ITS202: Algorithms and Data Structures

Advanced Data Structures

Ms. Sonam Wangmo

Gyalpozhing College of Information Technology
Royal University of Bhutan

November 24, 2020

Balanced Search Trees: AVL

Height-Balance Property or Balance Factor

For every internal position p of T , the heights of the children of p differ by at most 1.

Balance

Balance factor= height(left subtree) - height(right subtree).

AVL

Any binary search tree T that satisfies the height-balance property is said to be an AVL tree, named after the initials of its inventors: **Adel'son-Vel'skii and Landis**.

$|B(n)| <= 1$

Balanced Search Trees: AVL

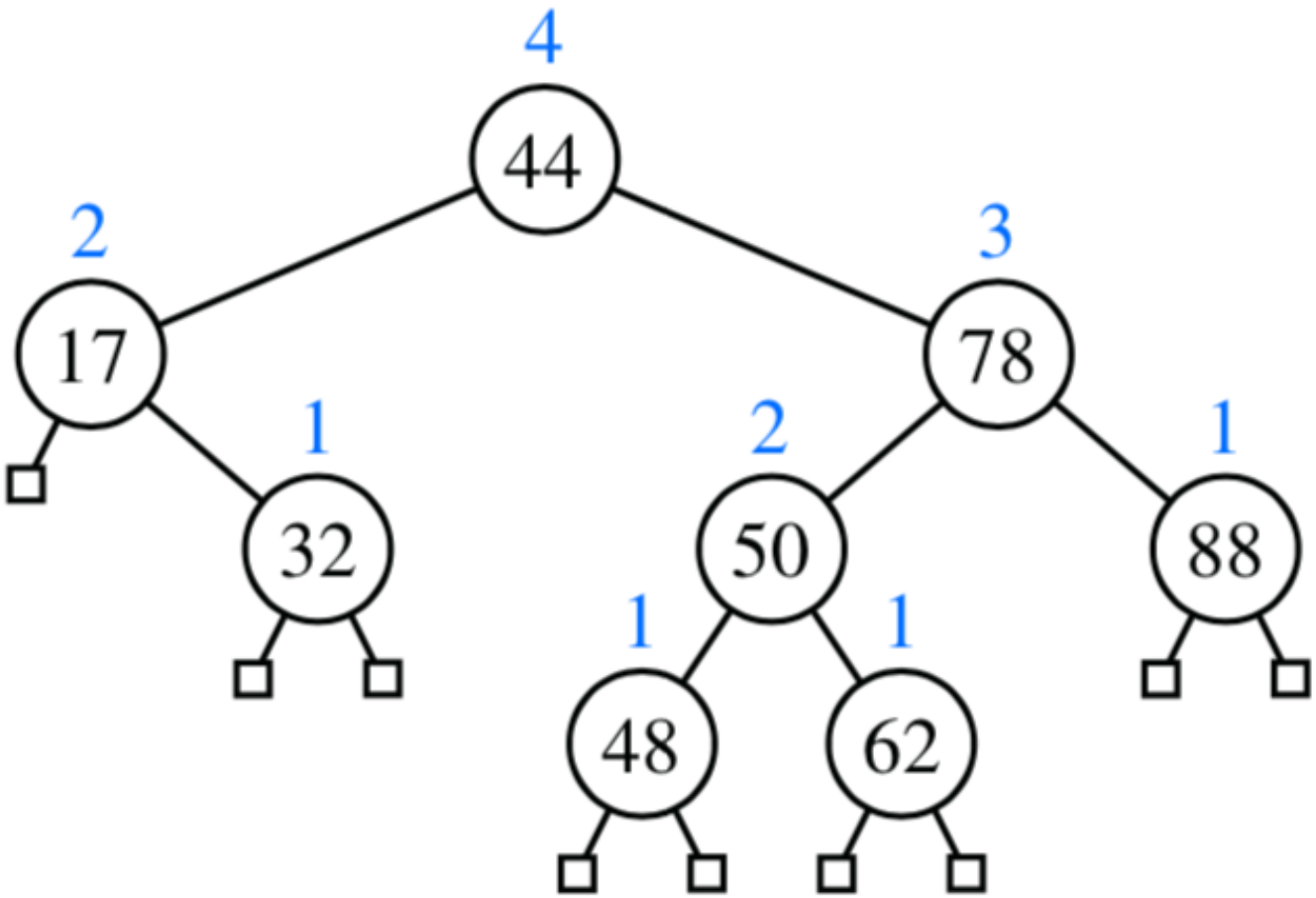

Figure 1: An example of an AVL tree

Balanced Search Trees: AVL

Height of a node

Height of longest path from it down to a leaf.
H($\emptyset$) = -1
H(Single Node) = 0
Height of a node = Maximum[height(leftchild), height(right child)]+1

Worsecase is when the right subtree has height 1 more than left subtree for every node.

