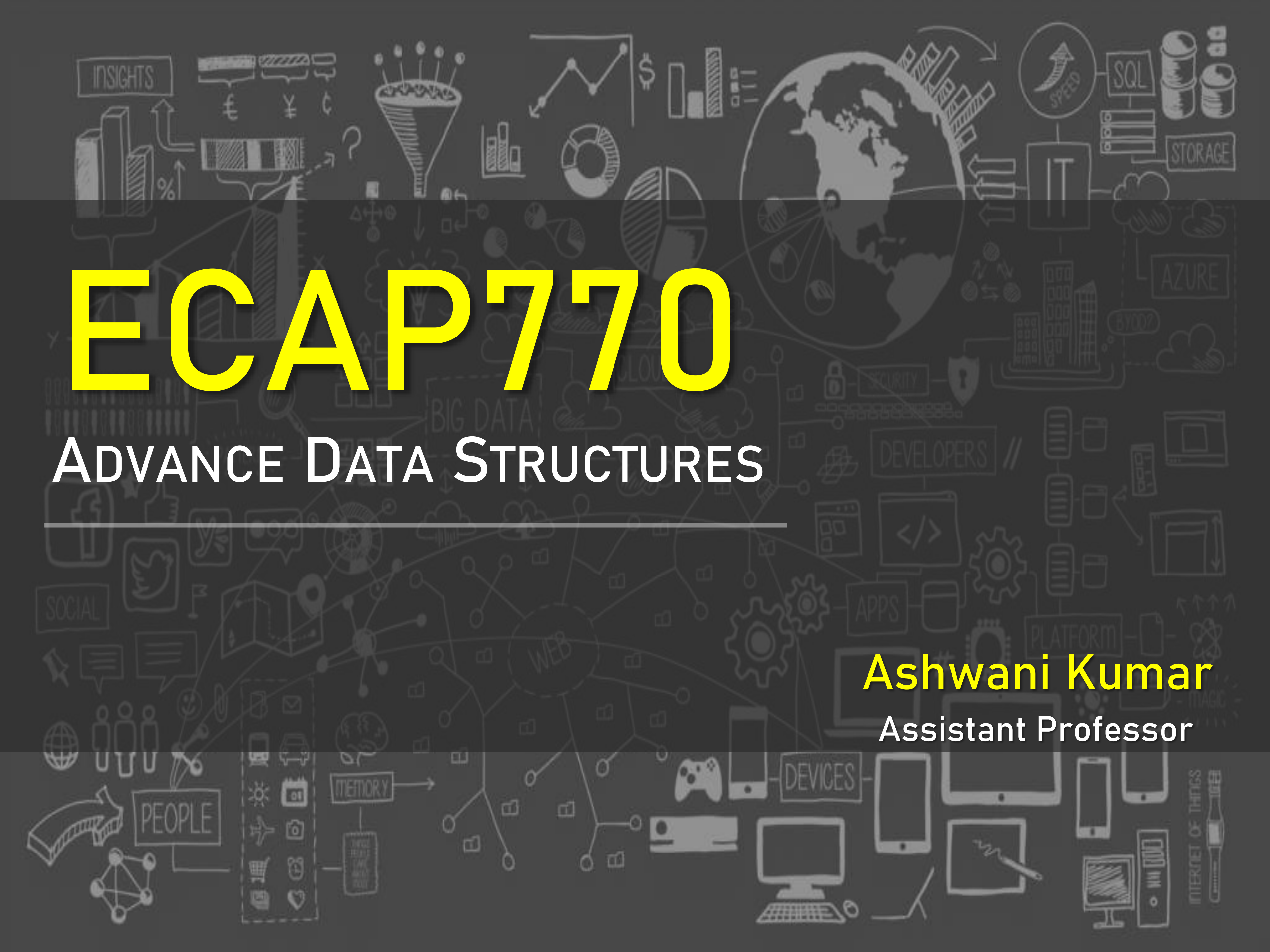


ECAP770

Advance Data Structures

Ashwani Kumar

Assistant Professor

Learning Outcomes

After this lecture, you will be able to

- Understand AVL tree balance operation

AVL tree

- AVL tree is a self balancing Binary Search Tree.
- AVL Tree is defined as height balanced binary search tree.
- In AVL tree each node is associated with a balance factor.

