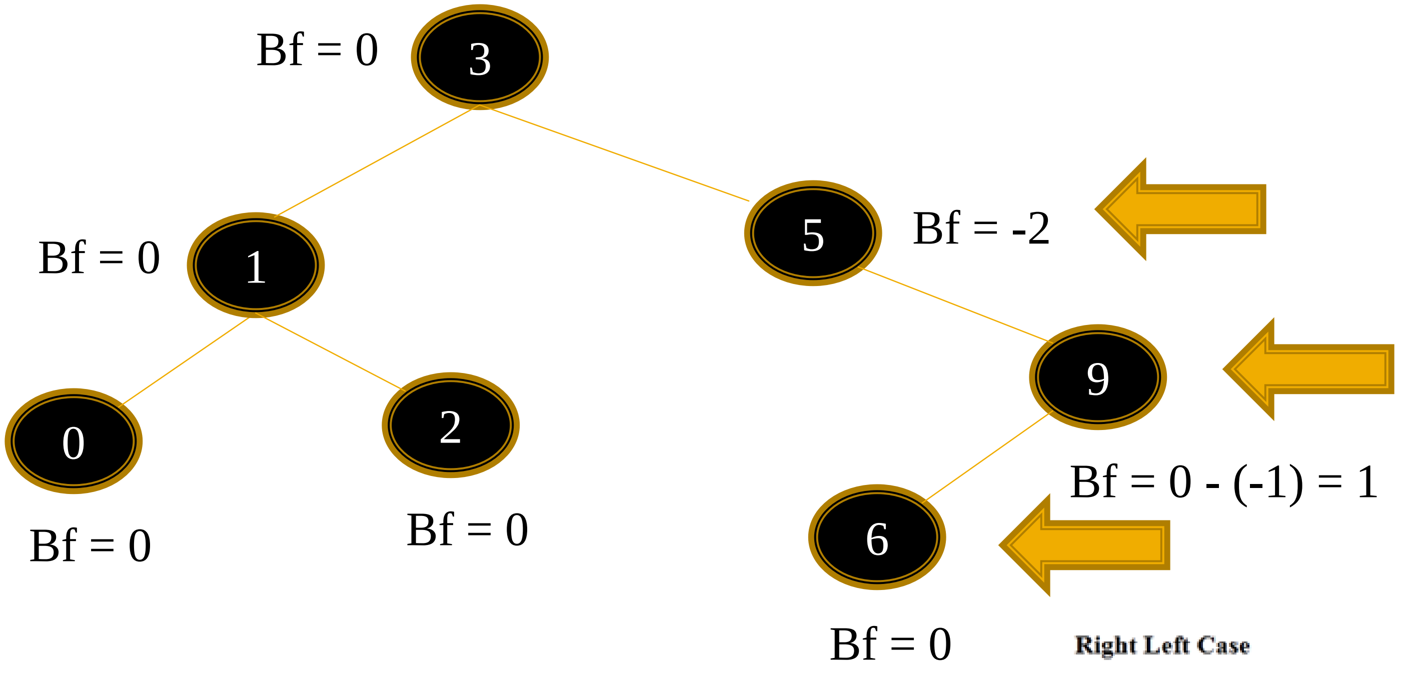


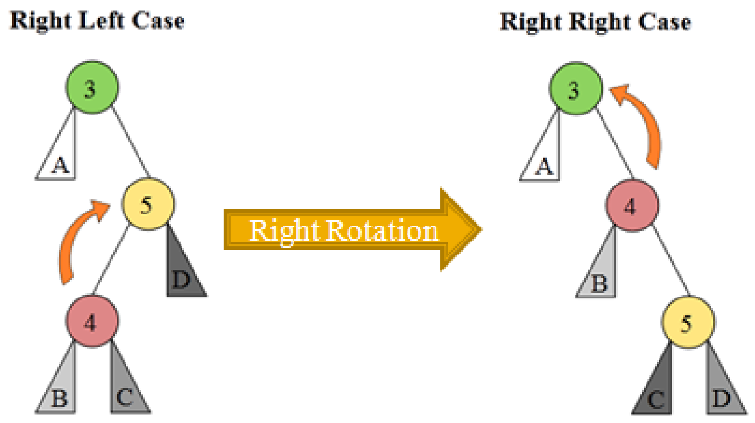
A=0
B = 2
C = NA
D = 9

3 5 9 1 0 2 [6 10 7 4 8]

