COEN-352 Tutorial #8

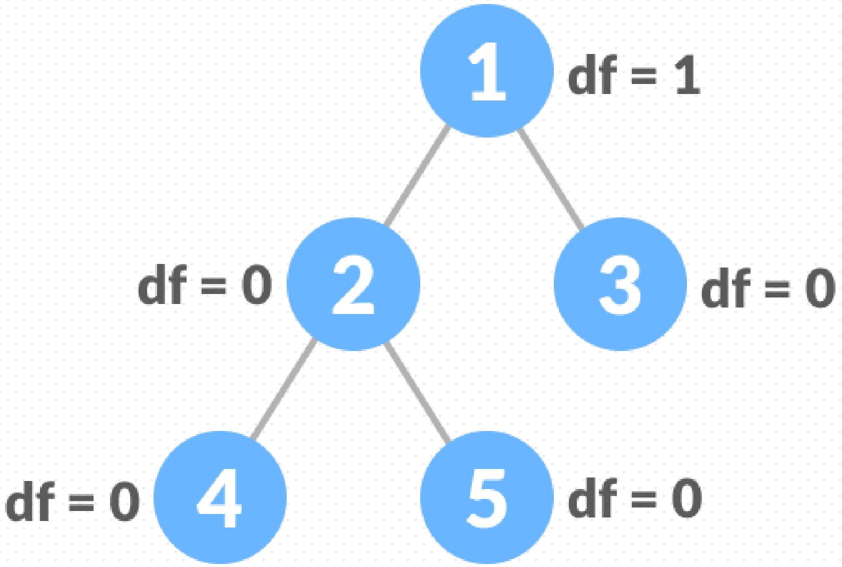
June 8th, 2023

Balanced & Unbalanced Trees

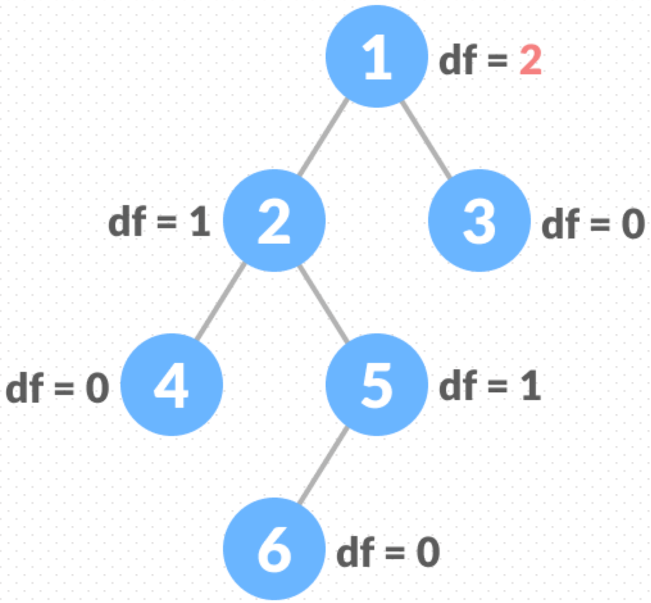
Balanced Tree: a binary tree in which the height of the left and right subtree of any node differ by not more than 1:

- Difference between the left and the right subtree for any node is not more than one
- The left subtree and right subtree are balanced

