

CSE221

Lecture 12: AVL Trees

Fall 2021

Young-ri Choi

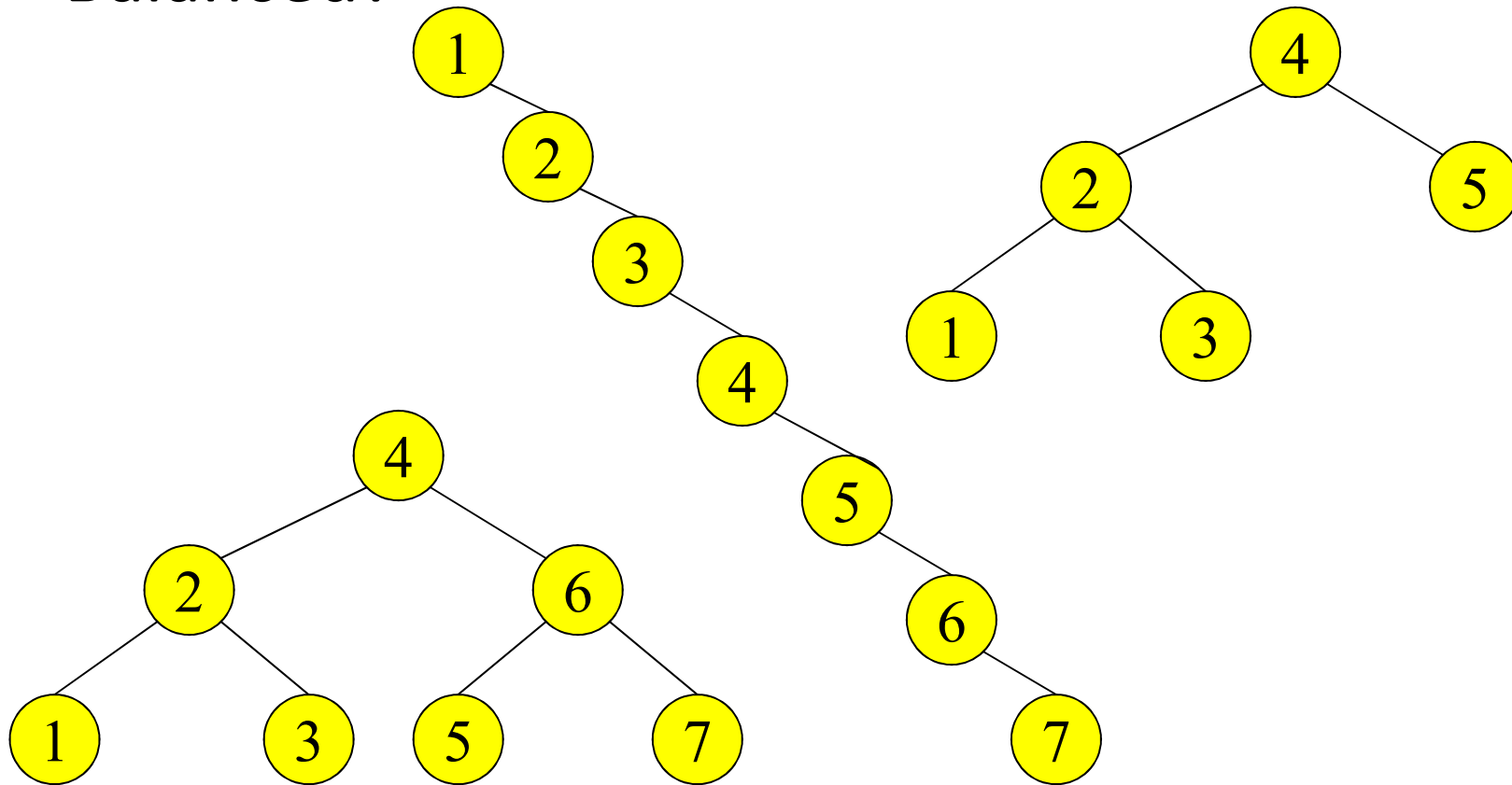
Acknowledgment: The content of this file is based on the slides of the textbook as well as the slides provided in former lectures at UNIST.

Dynamic Balanced Binary Search Trees

- Average and maximum search time in binary search trees depends on the height of the tree
- Minimum height of a binary tree with n nodes is $\log n$
- Dynamic balanced binary search tree is to keep the height of a binary search tree $O(\log n)$ for search, insert, delete

Balanced and Unbalanced BST

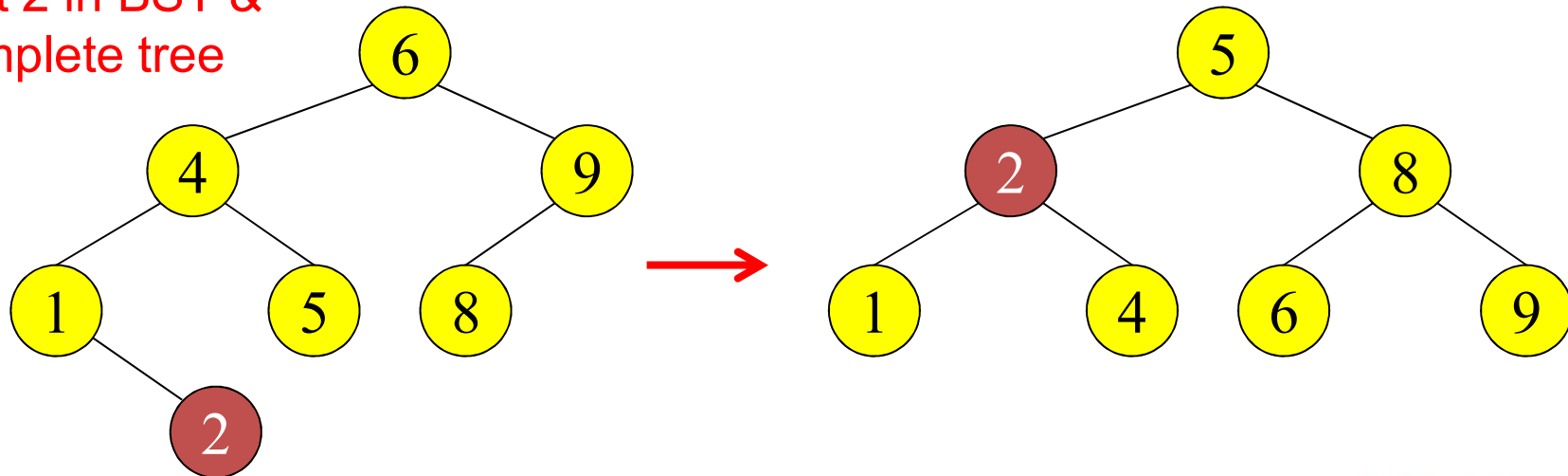
- Balanced?