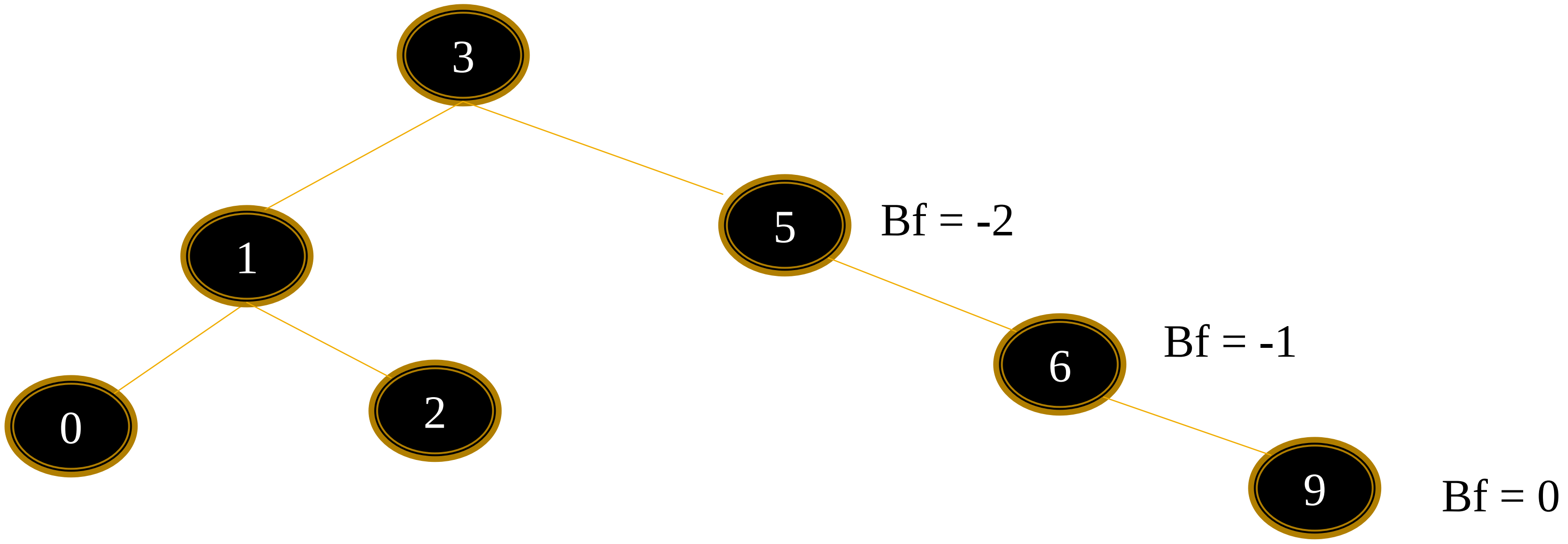


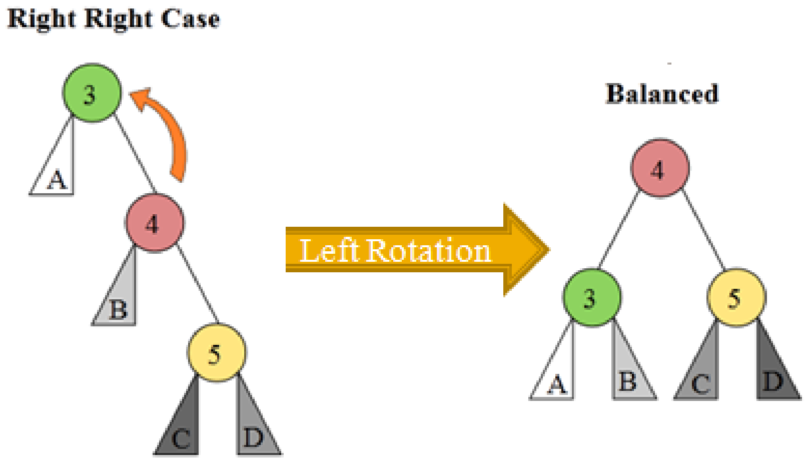
3 5 9 1 0 2 6 [10 7 4 8]



