Data Structures and Analysis of Algorithms
CST 225-3

AVL Trees

Uva Wellassa University

Drawback of BST

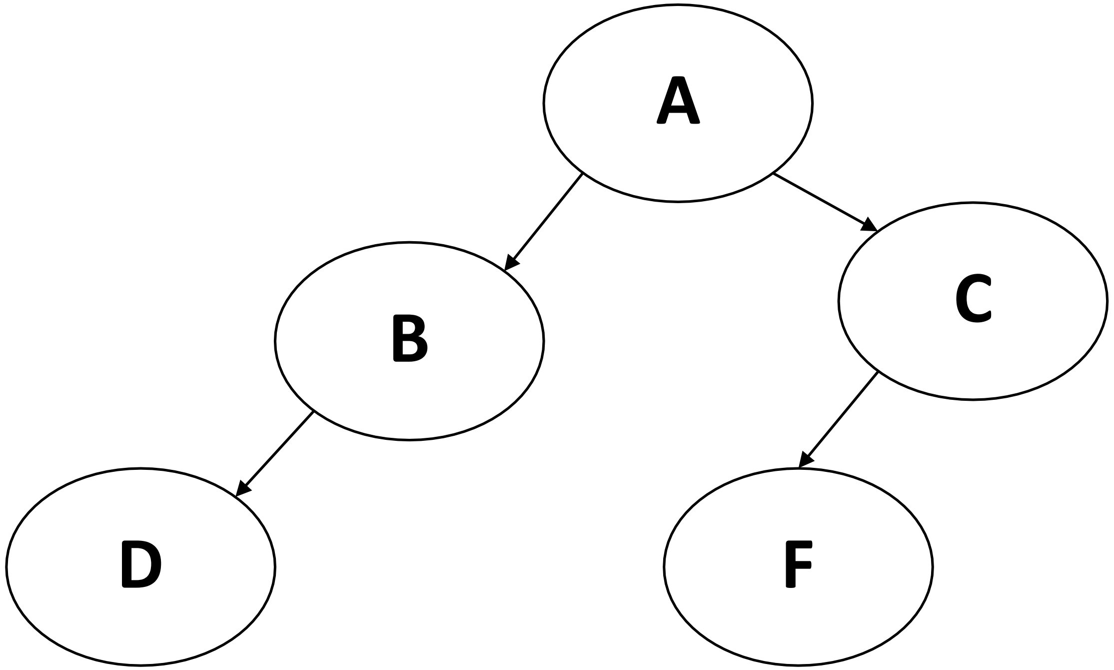
- Height is not under control.
- Height depends on the insertion of the element.
- Tree traversal takes more time.

What are AVL Trees?

- AVL tree is invented by Adelson, Velski & Landis. The tree is named AVL in honour of its inventors.
- AVL Tree can be defined as height balanced binary search tree in which each node is associated with a balance factor.

  Balance factor of a node = height(left-sub tree) – height(right-sub tree)
- **AVL tree property**: for every node in the tree, the height of the left and right sub trees differs by at most 1.
- Binary search tree is an AVL tree if balance factor of each node is 0, 1 or -1