Perfect Balance

- Want a **complete tree** after every operation?
 - Tree is full except possibly for the lower right
 - This is expensive

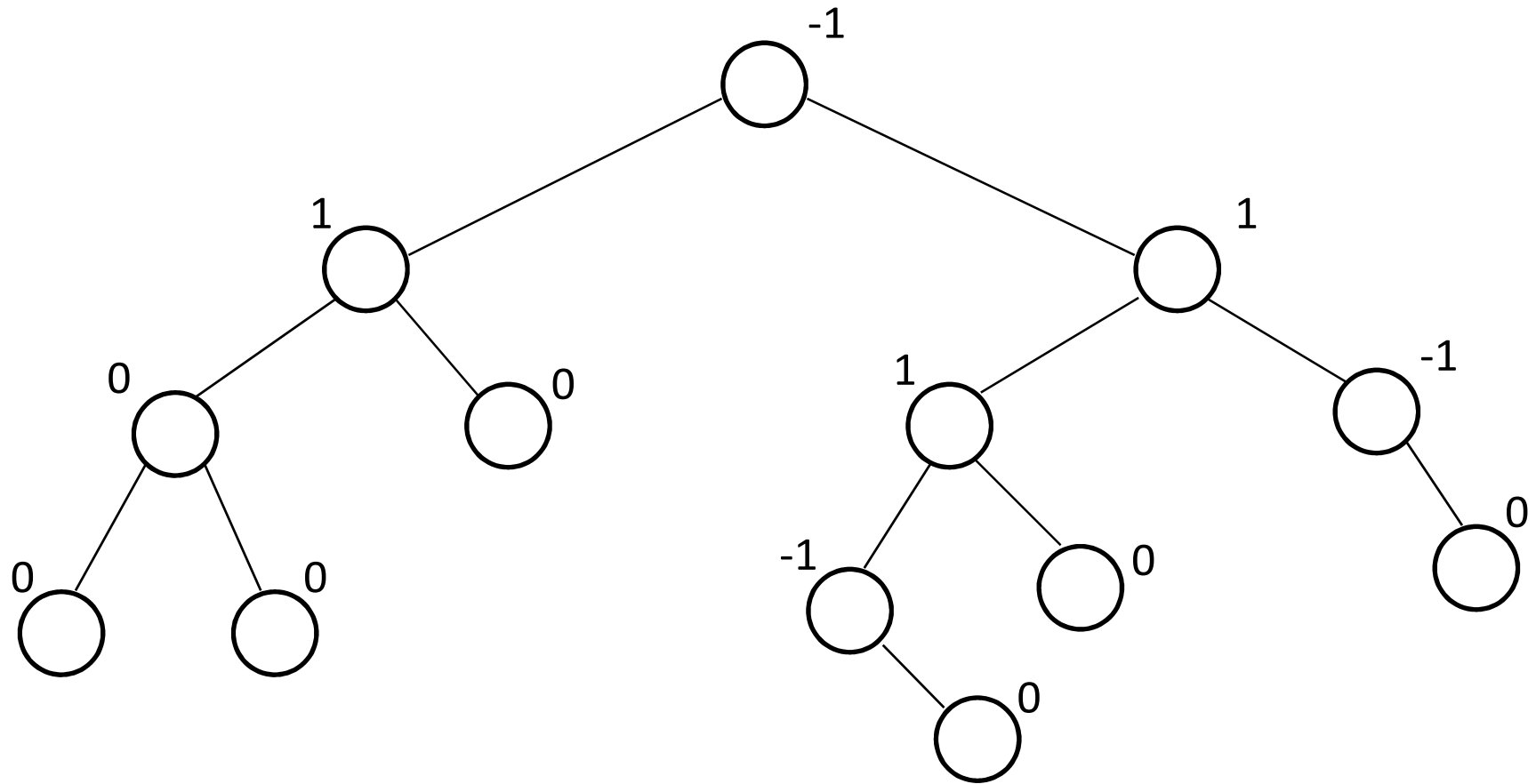
Insert 2 in BST &
complete tree



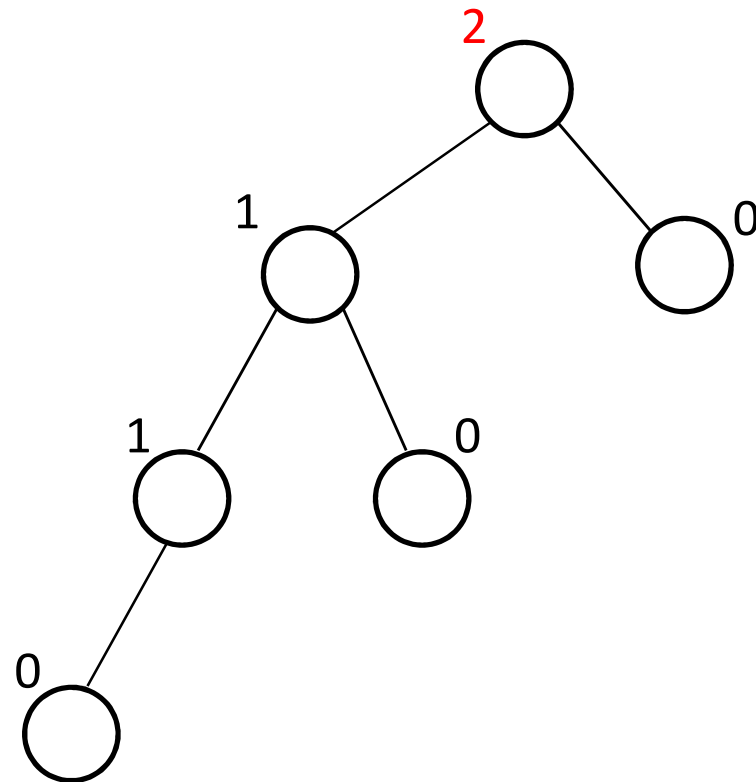
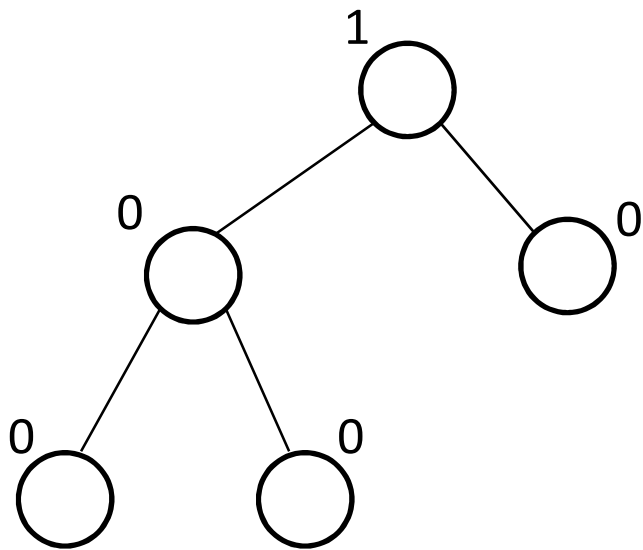
AVL Trees