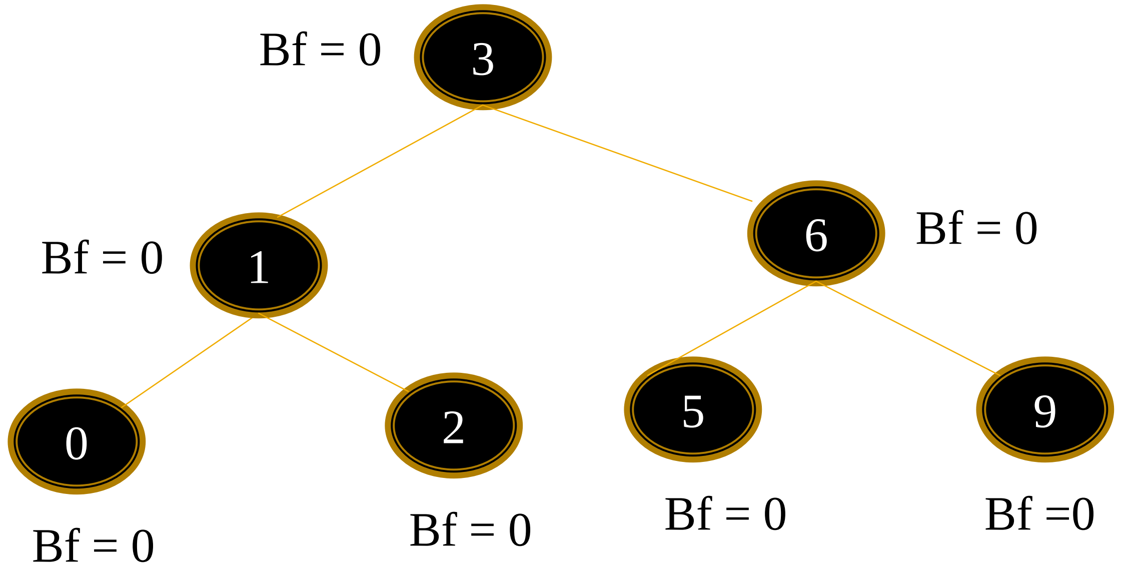


3 5 9 1 0 2 6 [10 7 4 8]