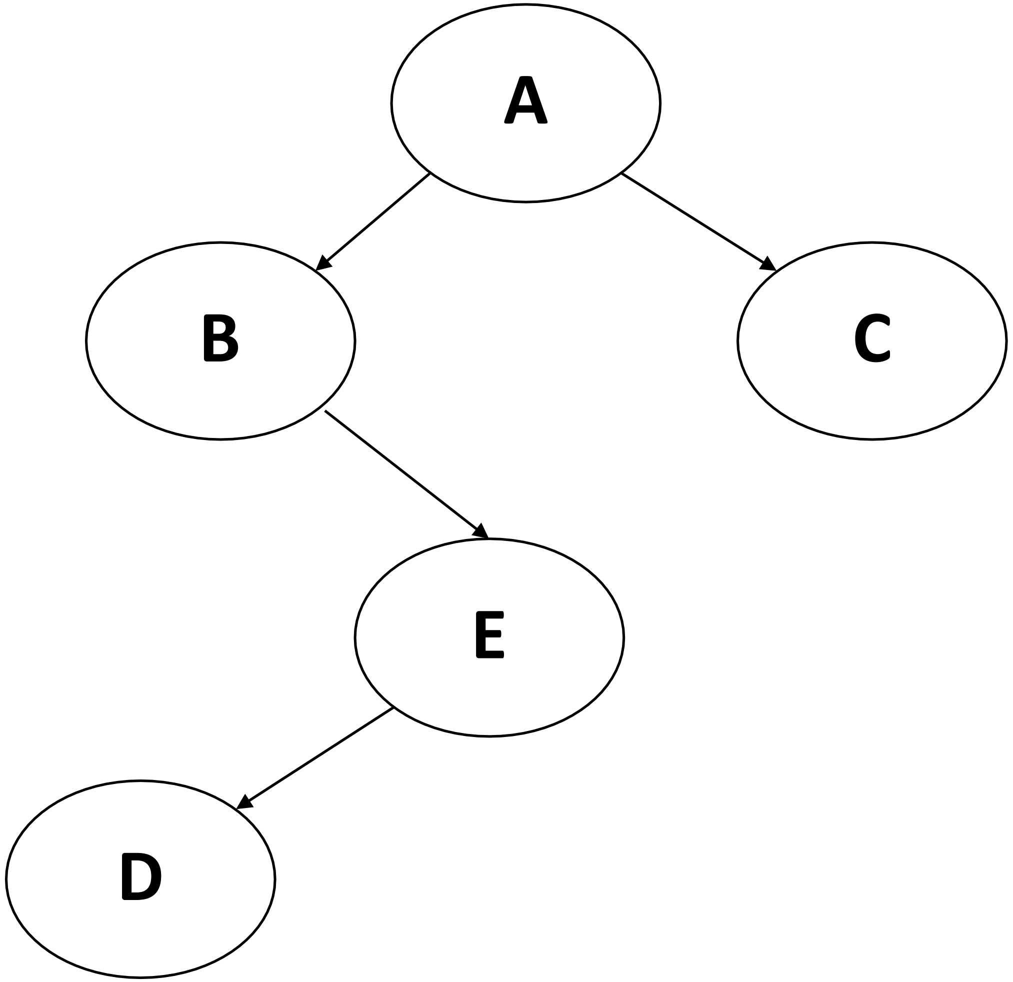
Example 1



Node	Height of Left Sub Tree	Height of Right Sub Tree	Balance Factor
A	2	2	0
B	1	0	1
C	1	0	1
D	0	0	0
F	0	0	0

This tree is balanced or AVL tree because every node‘s balance factor is in between -1,0 and 1.

Example 2



Node	Height of Left Sub tree	Height of Right Sub Tree	Balance Factor
A	3	1	2
B	0	2	-2
C	0	0	0
E	1	0	1
D	0	0	0

This tree is imbalanced balanced because every node‘s balance factor is not in between -1,0 and 1.

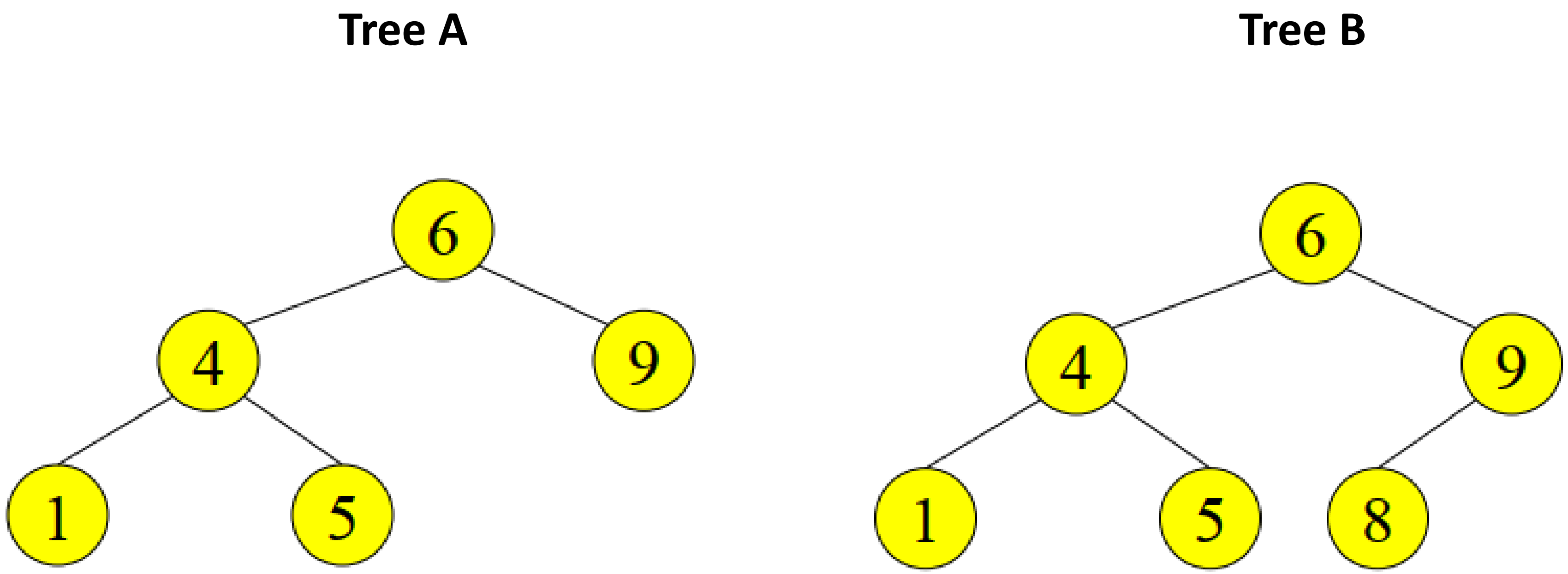
Maintaining Balance

To maintain AVL balance, observe that:

- Inserting a node can increase the height of a tree by at most 1
- Removing a node can decrease the height of a tree by at most 1

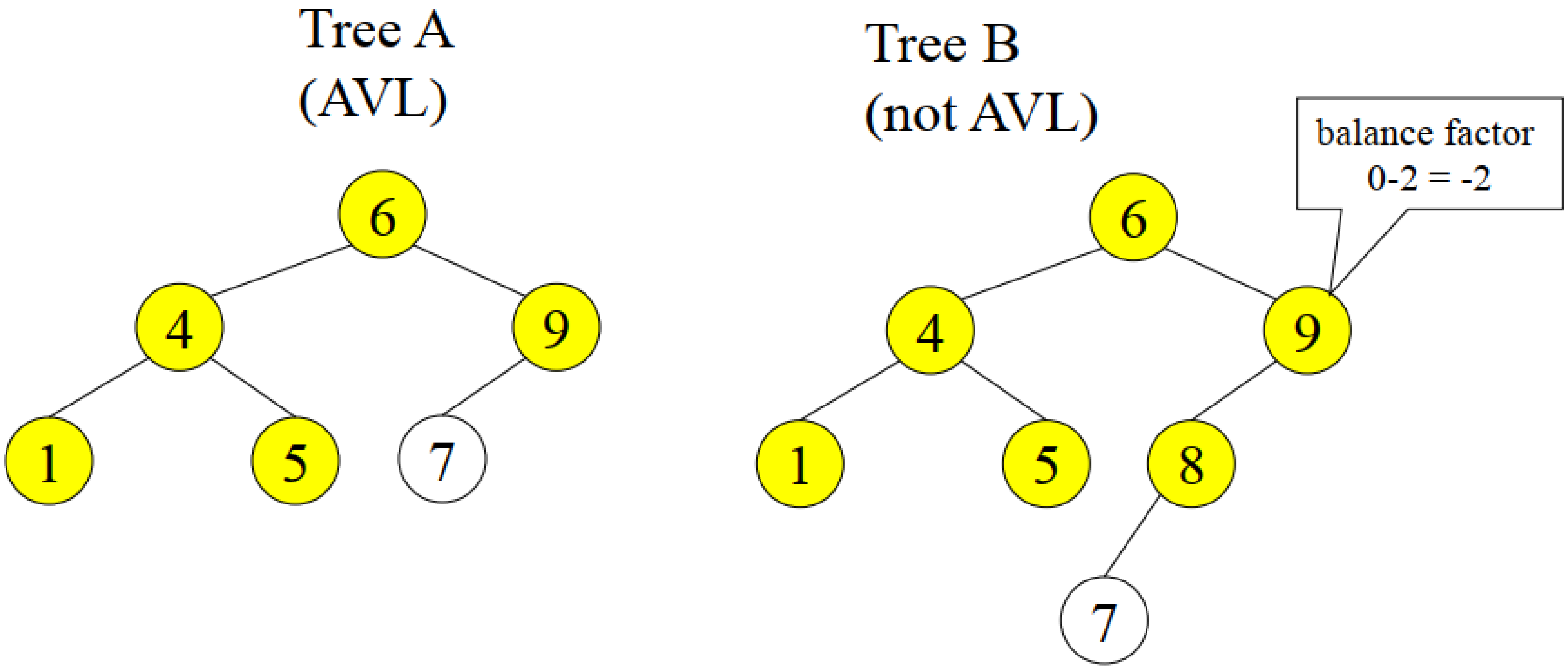
- If the tree is imbalanced based on balance factor, then rotations need to be performed in order to make it as a balanced tree.
- Rotations occur when tree becomes unbalanced from insertion or deletion.

Are these trees AVL or not?



- Insert 7 to both of above trees. Are these trees now AVL or not?

Trees after Inserting 7



Insertion and Rotation in AVL Trees

- Insert operation may cause balance factor to become 2 or –2 for some nodes.
- Only nodes on the path from insertion point to root node have possibly changed in height.
- So after the insertion, go back up to the root node by node, updating heights.
- If a new balance factor is 2 or –2, adjust tree by rotation around the node.

Rotation

- Rotation can be done with tree(3) nodes in the tree.
- Rotation is the process of moving the nodes to either left or right to make tree balanced.
- There are two types of rotations.