Balance Factor

- Balance factor of a node in an AVL tree is the difference between the height of the left sub tree and that of the right sub tree of that node.

- Balance Factor = (Height of Left Sub tree - Height of Right Sub tree) or (Height of Right Sub tree - Height of Left Sub tree)

- Balance factor value are: -1, 0 or 1.

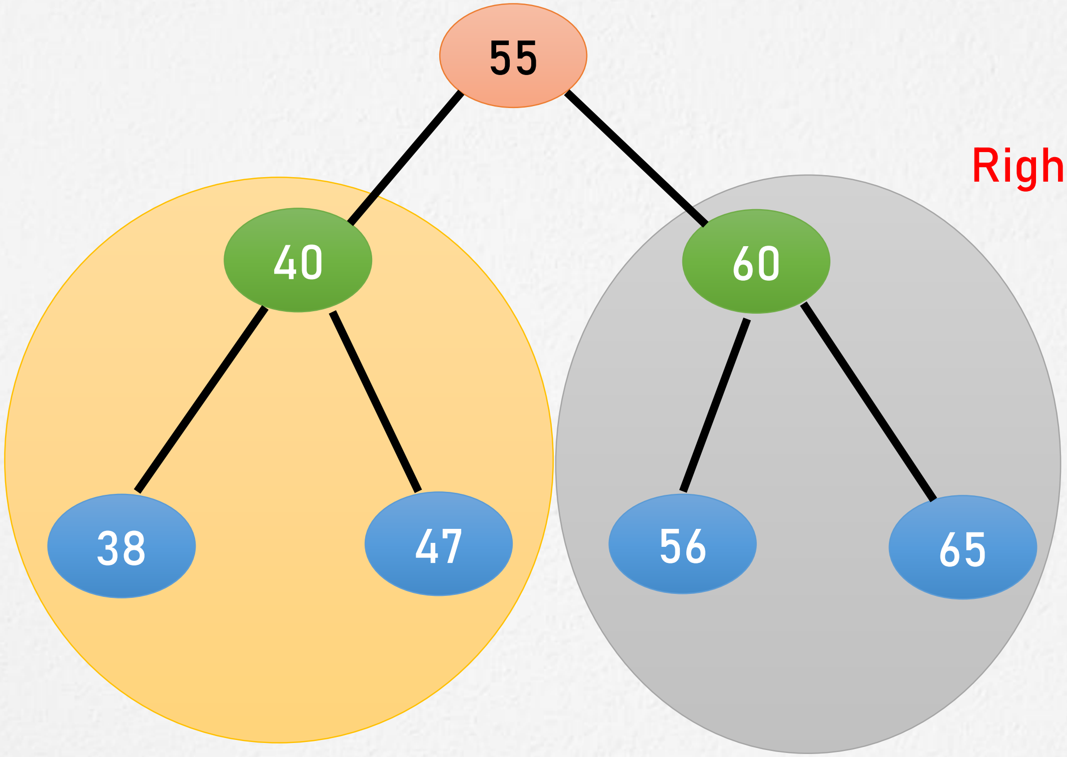
Balance Factor

- If balance factor of any node is 1, it means that the left sub-tree is one level higher than the right sub-tree.
- If balance factor of any node is 0, it means that the left sub-tree and right sub-tree contain equal height.
- If balance factor of any node is -1, it means that the left sub-tree is one level lower than the right sub-tree.