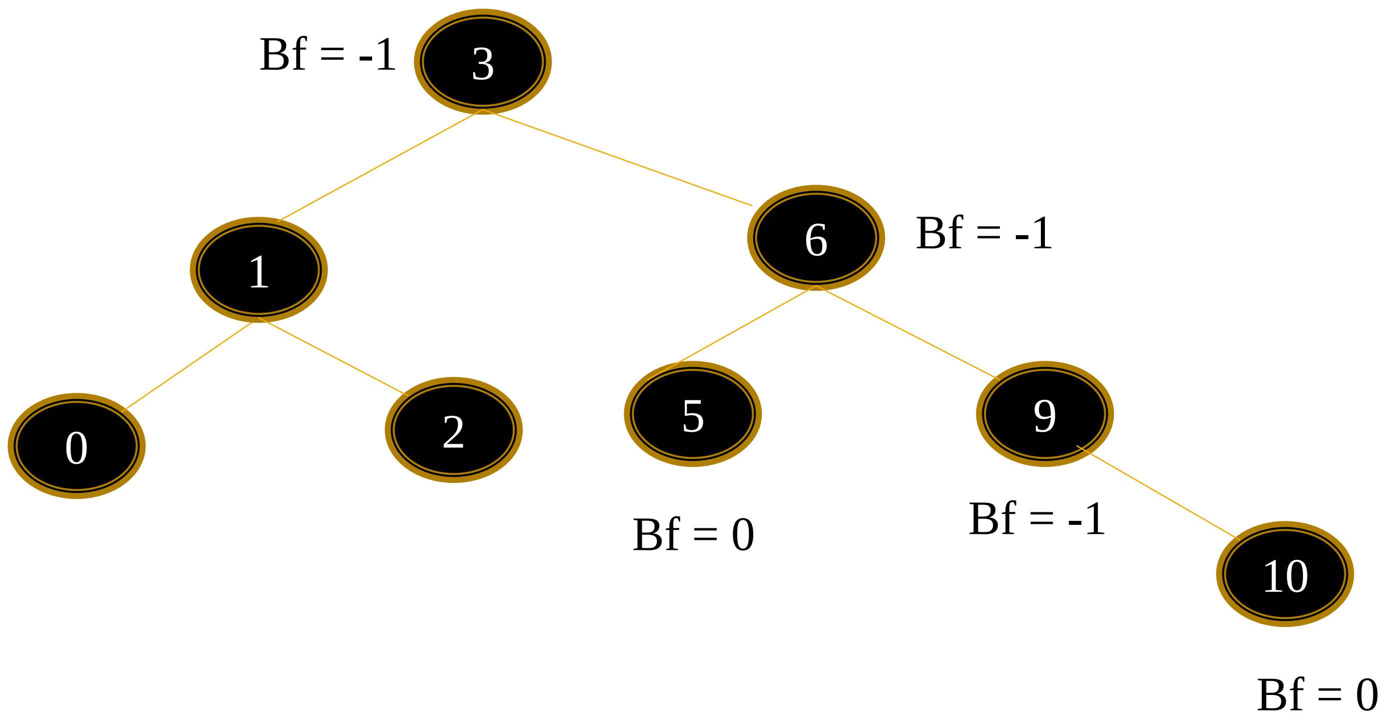




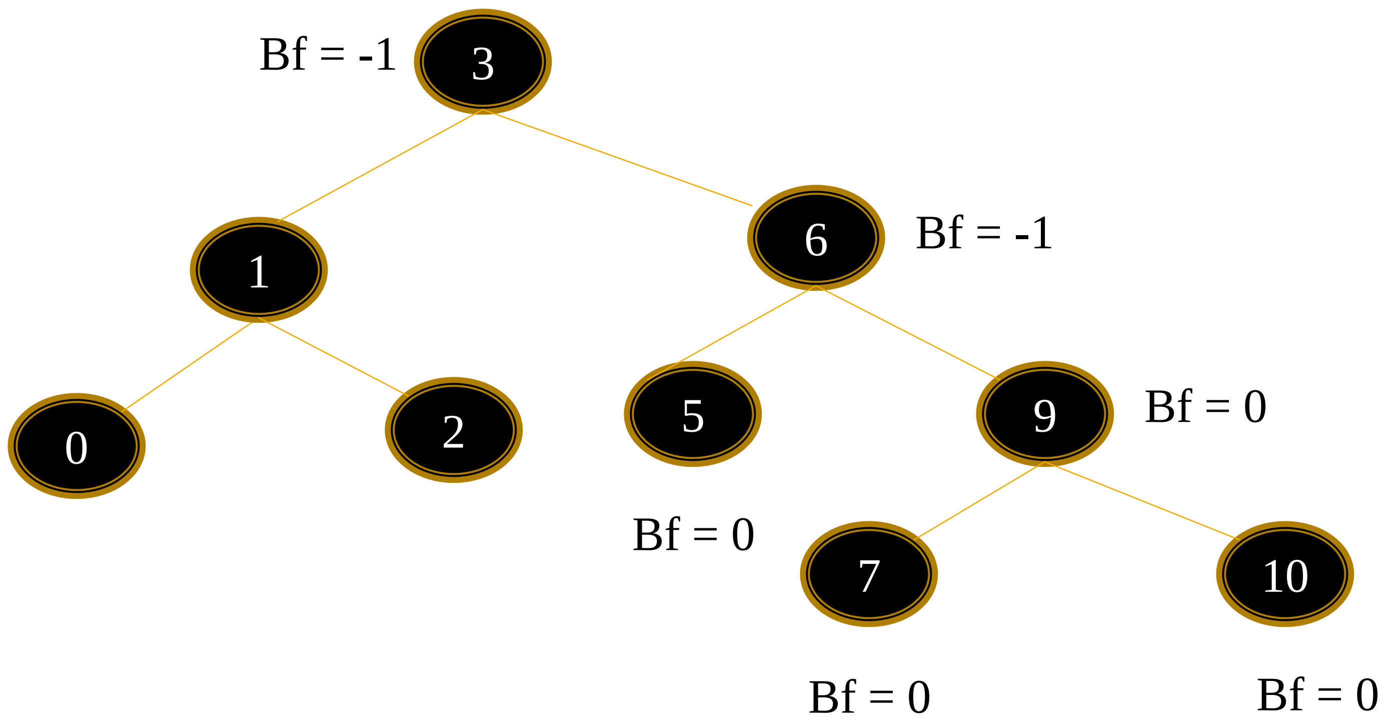
3 5 9 1 0 2 6 [10 7 4 8]