1. Single rotation- If the three nodes lie in a straight line, single rotation is needed to restore the balance.
2. Double rotation- If these three nodes lie in a "dog-leg" pattern, you need double rotation to restore the balance.

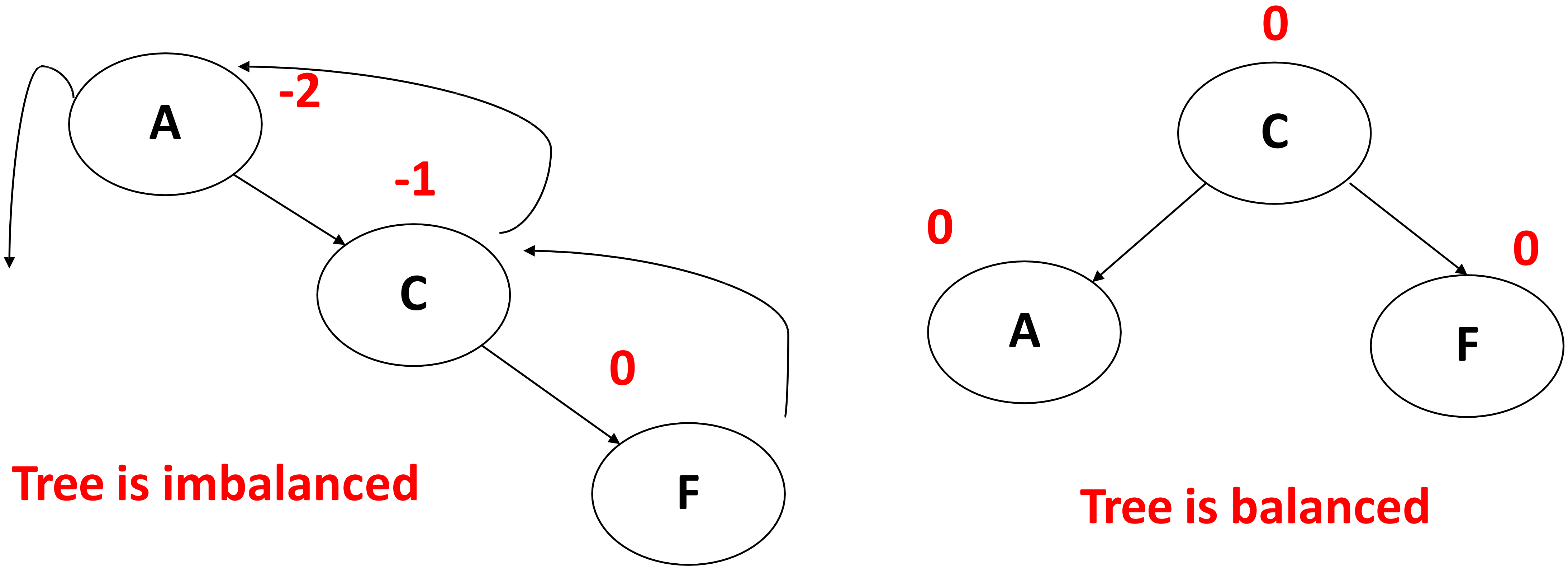
Rotation Classifications

1. Single Rotation
   - Left Left Rotation (LL Rotation)
   - Right Right Rotation (RR Rotation)