- Adelson-Velskii and Landis
- Height-balanced binary tree
 - But not perfectly balanced
- For every node x , define its balance factor
$$\text{balance factor of } x = h_{\text{left}}(x) - h_{\text{right}}(x)$$
- Balance factor of every node x is 1, 0, or -1
 - Height of sibling trees differs **no more than 1**

Example of AVL tree

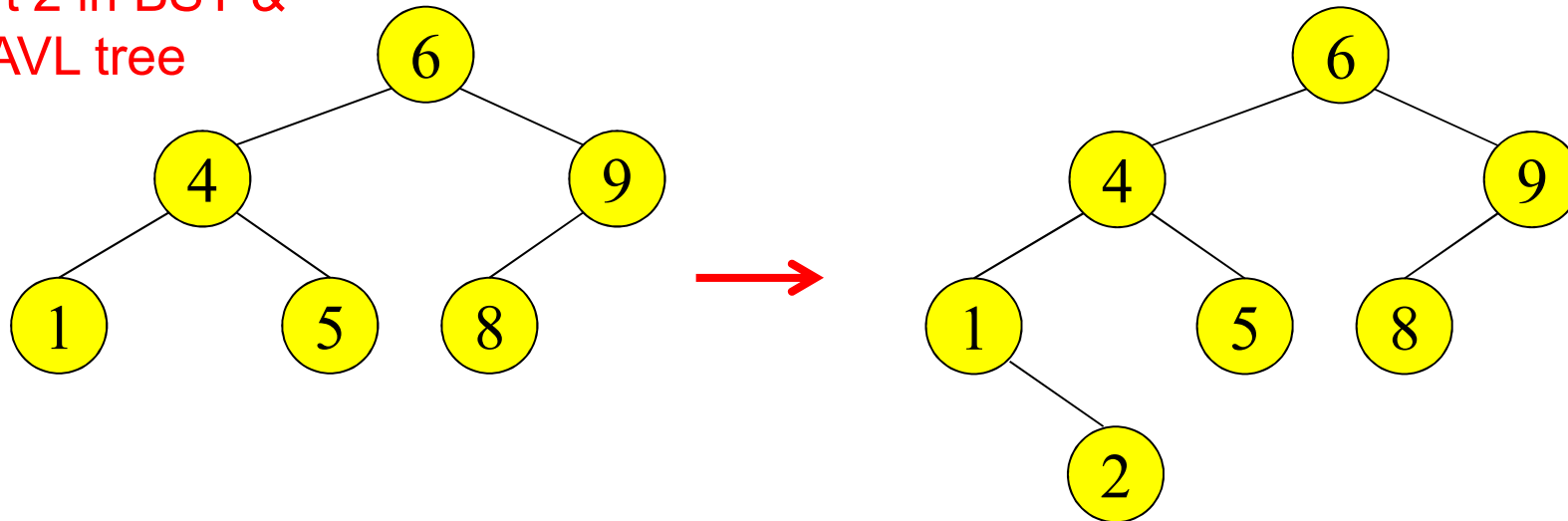


AVL Trees?



Example of AVL tree

Insert 2 in BST &
AVL tree



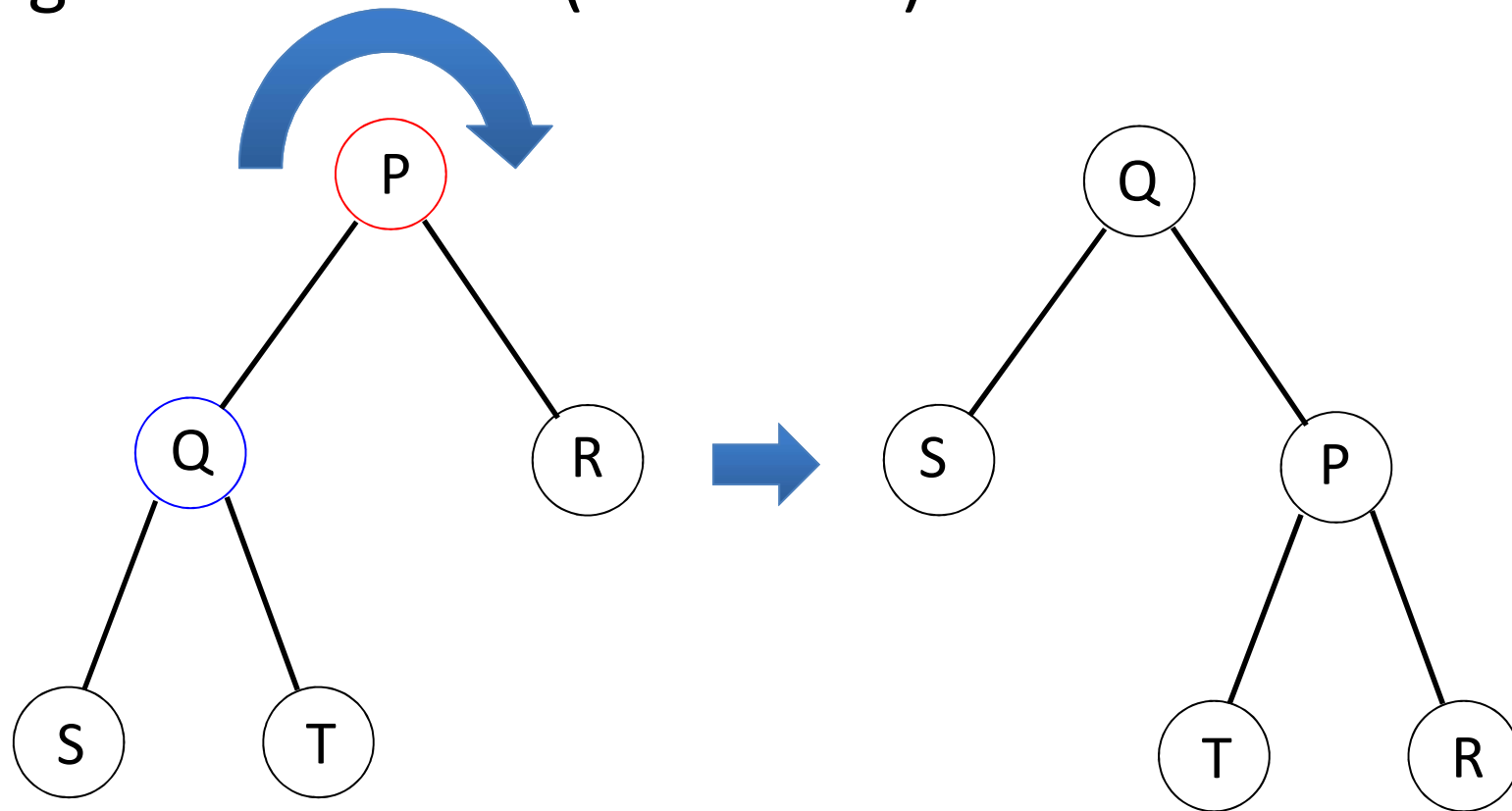
This is not perfectly balanced but height-balanced AVL tree!