Balanced Search Trees: AVL

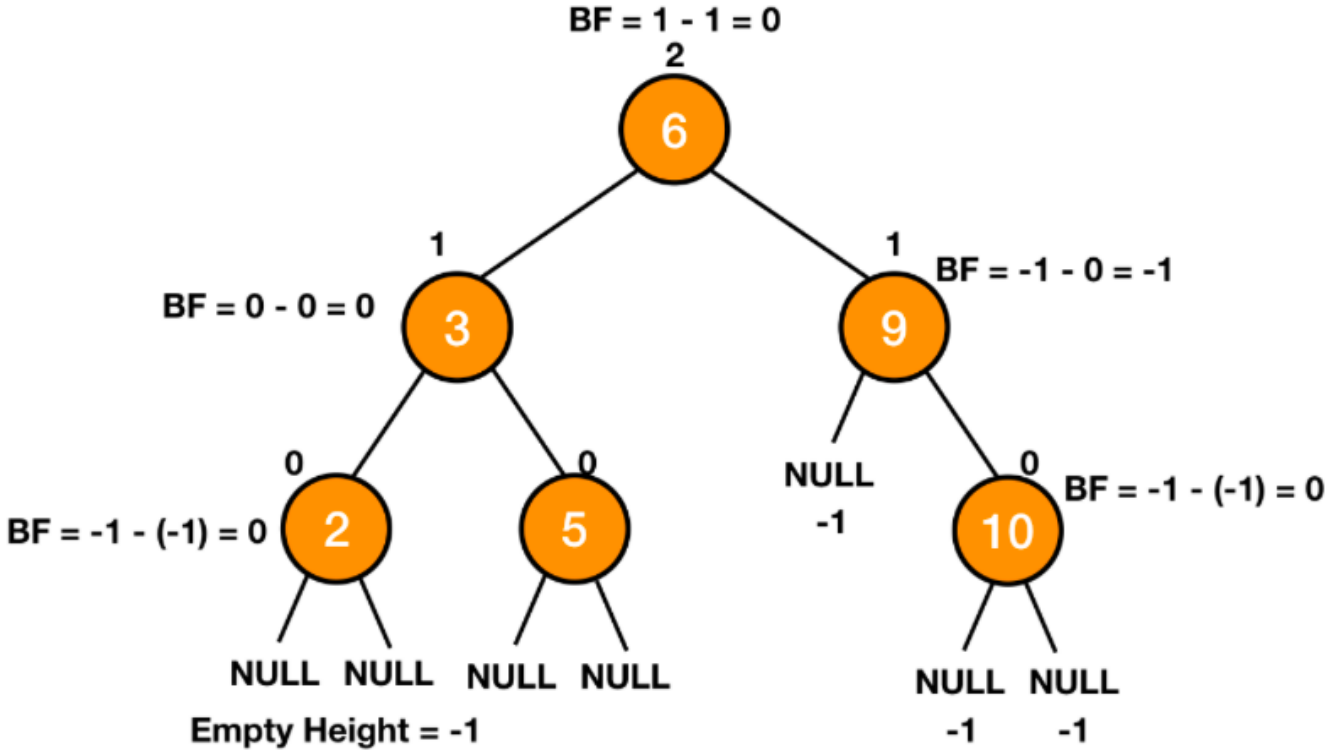

Figure 2: Balanced BST:AVL

Note: Positive Balance: Left-Heavy

Negative Balance: Right-Heavy

B(n)=H(T_L) - H(T_R)

Insertion in AVL Tree

1. Simple BST indert
2. Fix the AVL property from changed node up.
3.

Note: Inserting a new node can cause the balance factor of some node to become 2 or -2. In that case, we fix the balance factors by use of rotations.