2. Double Rotation
   - Left Right Rotation (LR Rotation)
   - Right Left Rotation (RL Rotation)

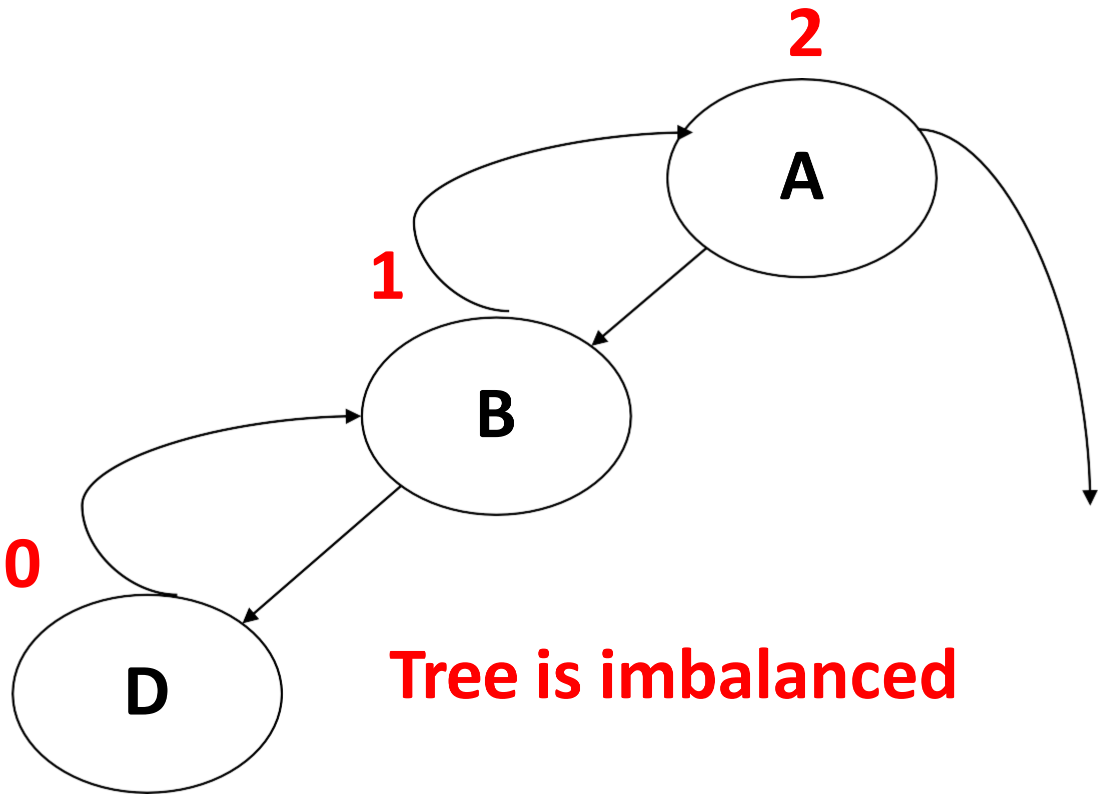
Single LL Rotation

- Every node moves one position to left from the current position.

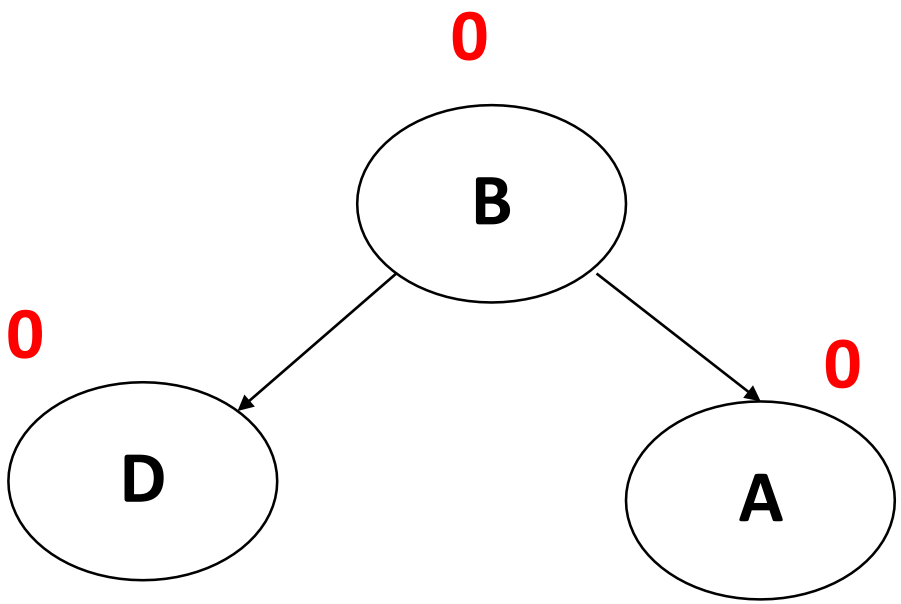


Single RR Rotation

- In RR rotation every node moves one position to right from the current position.