Tree Rotation

- Fundamental operation on a binary tree that
 - Is triggered on every insert and delete
 - Adjusts height of the trees
 - Moves one node up and one node down
- It will preserve inorder traversal sequence
 - Binary search tree property

Tree Rotation

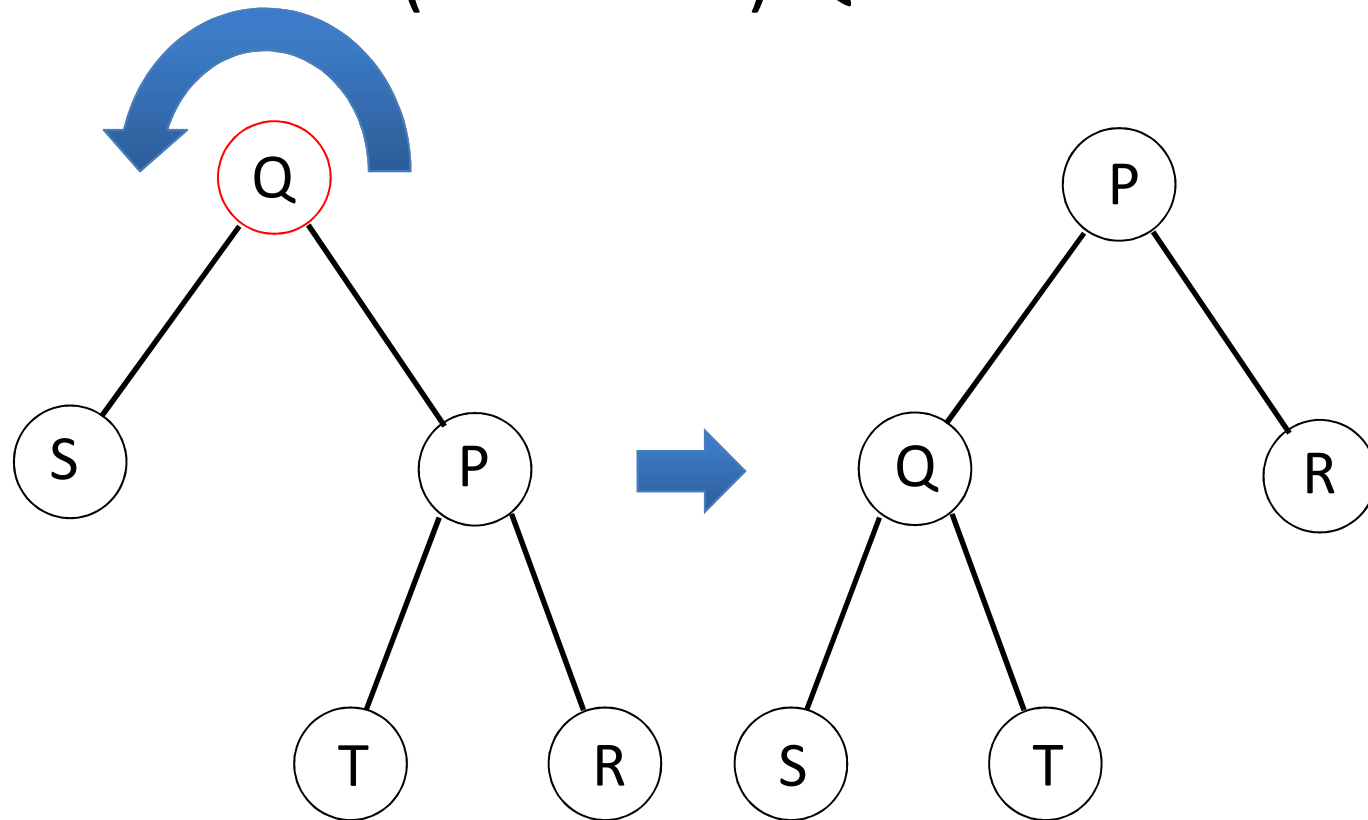
- Right rotation on (rooted at) P



Inorder traversal sequence: S Q T P R

Tree Rotation

- Left rotation on (rooted at) Q

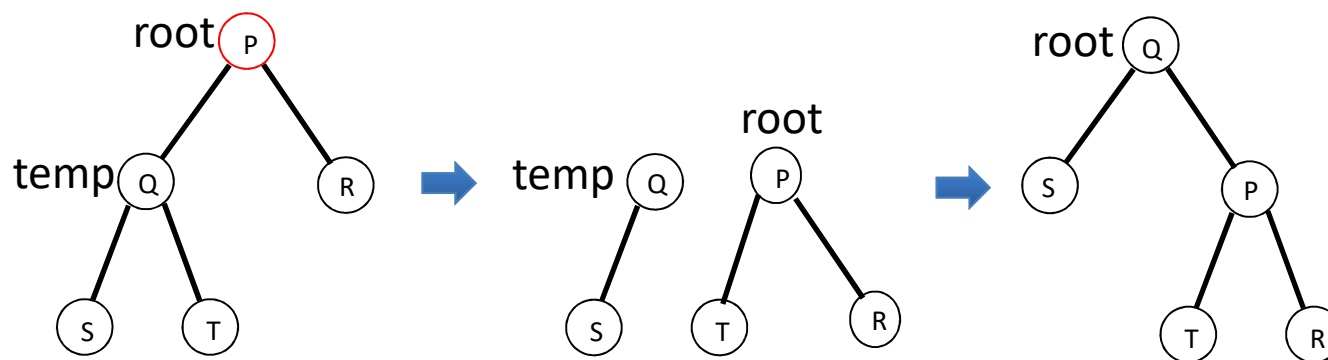


Inorder traversal sequence: S Q T P R

Tree Rotation

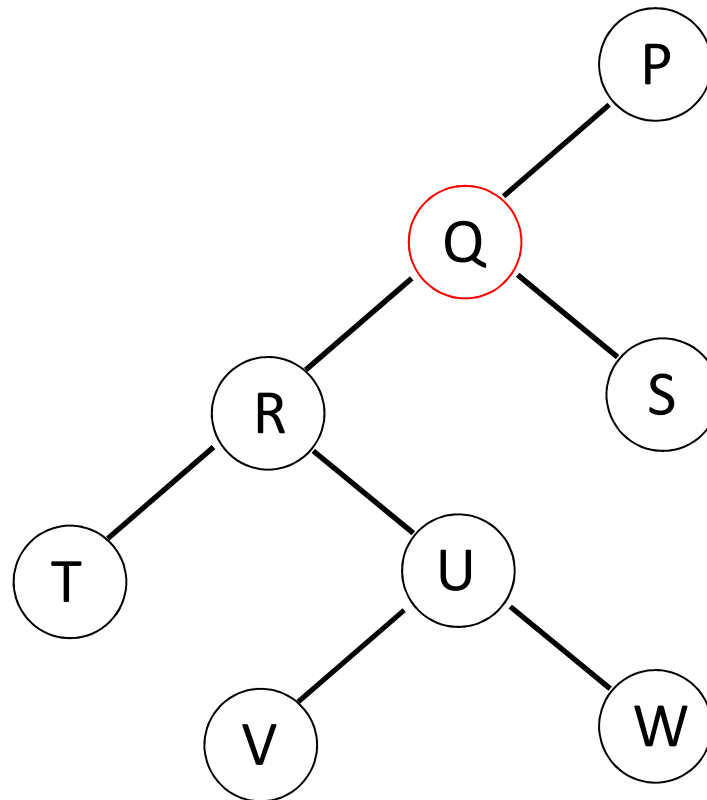
- Pseudo code for right rotation

```
temp = root->leftChild  
root->leftChild = temp->rightChild  
temp->rightChild = root  
root = temp
```