AVL tree

Left Sub Tree = 2

Right Sub Tree = 2



Balancing factor = 0

2-2 = 0 (Balanced Tree)

0 = (0, 1, -1)

AVL Tree Rotations for balancing

- Rotations are performed in AVL tree only in case if Balance Factor is other than -1, 0, and 1.
- Left rotation
- Right rotation
- Left-Right rotation
- Right-Left rotation

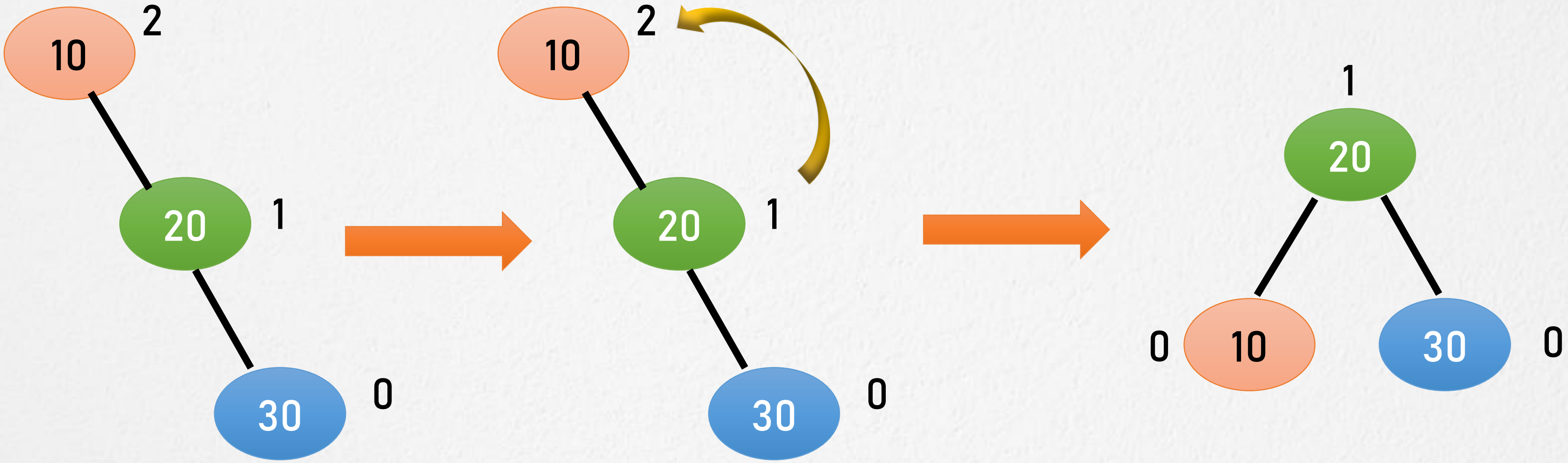
AVL Tree Rotations

- The Left rotation and Right rotation are single rotations.
- Left-Right rotation and Right-Left rotation are double rotations.

