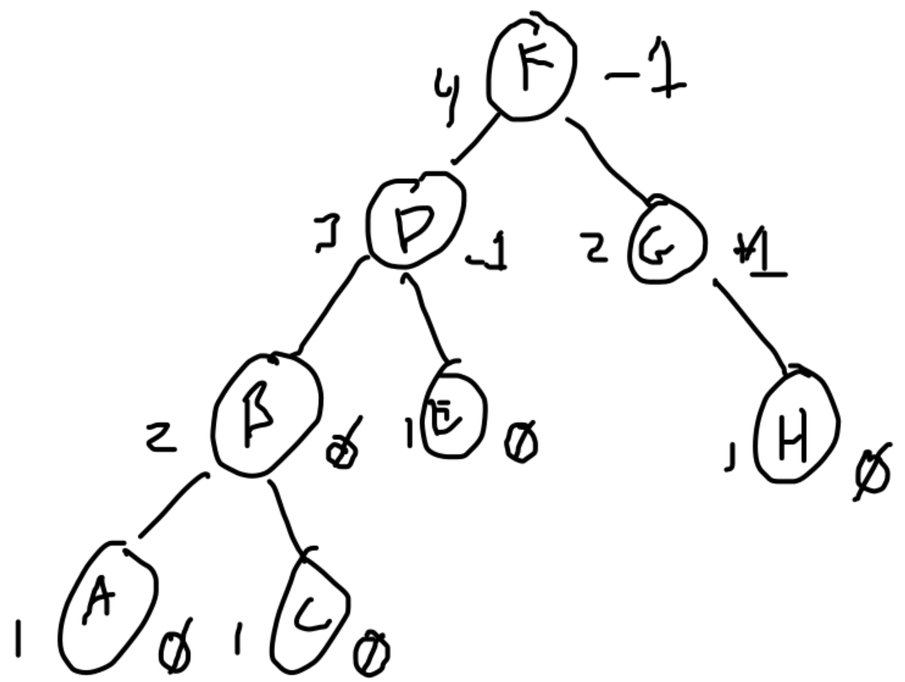
AVL Trees

balance factor: height right subtree - height left subtree

- must be -1, 0 or +1

H  BF



AVL Tree Operations

- Find → no change

- Add

1. add as usual for BST
2. starting at new node and working to root:
   - recalculate balance factors
   - if BF is -2 or +2,
     perform rotation or zig-zag