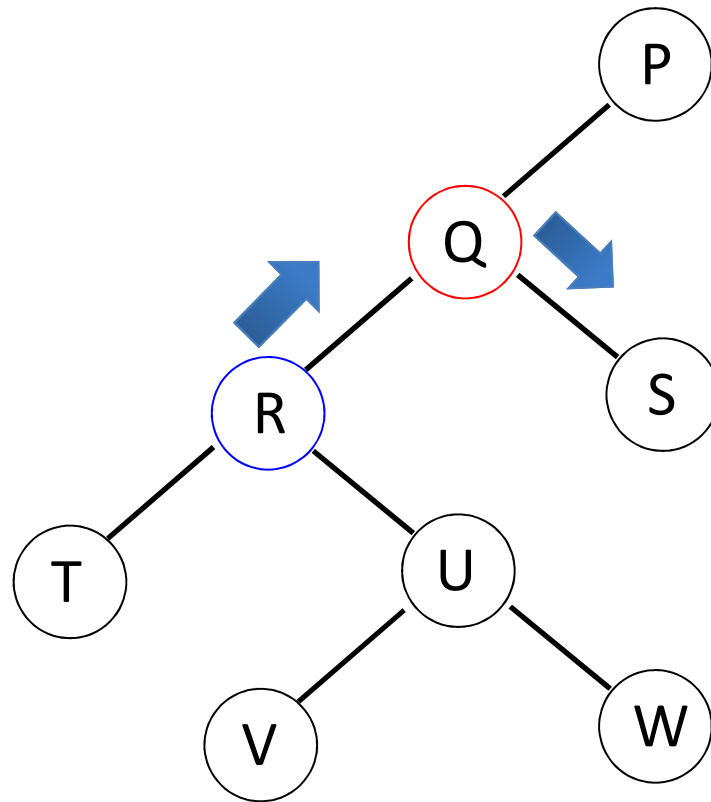


Example

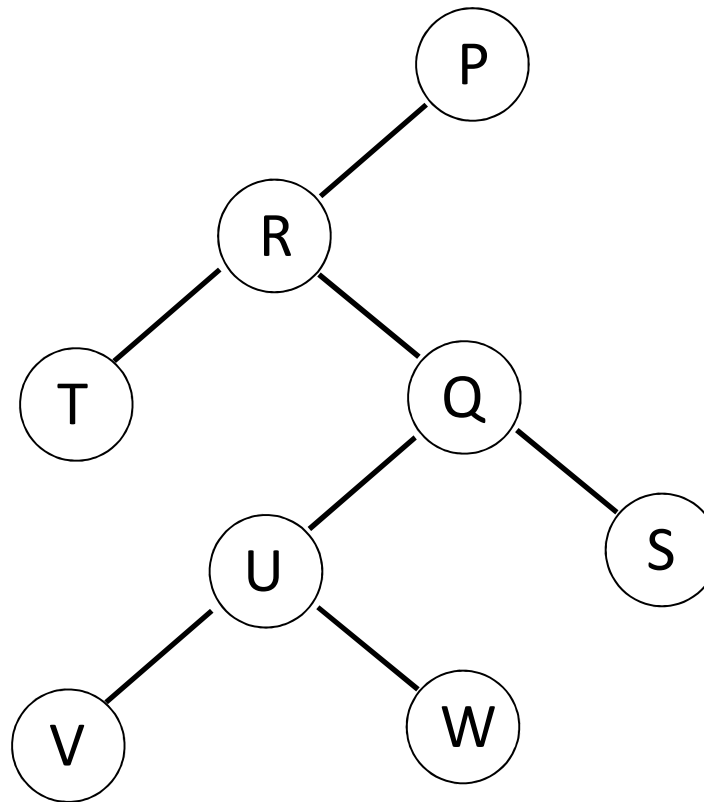
- Right rotate on Q?