AVL Tree: Rotations

Rotation means maintaining the BST invariant at the same time as maintaing the balance threshold.

Rotations fix imbalance.

Left and Right Rotations

Left-Heavy

1. Right rotation
2. Left-Right rotation

Right-Heavy

1. Left rotation
2. Right-Left rotation

Rotations: Right

When we do rotations, focus on 2 nodes, x and y

Insert 3,2,1

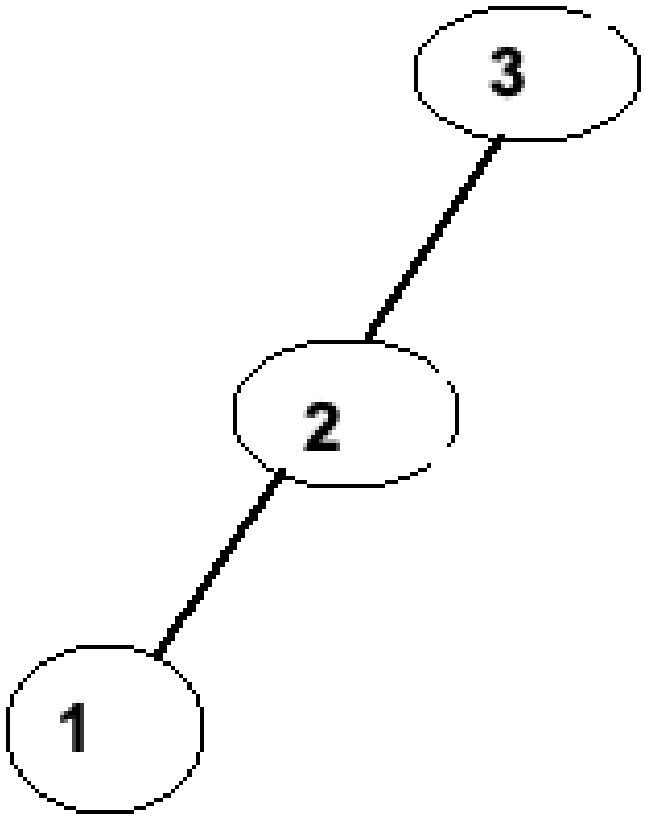

Figure 3: BST

Rotations: Right

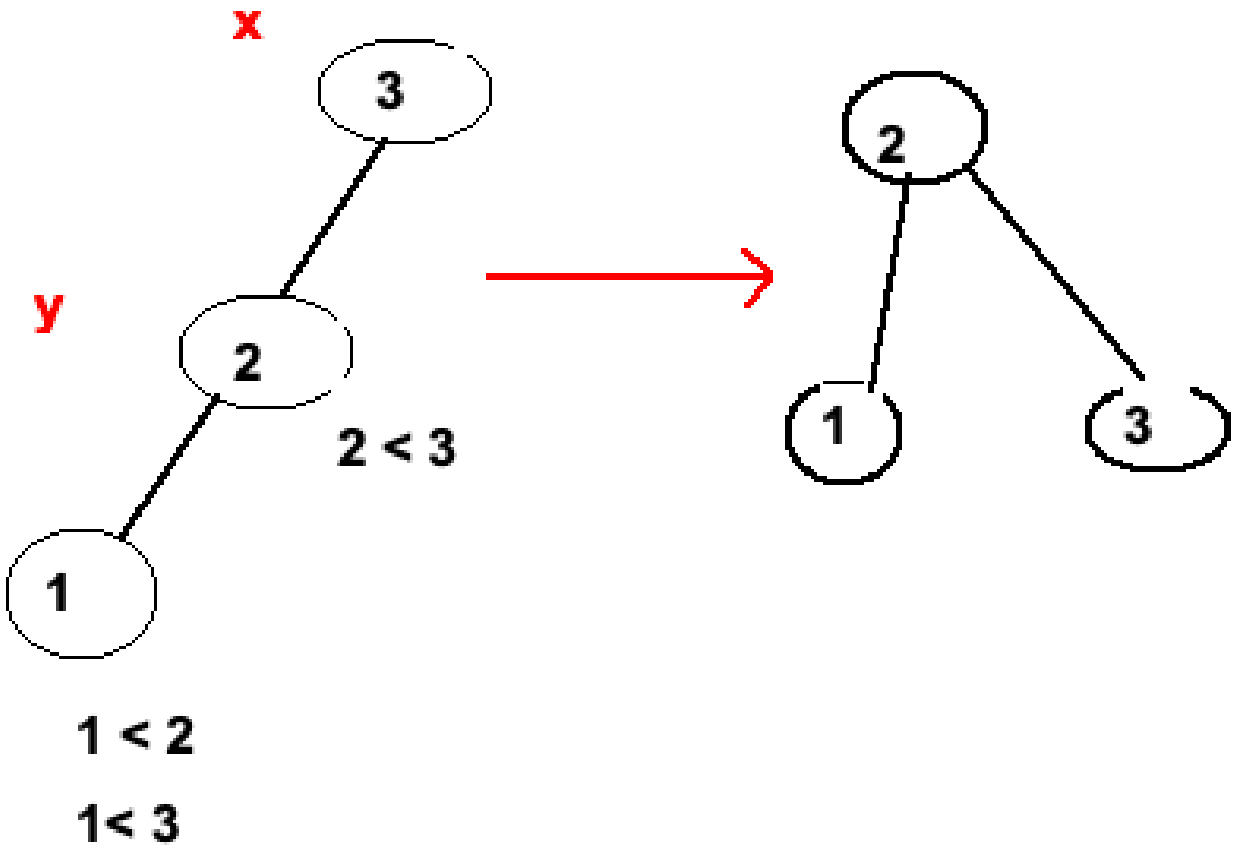

Figure 4: AVL Tree

Rotations: Left-Right

When we do rotations, focus on 2 nodes, x and y