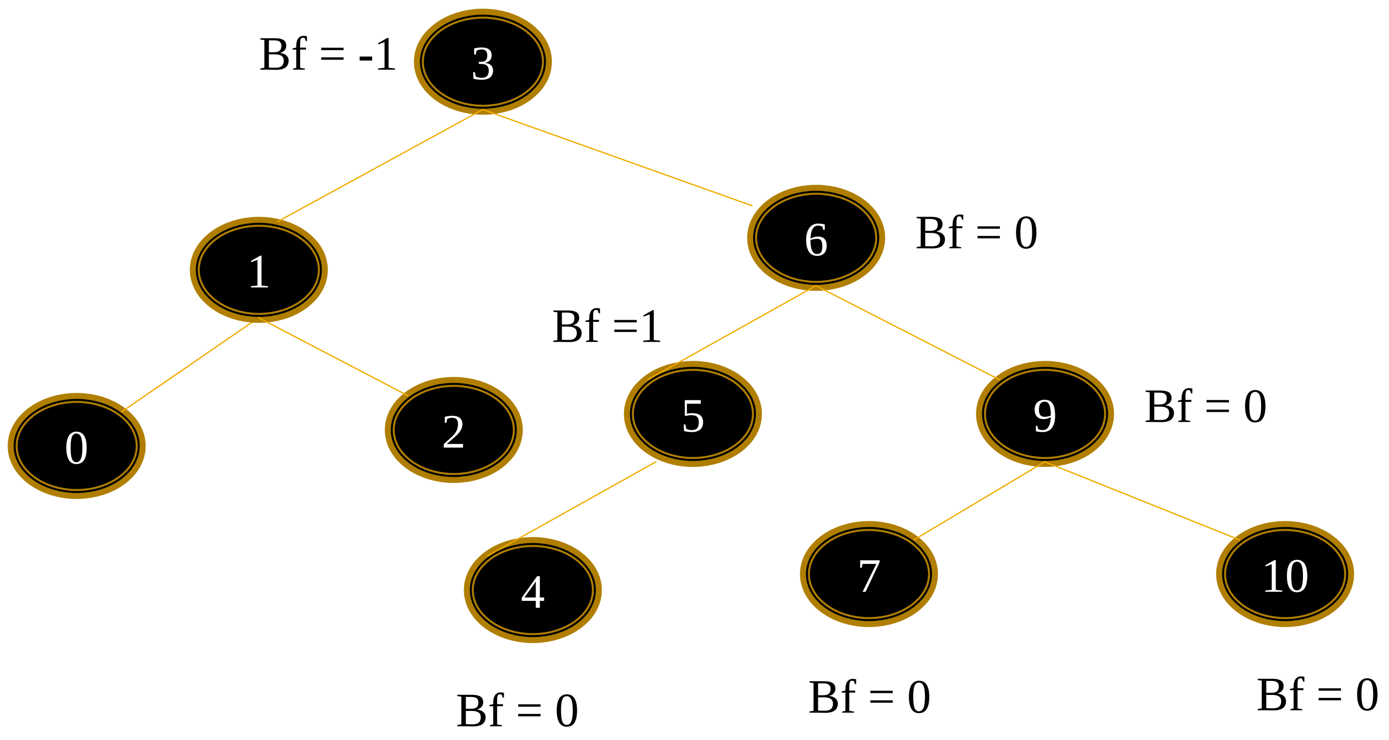
3 5 9 1 0 2 6 10 [7 4 8]