Example

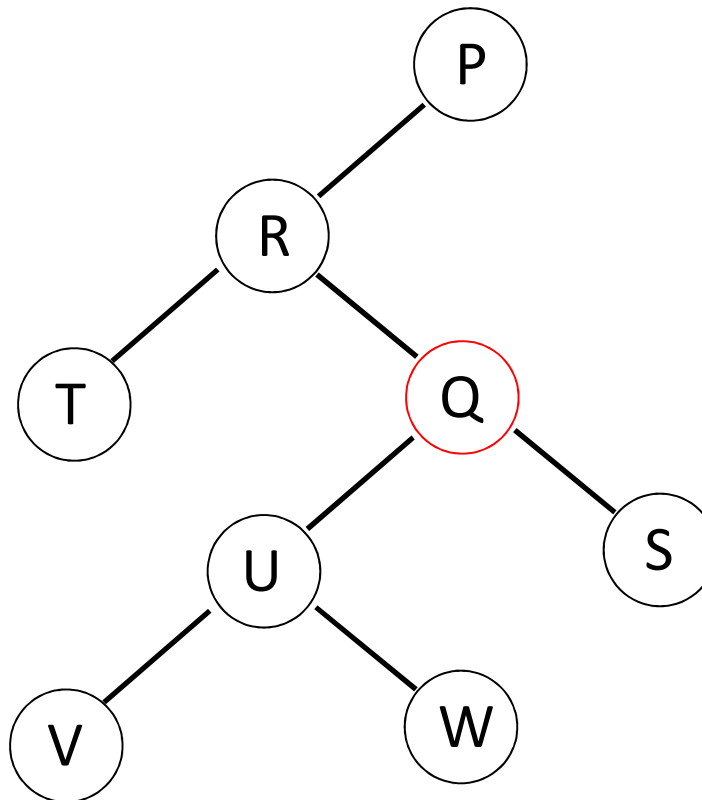


Example

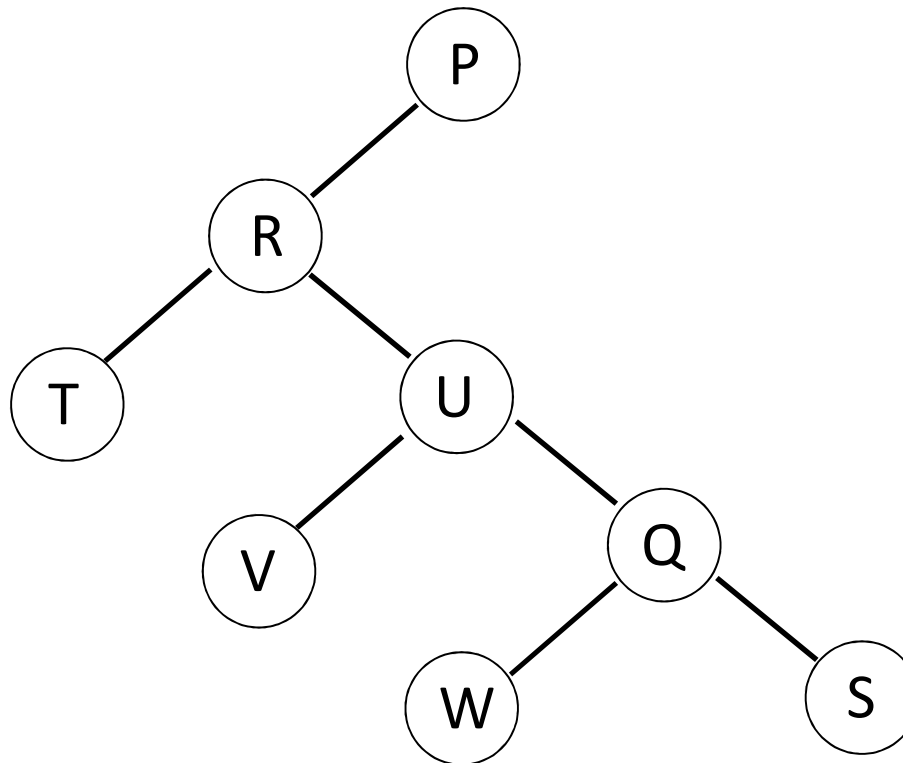


Example

- Right rotate on Q?



Example

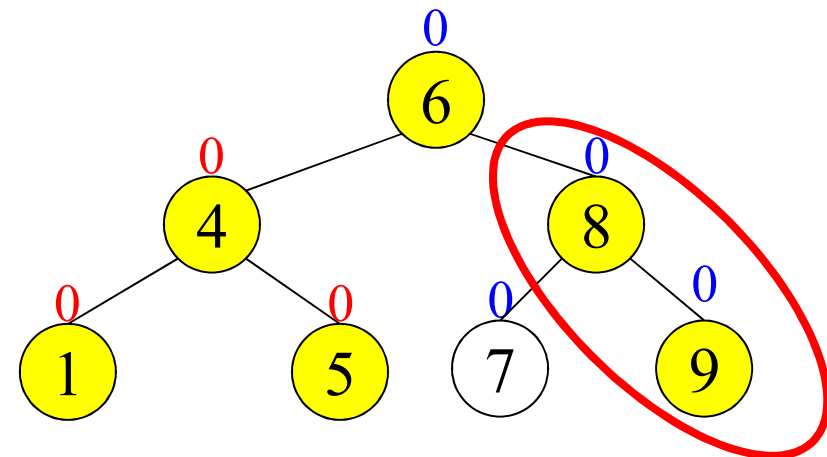
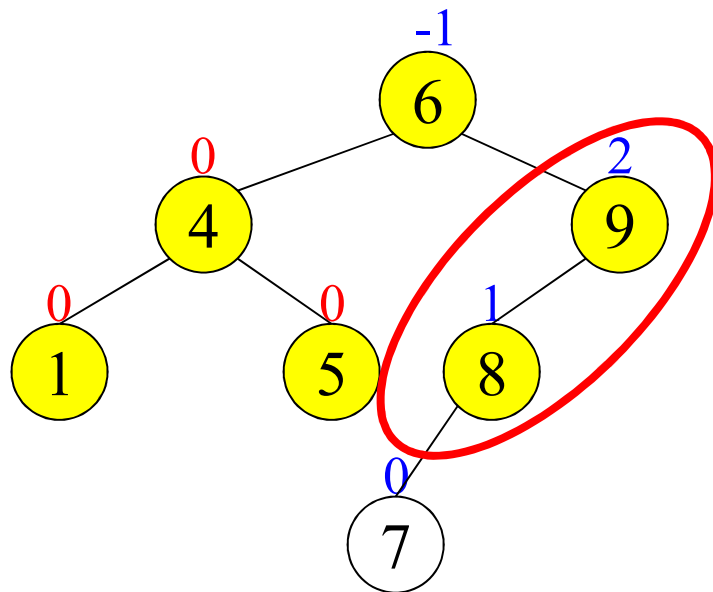


Insert and Rotation in AVL Trees

- Insert operation may cause balance factor to become 2 or -2 for some node
- Only nodes on the path from insertion point to root node have possibly changed in height
- After the Insert,
 - Update heights while moving up to the root from the insertion point
 - If a balance factor is 2 or -2, adjust tree by *rotation*