Left rotation

Data = 10, 20, 30

-1, 0, 1



Right unbalanced tree

Left rotation

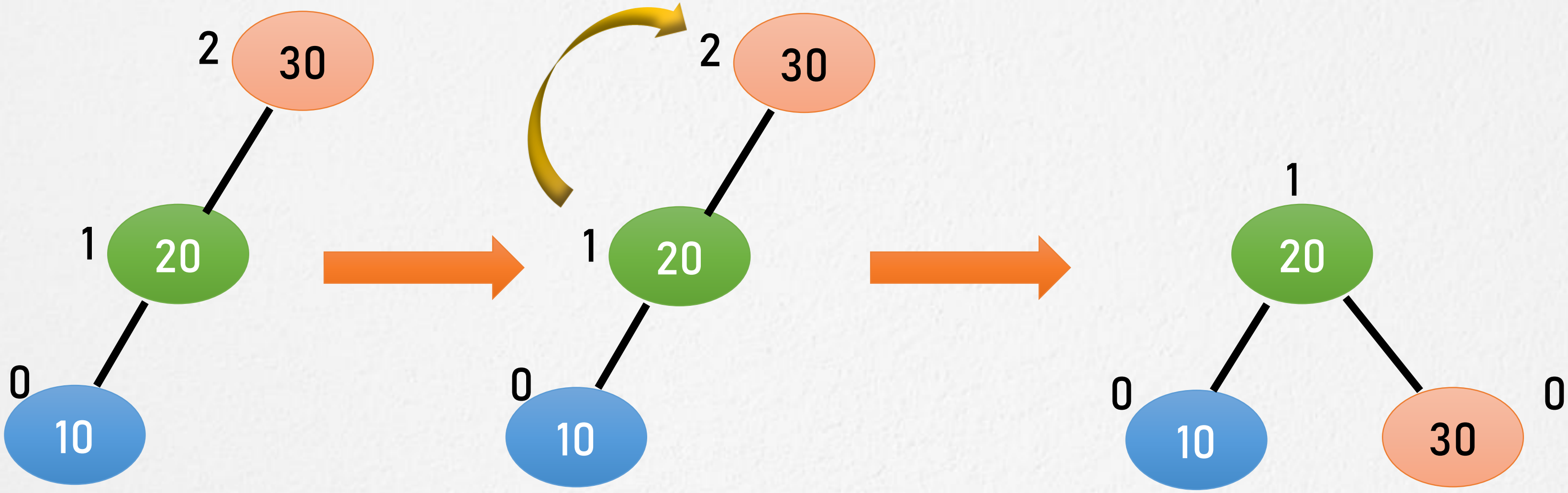
Balanced tree

0 - 2 = -2

Right rotation

Data = 30, 20, 10

-1, 0, 1



Left unbalanced tree

2 - 0 = 2

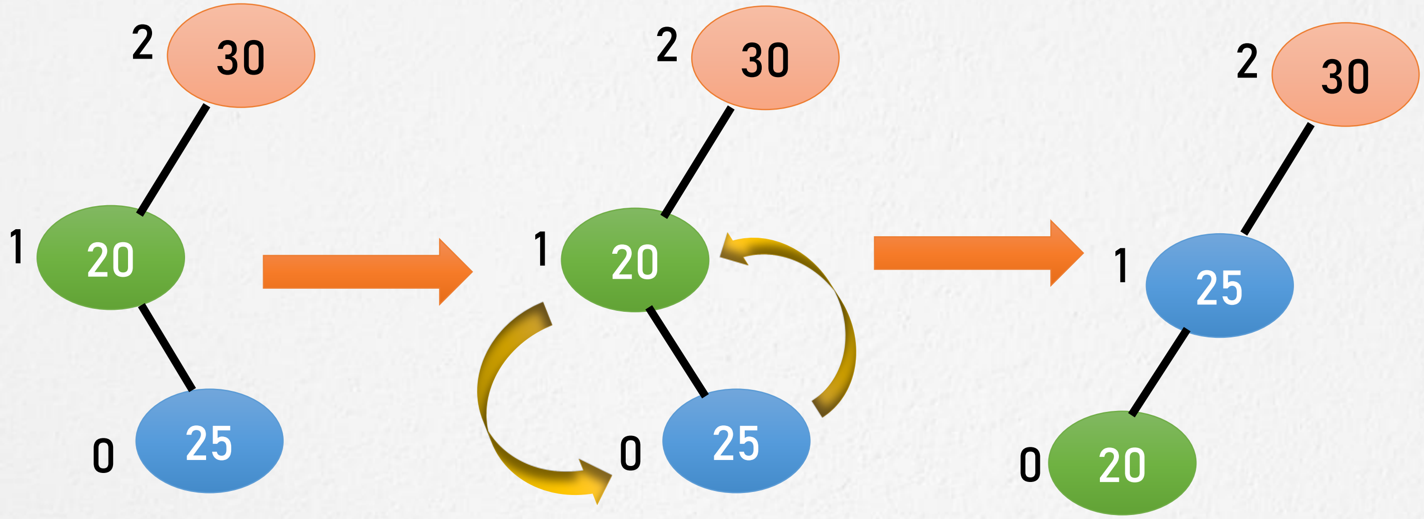
Right rotation

Balanced tree

Left-Right rotation

Data = 30, 20, 25

-1, 0, 1