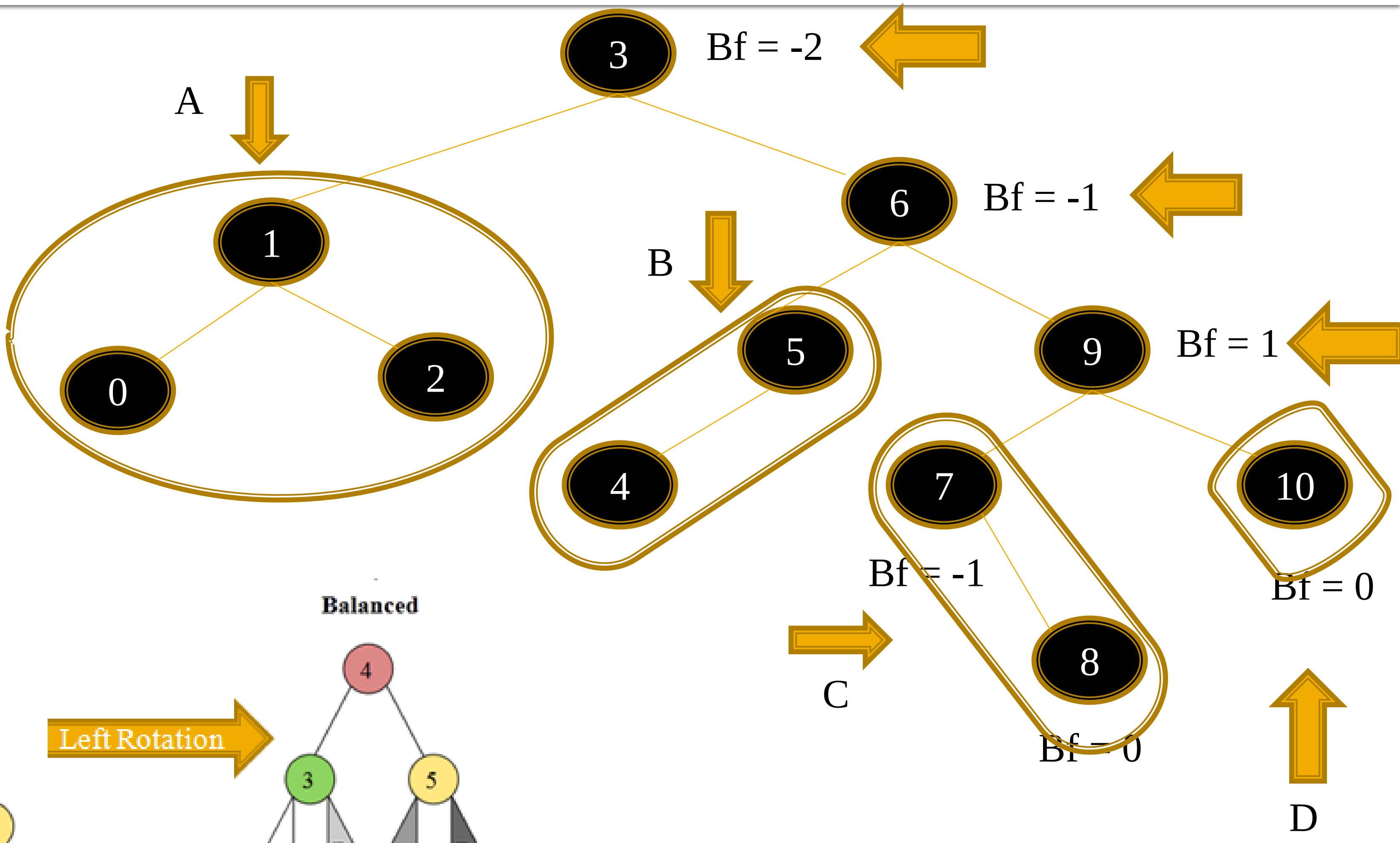
3 5 9 1 0 2 6 10 7 [4 8]

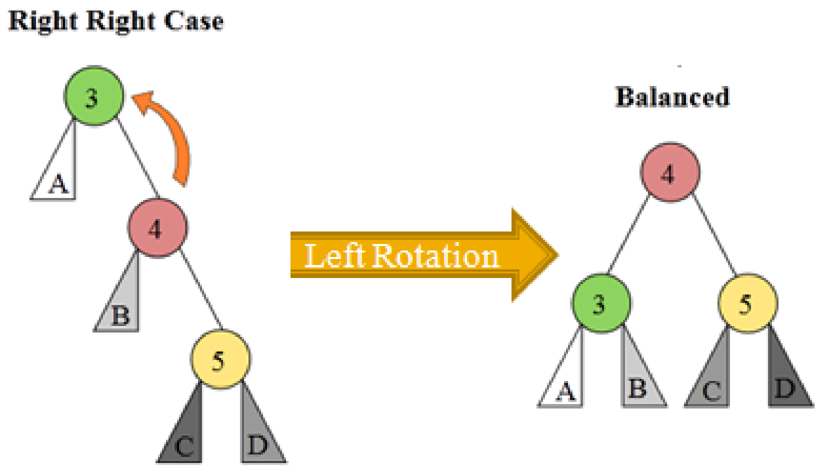


3 5 9 1 0 2 6 10 7 4 [8]