Tree is imbalanced

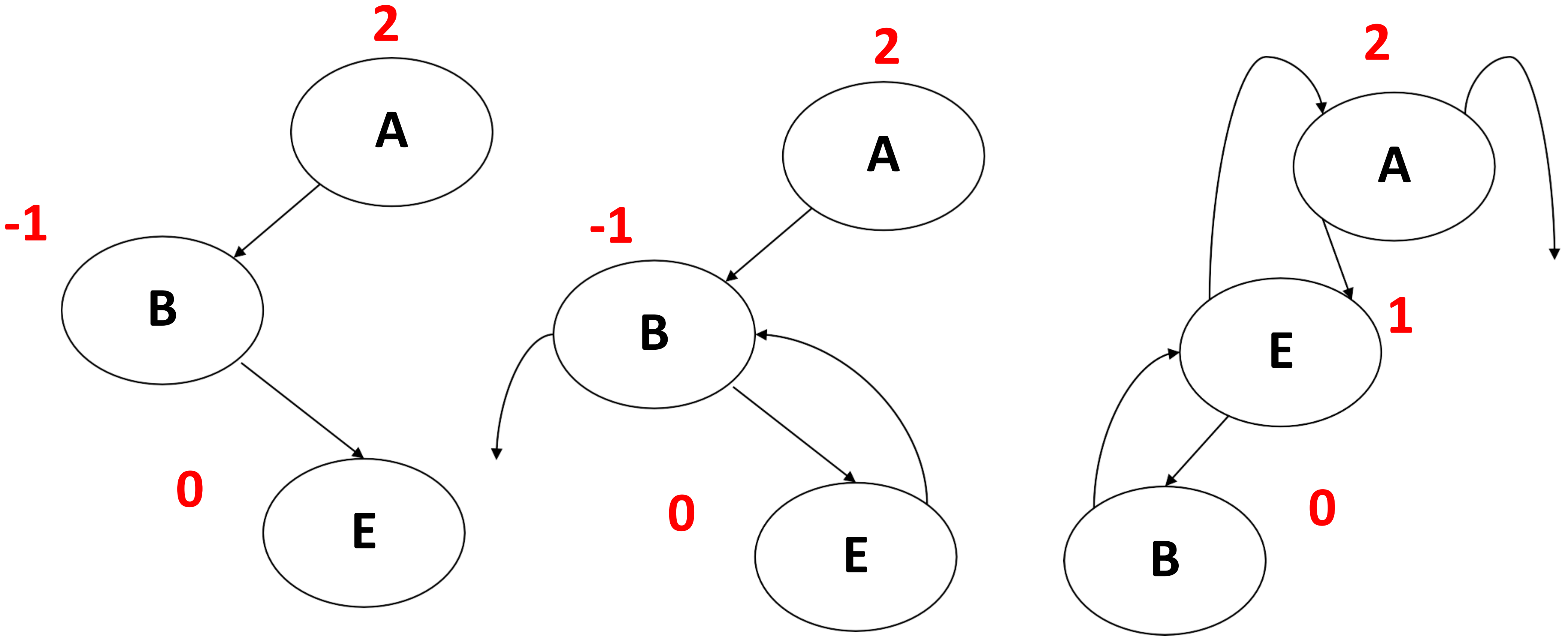


Tree is balanced

Double LR Rotation

- The LR Rotation is combination of single left rotation followed by single right rotation.
- In LR Rotation, first every node moves one position to left then one position to right from the current position.