Unbalanced tree

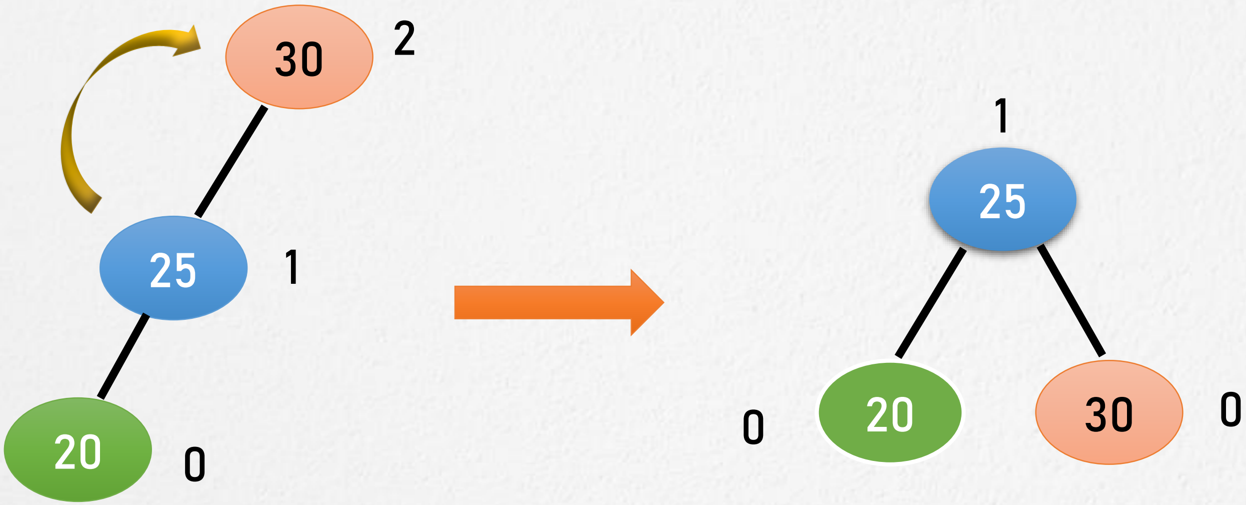
2 - 0 = 2

Left rotation

Left-Right rotation

Data = 30, 20, 25

-1, 0, 1



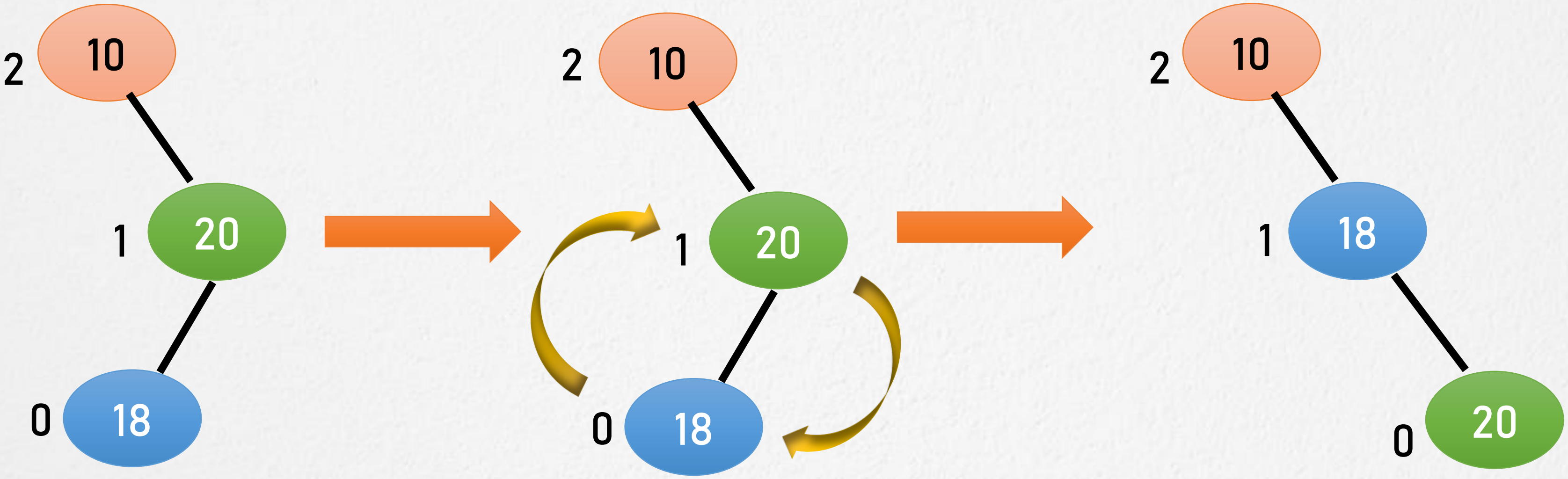
Right rotation

Balanced tree

Right-Left rotation

Data = 10, 20, 18

-1, 0, 1



Unbalanced tree

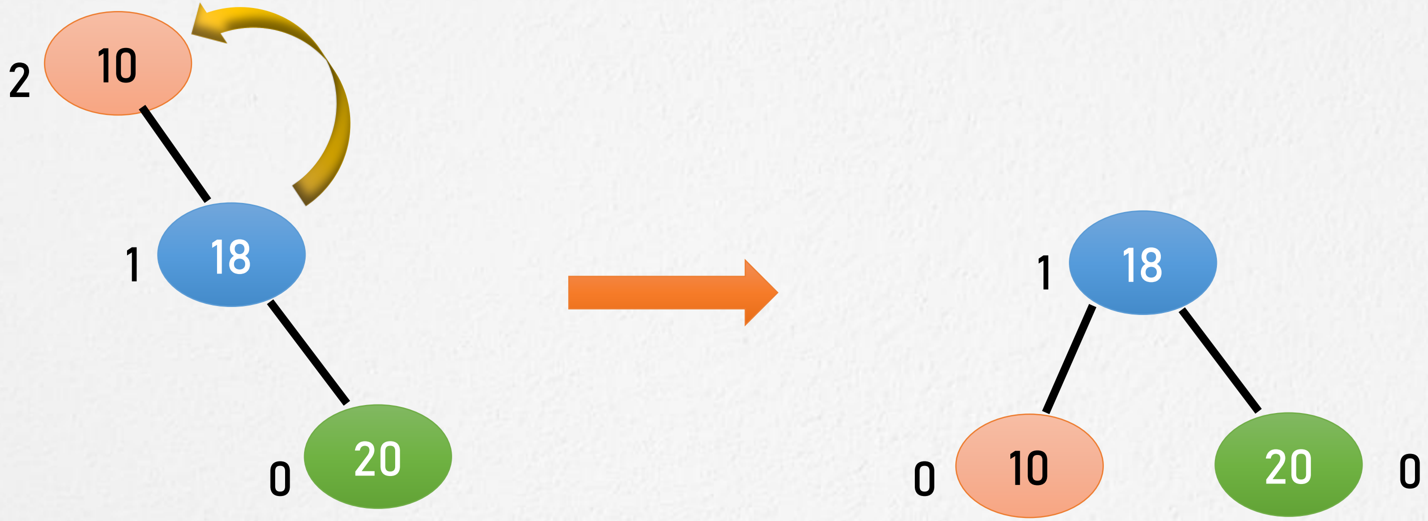
0 - 2 = -2

Right rotation

Right-Left rotation

Data = 10, 20, 18

-1, 0, 1



Left rotation

Balanced tree

Advantages of AVL tree

- The height of the AVL tree is always balanced. The height never grows beyond log N, where N is the total number of nodes in the tree.
- Search time complexity is better as compared to Binary Search trees.
- AVL trees have self-balancing capabilities.

That’s all for now…