Balanced Tree



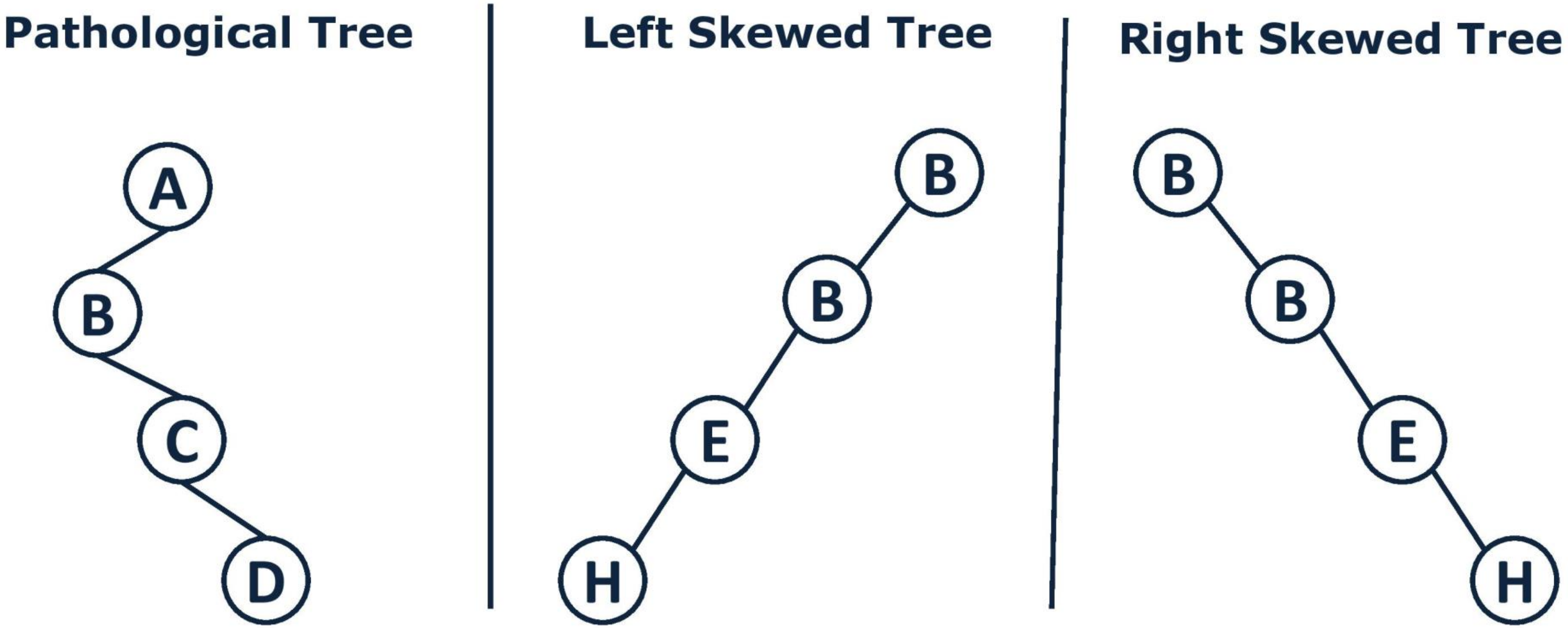
Unbalanced Tree



df = |height of left child - height of right child|

Useless Binary Search Tree

- What is the worst-case Binary Search Implementation?



Source: https://codepumpkin.com/binary-tree-types-introduction/

Self-Balancing Binary Trees

Self-balancing Binary Search Trees (SBBST):

- A type of BST which tries to balance itself, in case of arbitrary insertions and deletions, by using **rotations** mainly.
- The height is typically maintained in order of **Log n** so that all operations take **O(Log n)** time on average.

Types of SBBST:

- **Red-black tree:** A tree that balances itself based on coloring
- **AVL tree:** Tree rotations are done on the Nodes
- **2-3 tree:** Each node can either have two or three children