Single Rotation Example

- Right rotation on 9



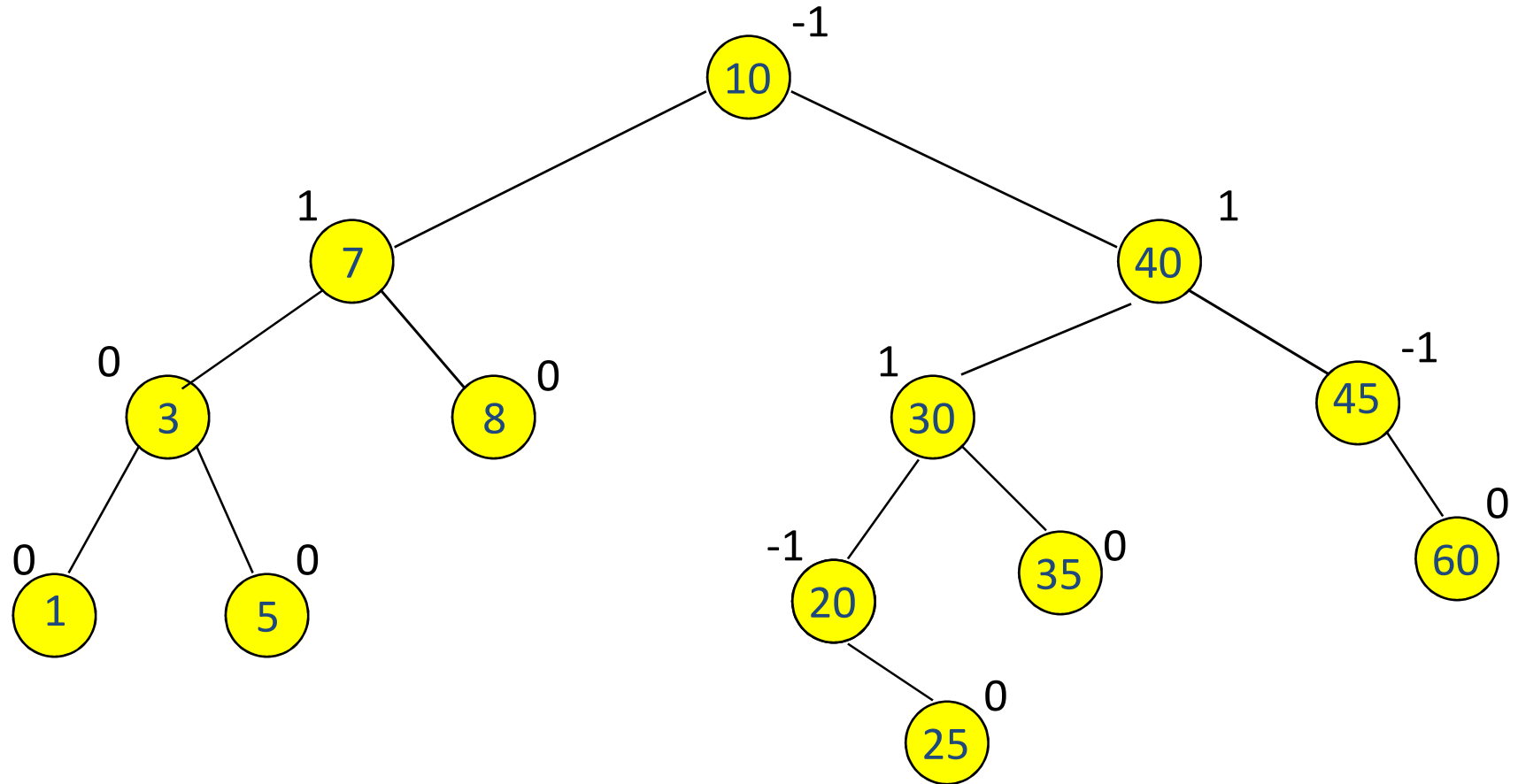
Blue: balance factor changed

Red: balance factor intact

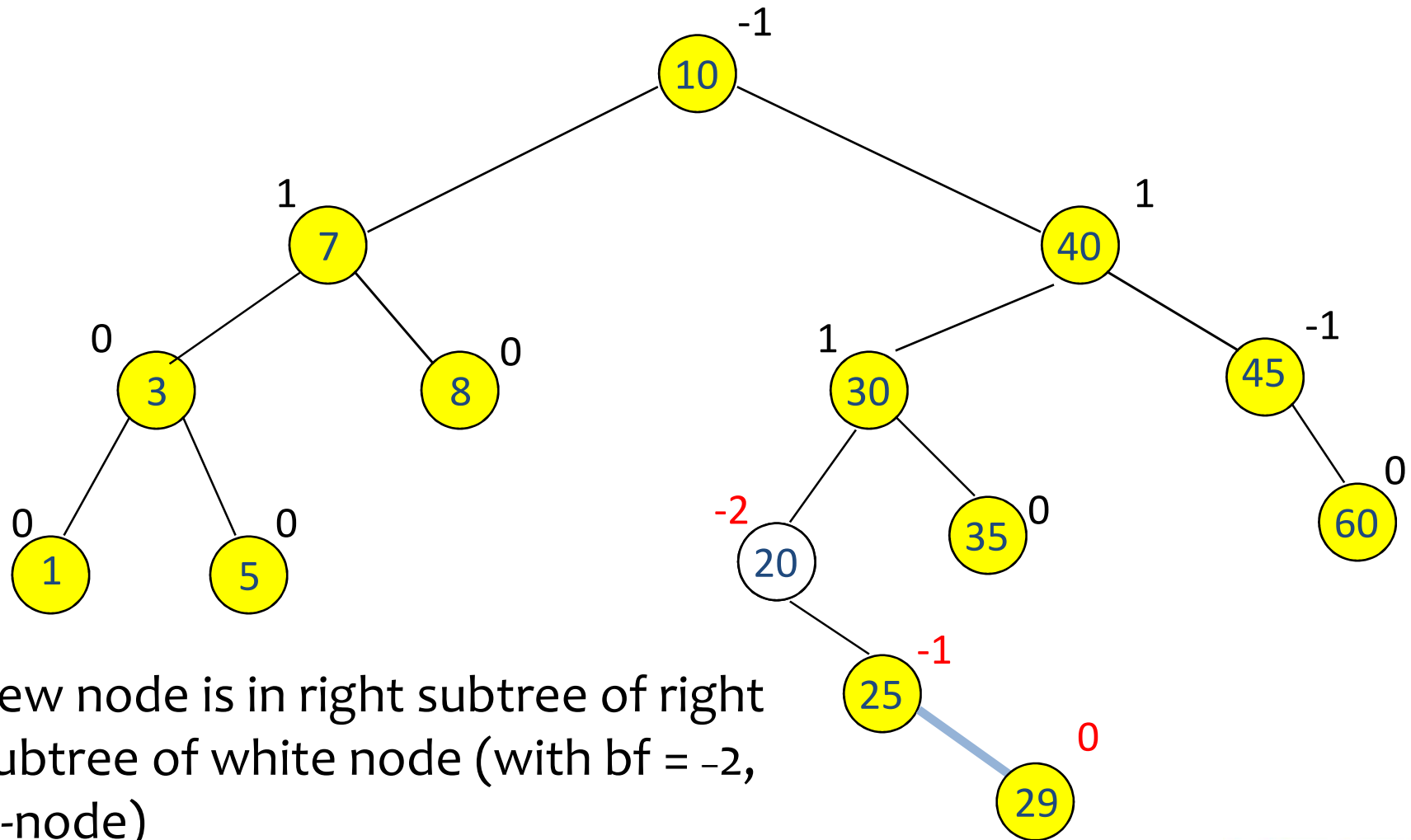
A-Node

- Let A be the nearest ancestor of the newly inserted node whose balance factor becomes +2 or -2 following the insert
- Rotation must be done on A-node