Double LR Rotation - Example

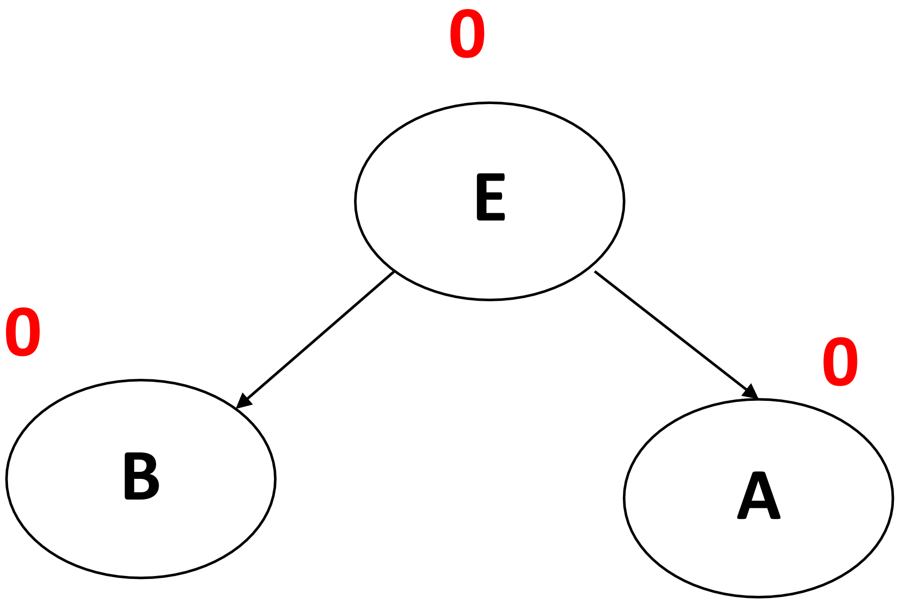


Tree is imbalanced

LL Rotation

RR Rotation

Double LR Rotation - Example

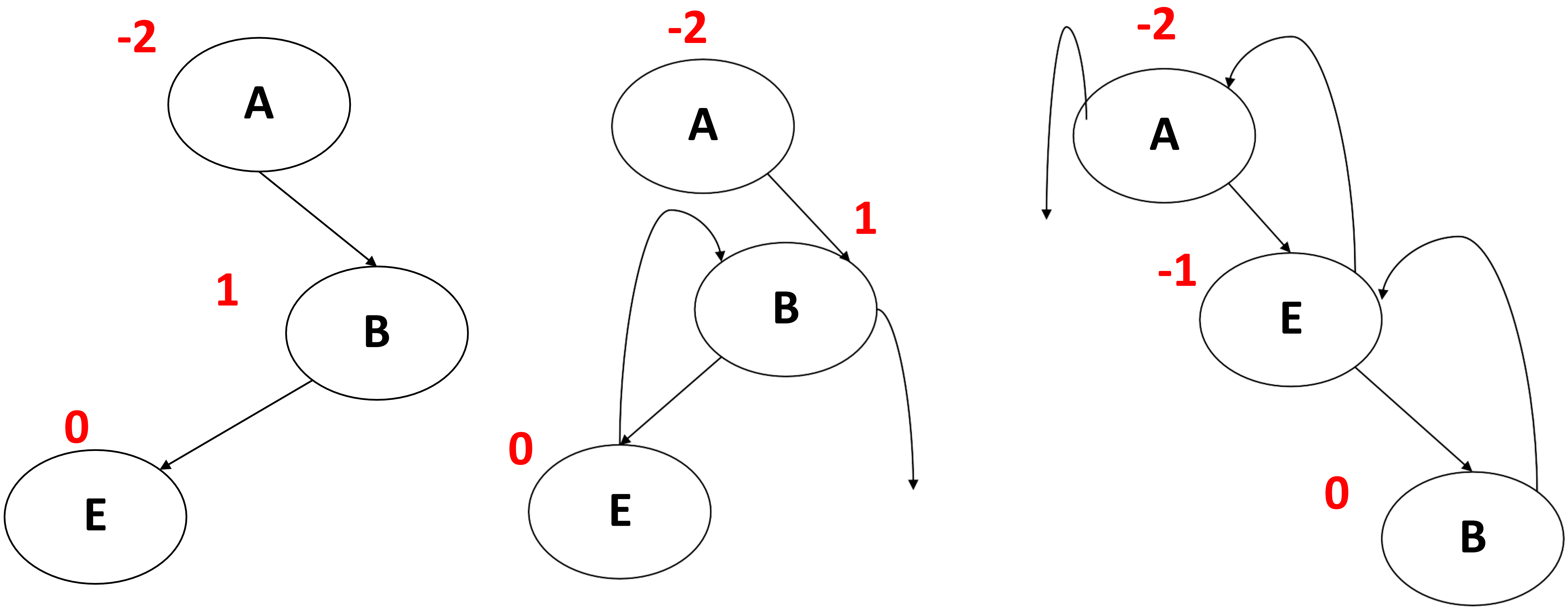


Tree is balanced now

Double RL Rotation

- The RL Rotation is combination of single right rotation followed by single left rotation.
- In RL Rotation, first every node moves one position to right then one position to left from the current position.