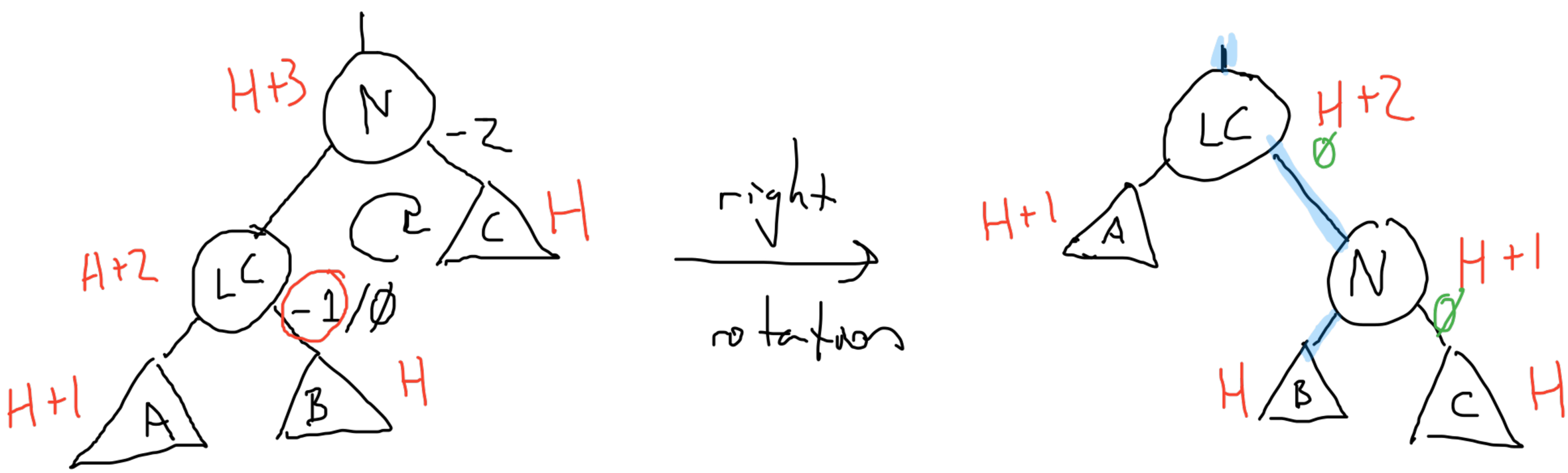
- Remove

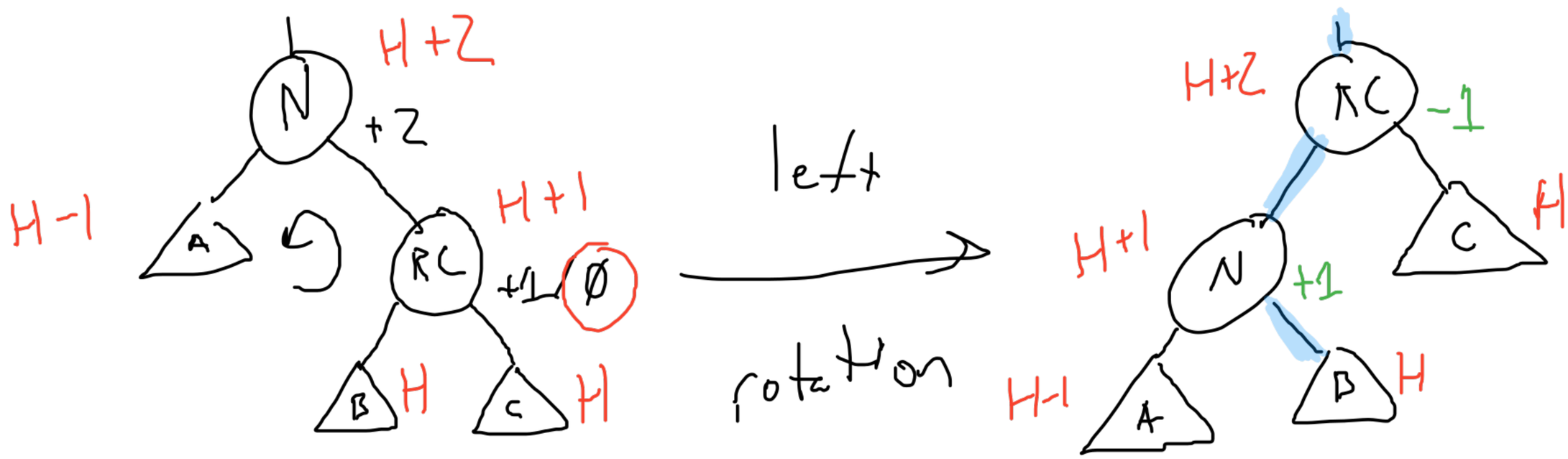
1. Remove using replace strategy
2. Work up, computing BFs and rotating/zig-zagging as needed (≤ to O(h) times)