Insert 29

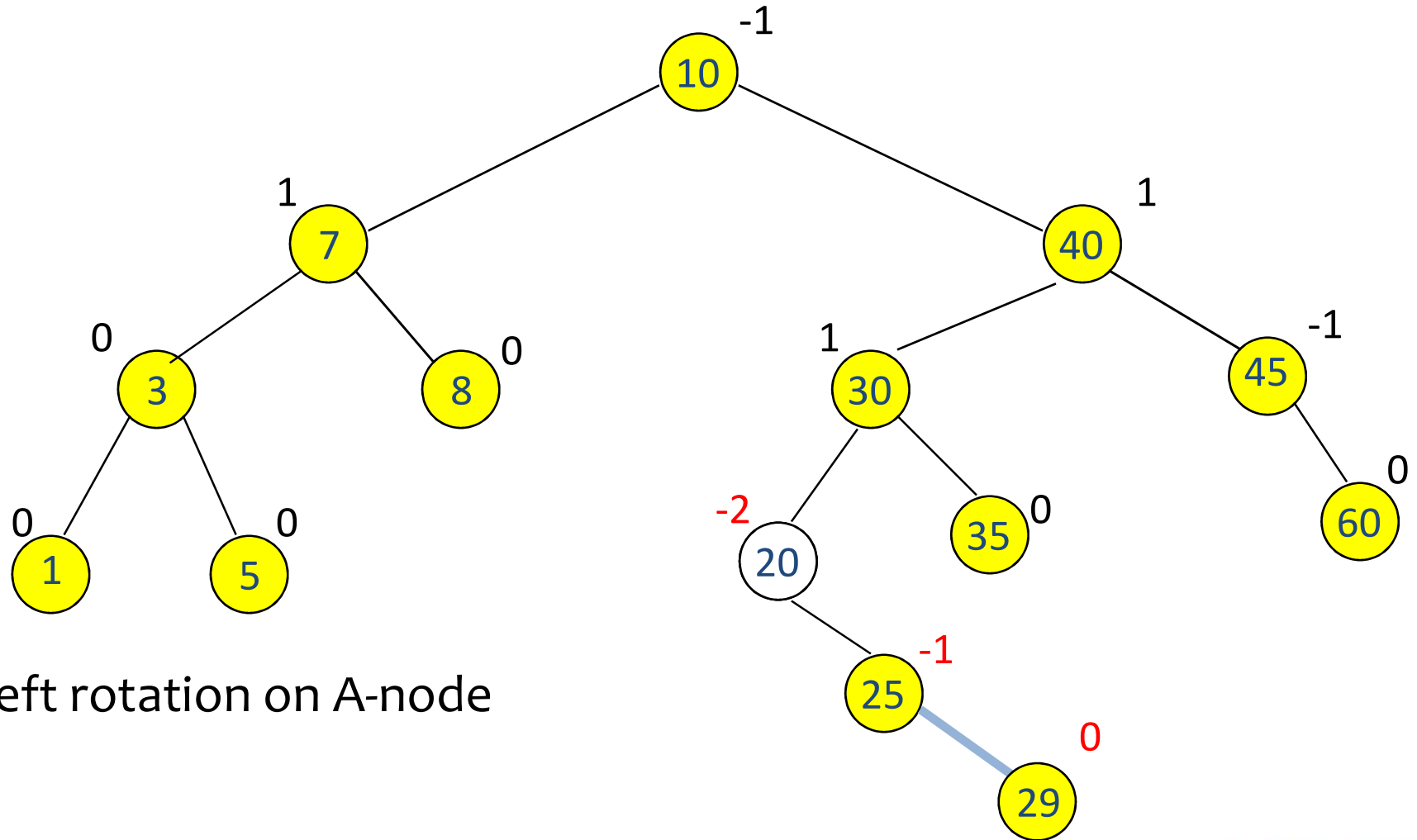


Insert 29

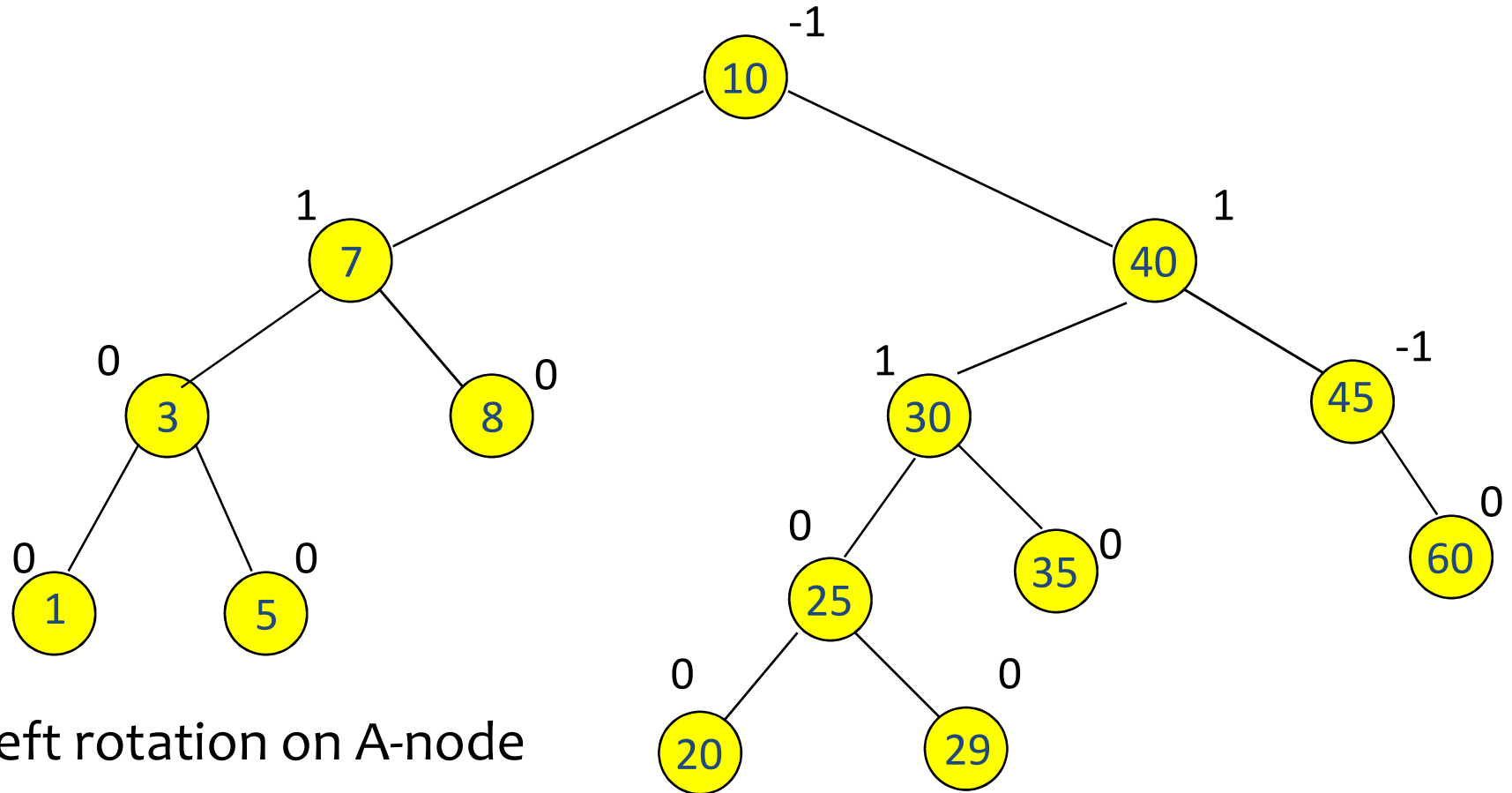


new node is in right subtree of right subtree of white node (with bf = -2, A-node)

Insert 29



Insert 29



Left rotation on A-node