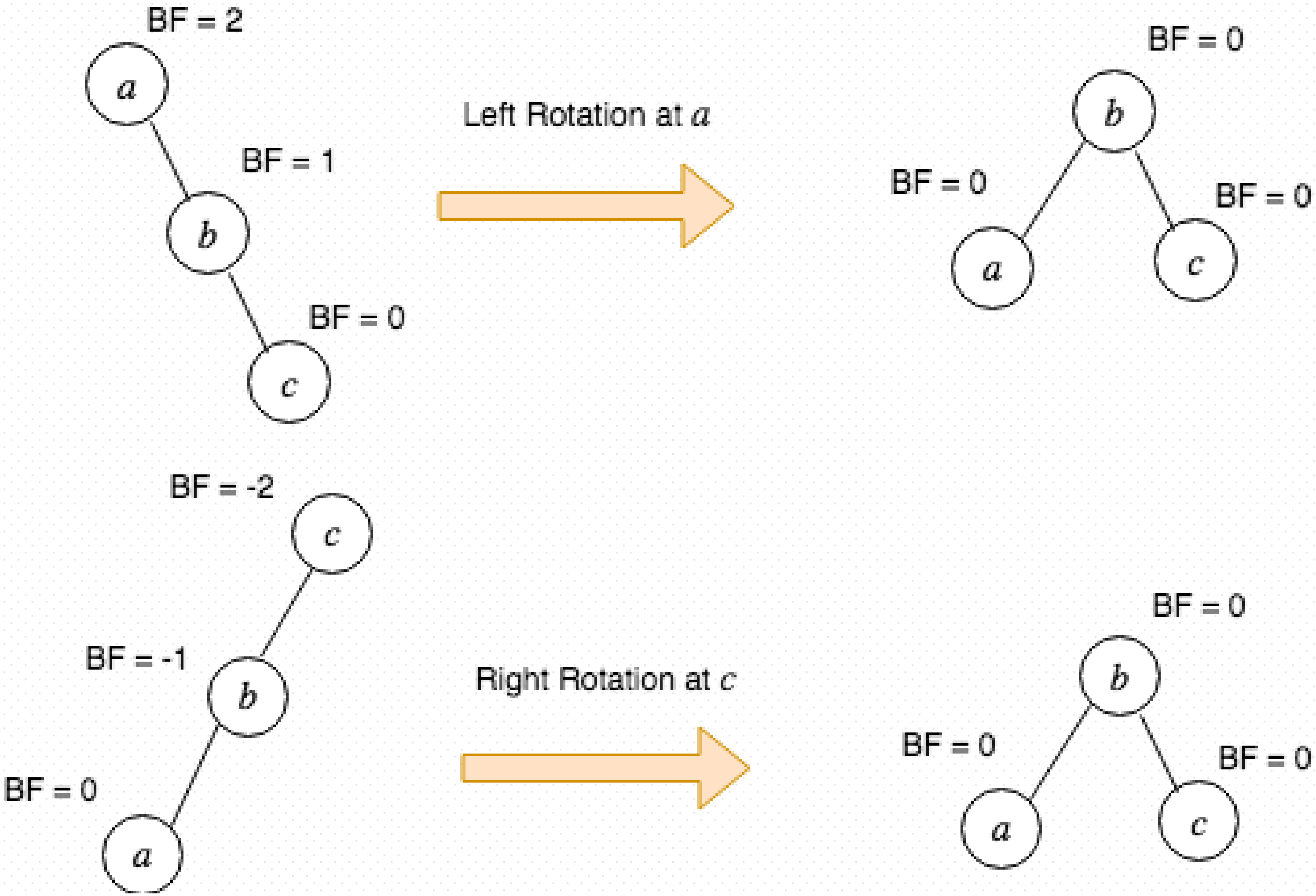
AVL Trees

AVL Trees are height balanced binary search trees that does only frequent rotations after arbitrary insertions and deletions.

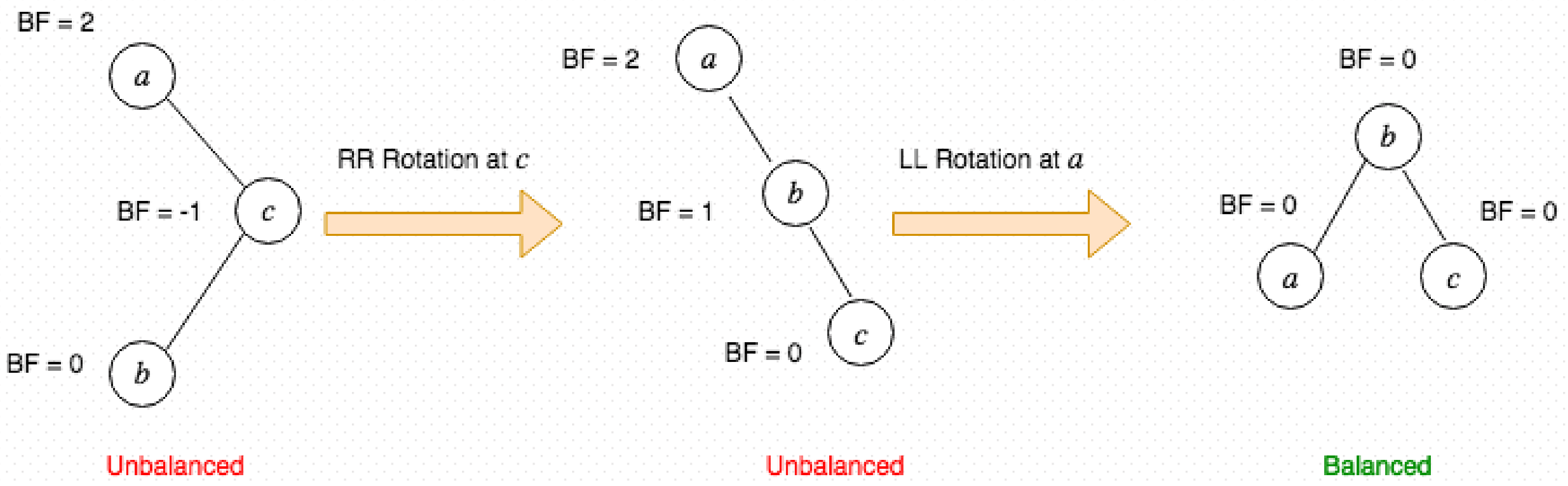
Rotations:

- Left Rotation (LL)
- Right Rotation (RR)
- Left-Right Rotation (LR)
- Right-Left Rotation (RL)

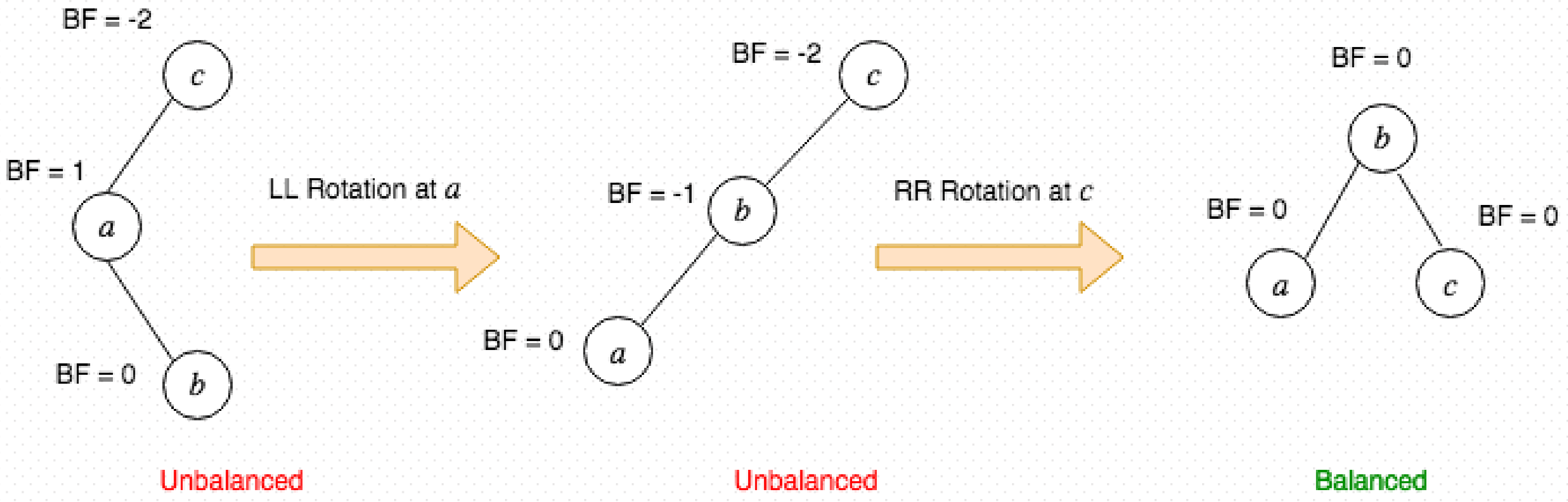
AVL Trees - Left and Right Rotations



AVL Trees – Left-Right and Right-Left Rotations



LR Rotation



RL Rotation

EXERCISE

Exercise: Write an algorithm that checks if a Binary Search Tree is Balanced or not.

- Assume the balanced tree definition used here.
- **Hint:** check out the code for the `height()` function.

THANK YOU