Rotations

height

balance factor

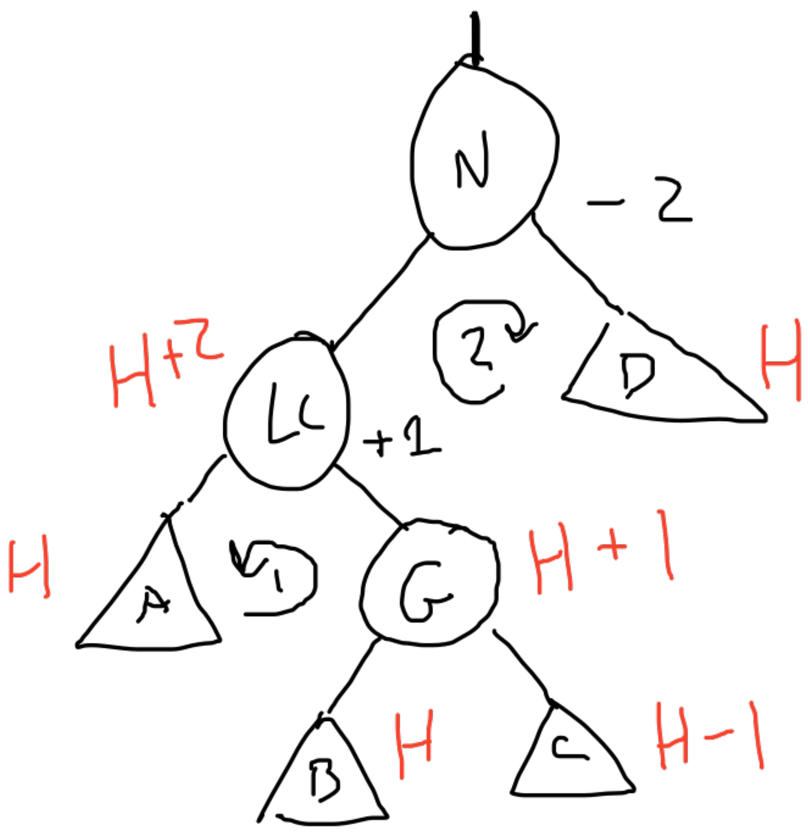




Zig - zags

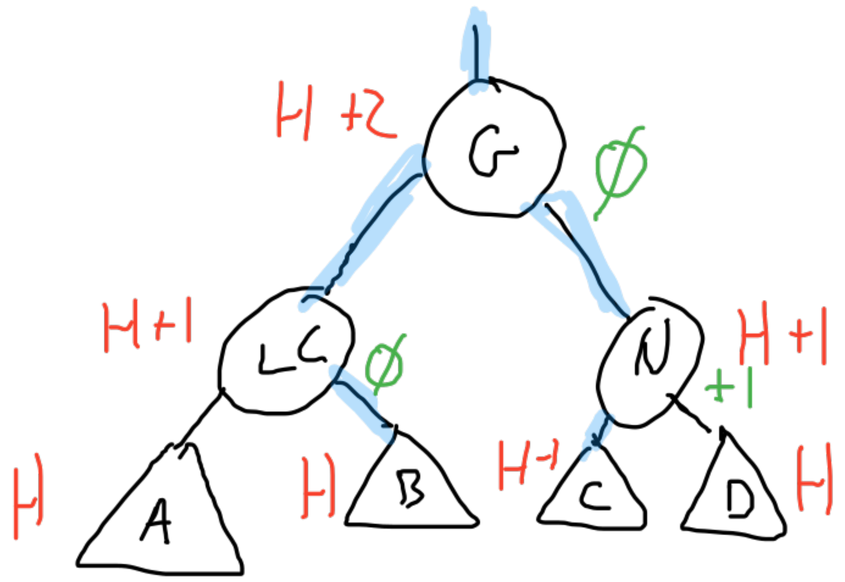
height

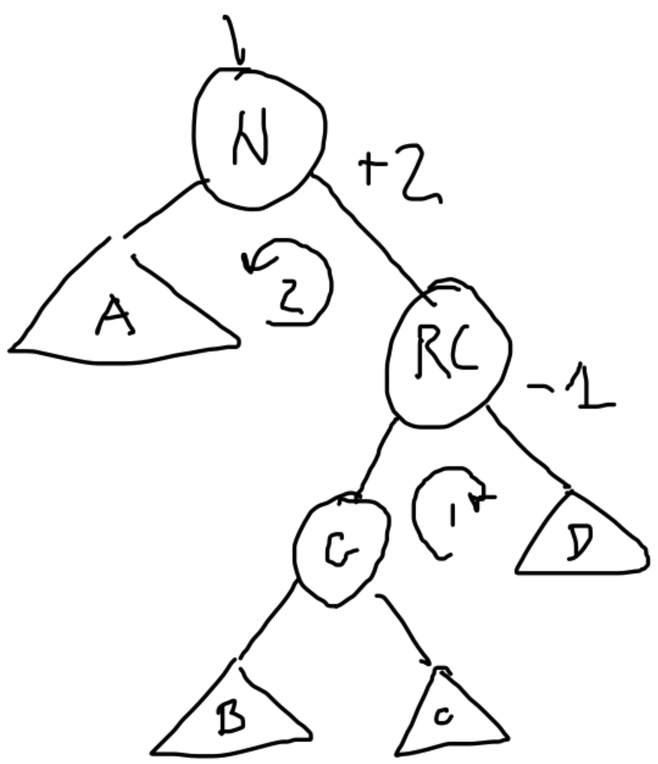
balance factor



left-right