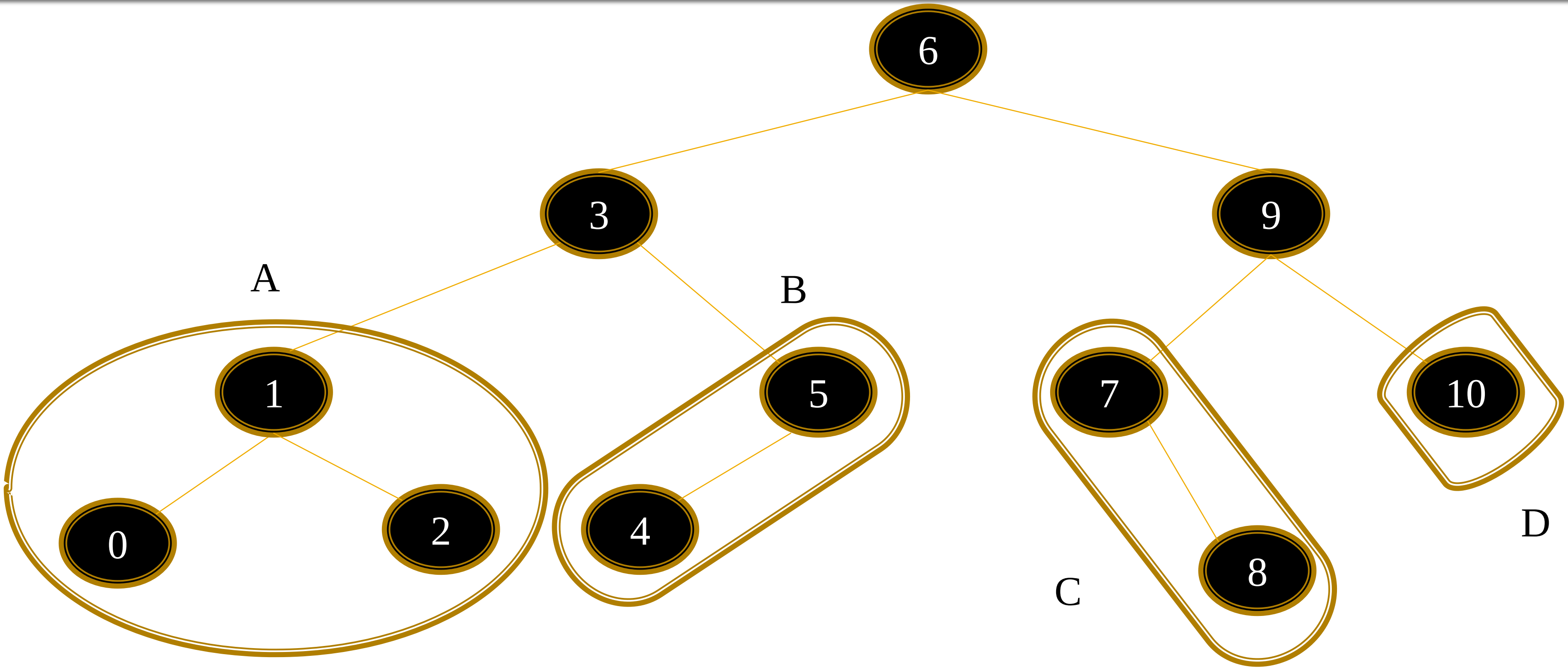


3 5 9 1 0 2 6 10 7 4 8





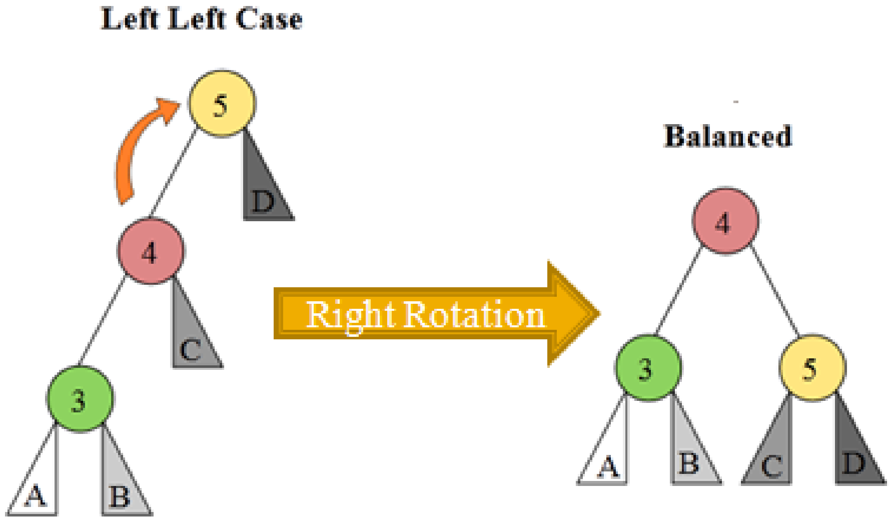
3 5 9 1 0 2 6 10 7 4 8



Implementation

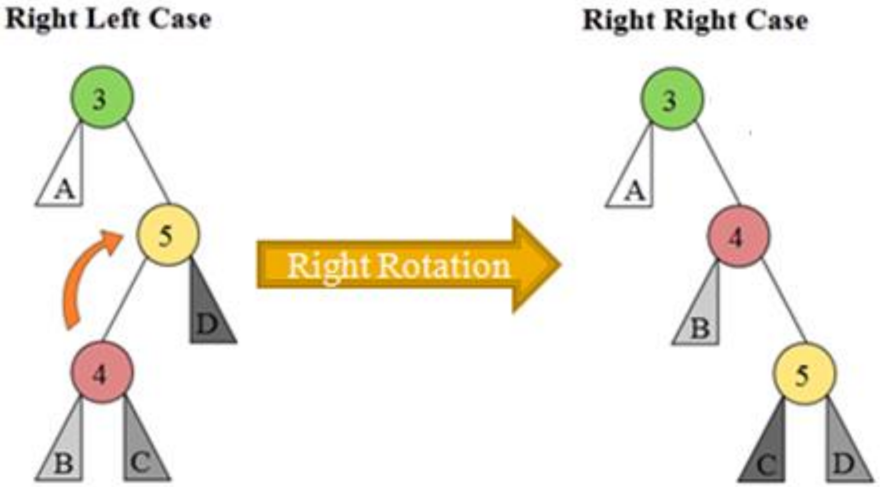
- The AVL is very easy in implementation
- Same insertion as in BST
- Insertion recursion backtrack = go up in flow
- We could update height after insertion