Double RL Rotation - Example

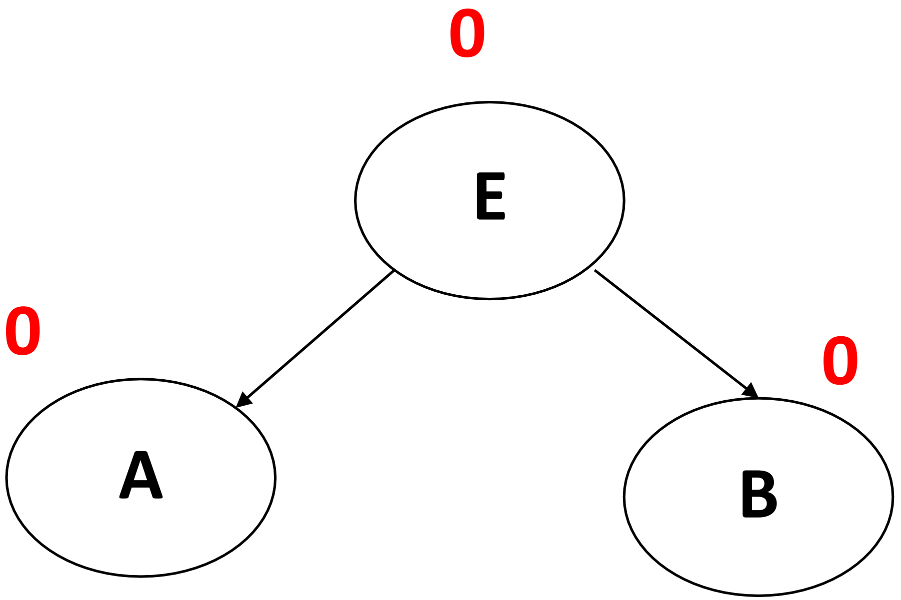


Tree is imbalanced

RR Rotation

LL Rotation

Double RL Rotation - Example



Tree is balanced now

Construction of AVL Trees

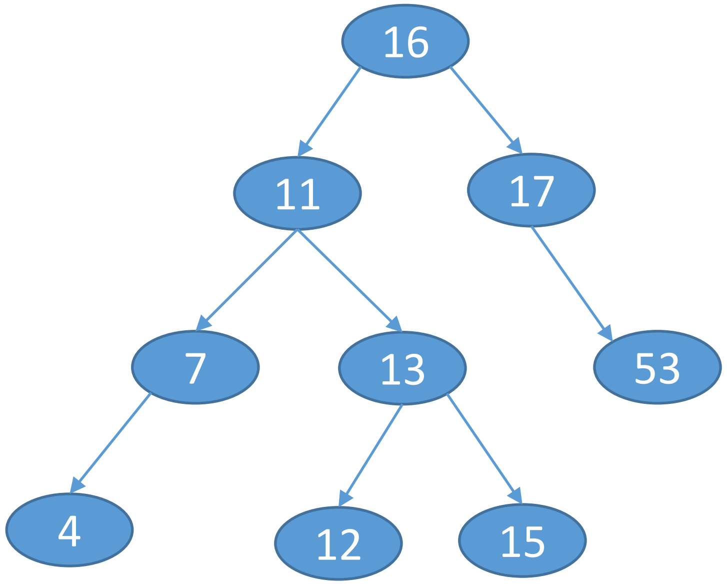
- Insert the following elements and construct a AVL tree.

  16, 17, 11, 7, 53, 4, 13, 12,15

Construction of AVL Trees

- Insert the following elements and construct a AVL tree.

16, 17, 11, 7, 53, 4, 13, 12,15

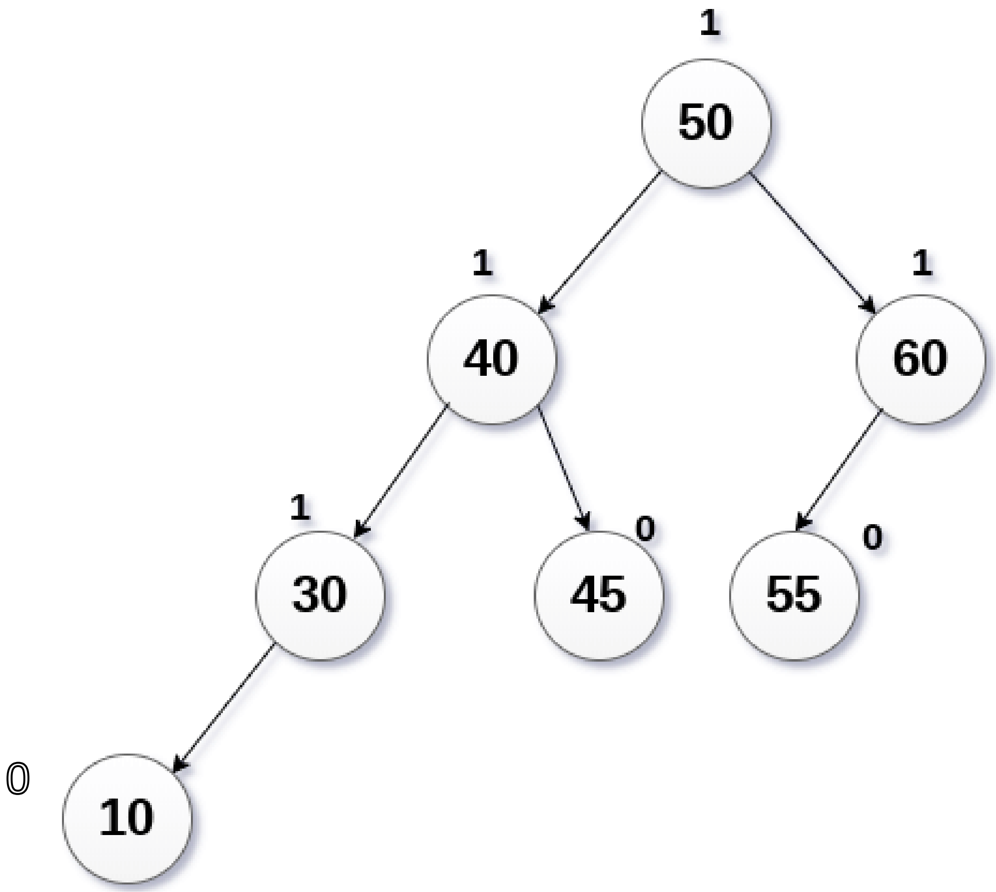


Deletion in AVL Tree

- Deletion of element is same as in BST.
- After the deletion, check the balance factor of each node of the tree.
- If not balanced, then balance the tree.

Exercise

- Delete node 55 from the following AVL tree.



Applications of AVL Trees

- Used frequently for quick searching.

Questions?