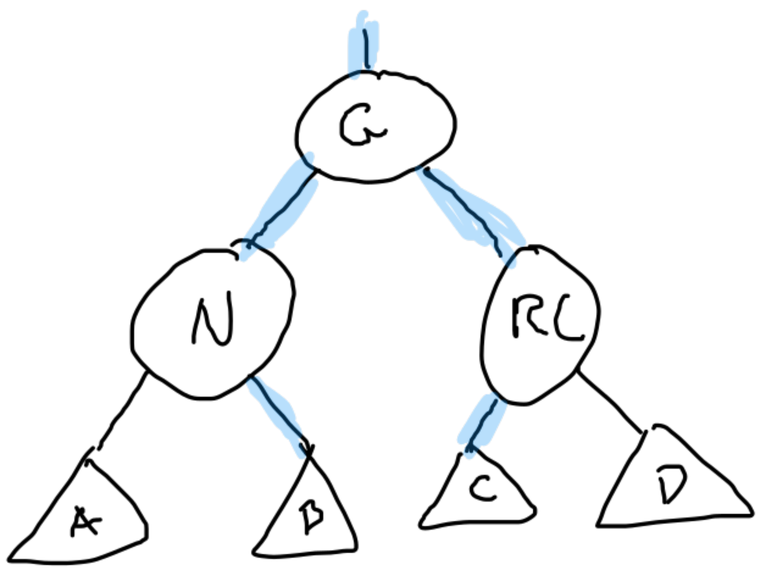
zig-zag





right-Left

zig-Zag



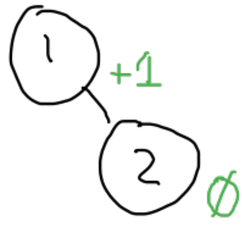
AVL Tree Practice

height

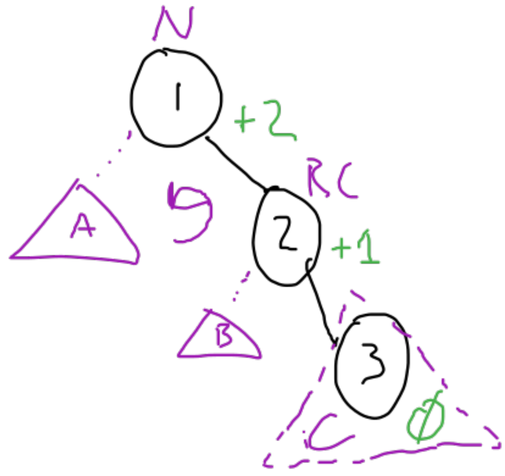
balance factor

Add 1:

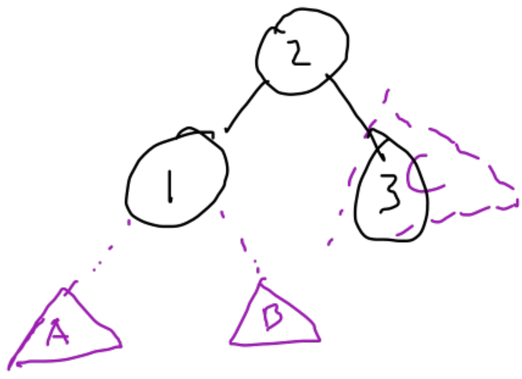
Add 2:



Add 3:

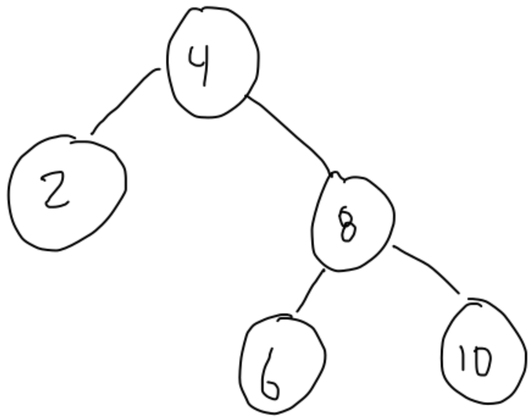


left rotation

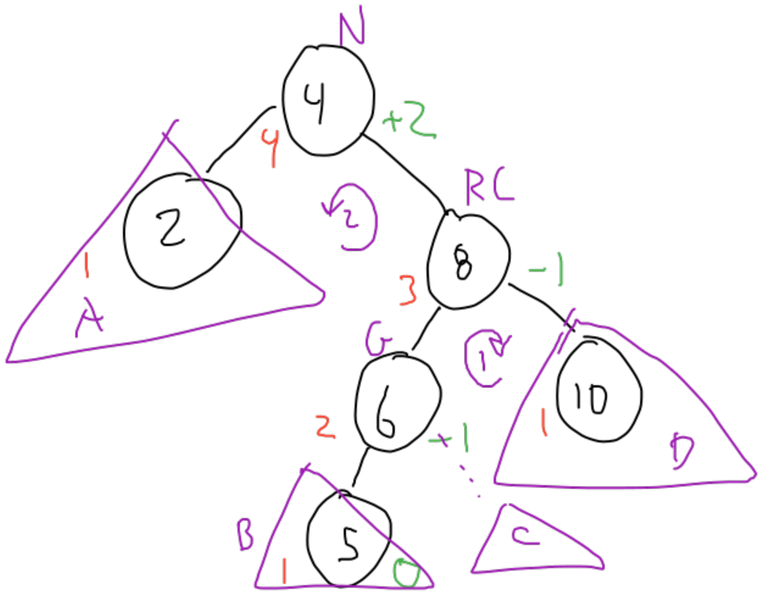


AVL Tree Practice height balance factor

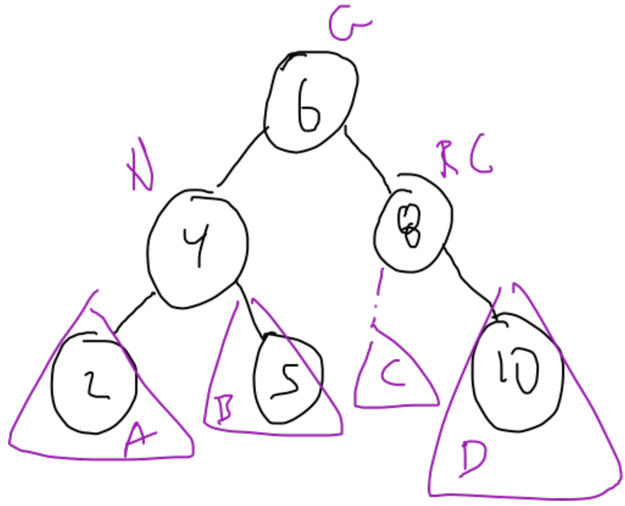
Start:



Add 5:



left-right ↓ zig-zag



Assignment 3

- node class

class TreeNode {

private final TreeNode left;

private