- Then calc the BF, if |BF| > 1 => balance
- Just one last notice on rotation

Rotation Implementation



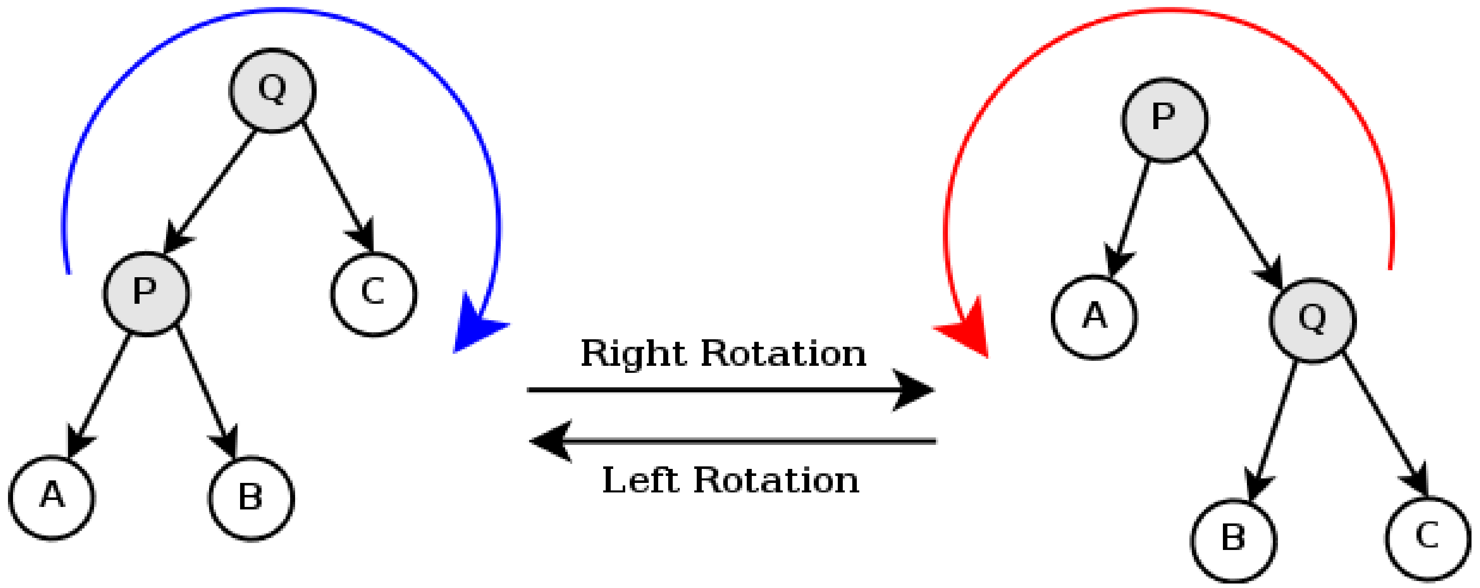
++ at root 5
Rotate root = **5**

Same Rotation behavior
One code



-+ at root 3
Rotate Right Branch = **5**

Rotation General Rule


Q
P
C
A
B
Right Rotation
Left Rotation
P
A
Q
B
C