Insert 3,1,2

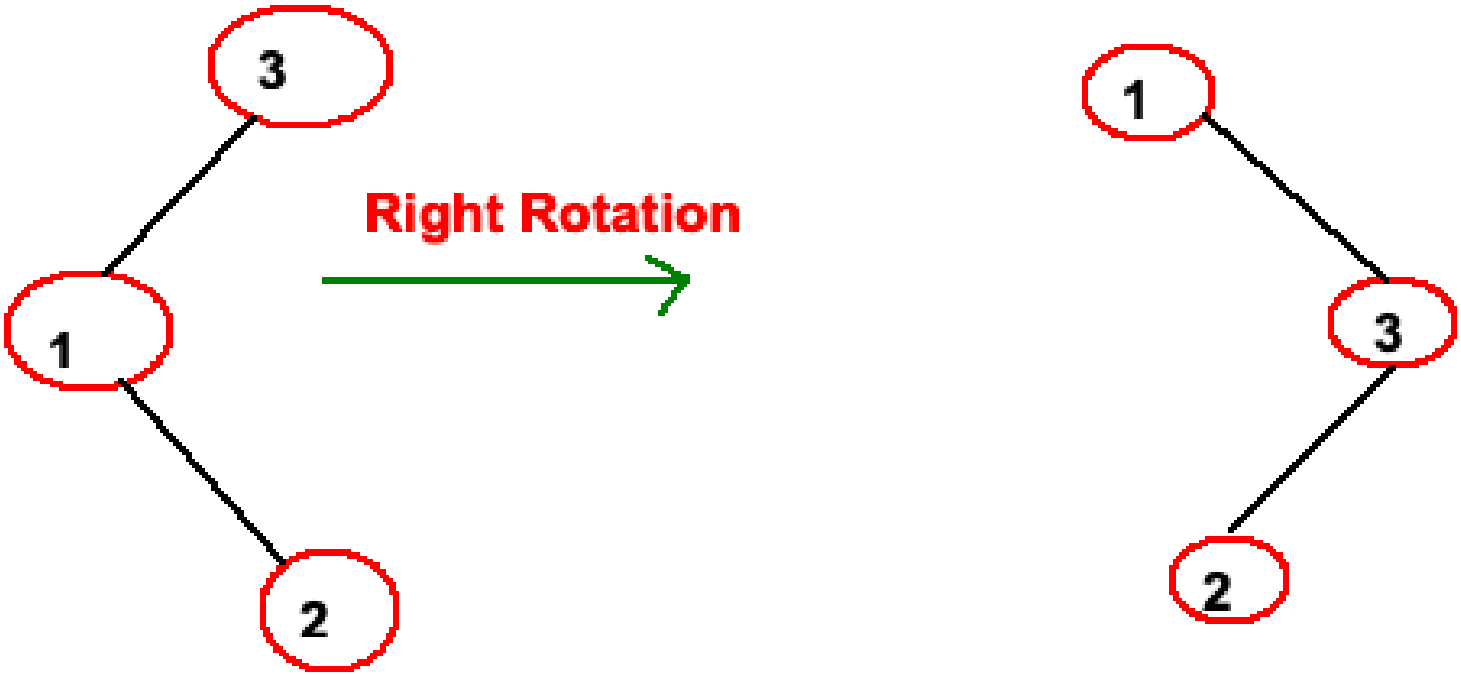

Figure 5: Right rotation

Rotations: Left-Right

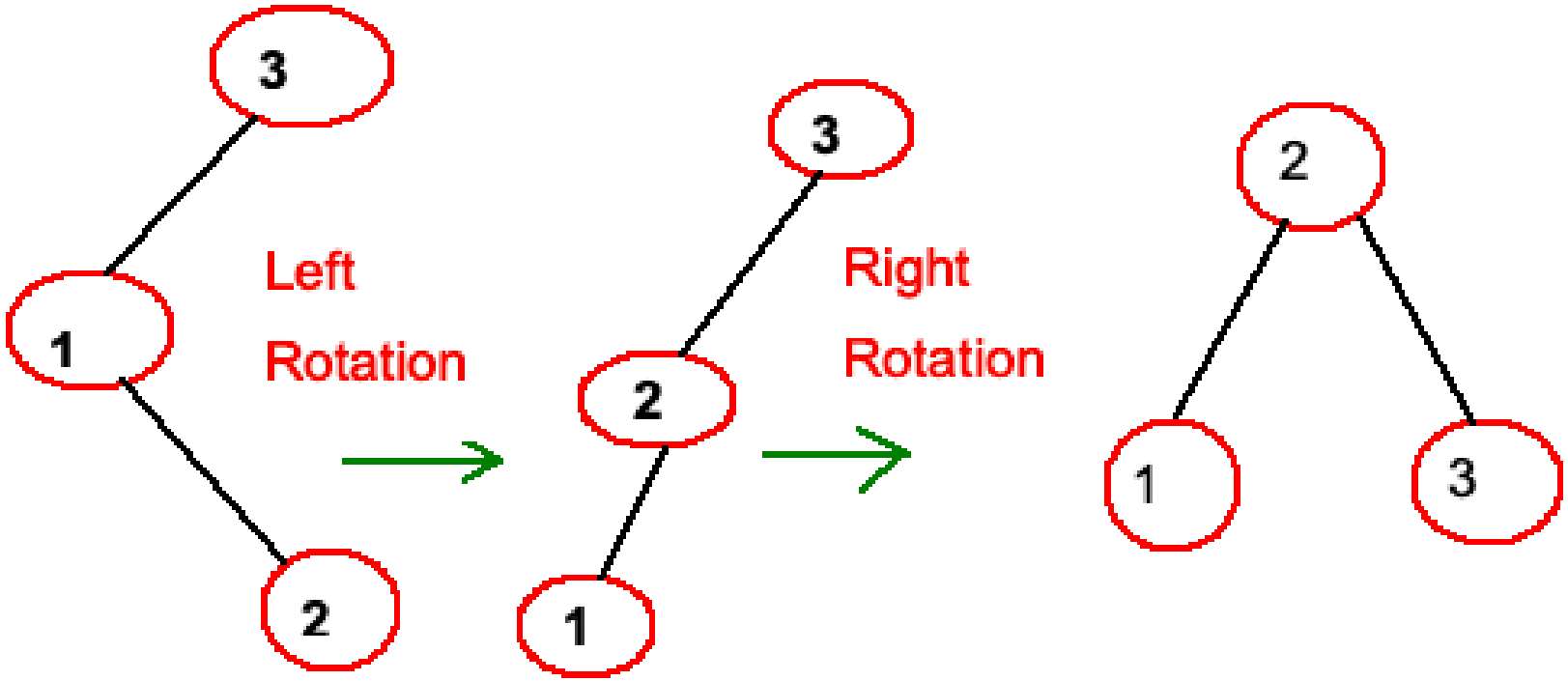

Figure 6: AVL Tree: Left-right rotation

Rotations: Left

When we do rotations, focus on 2 nodes, x and y

Insert 1,2,3

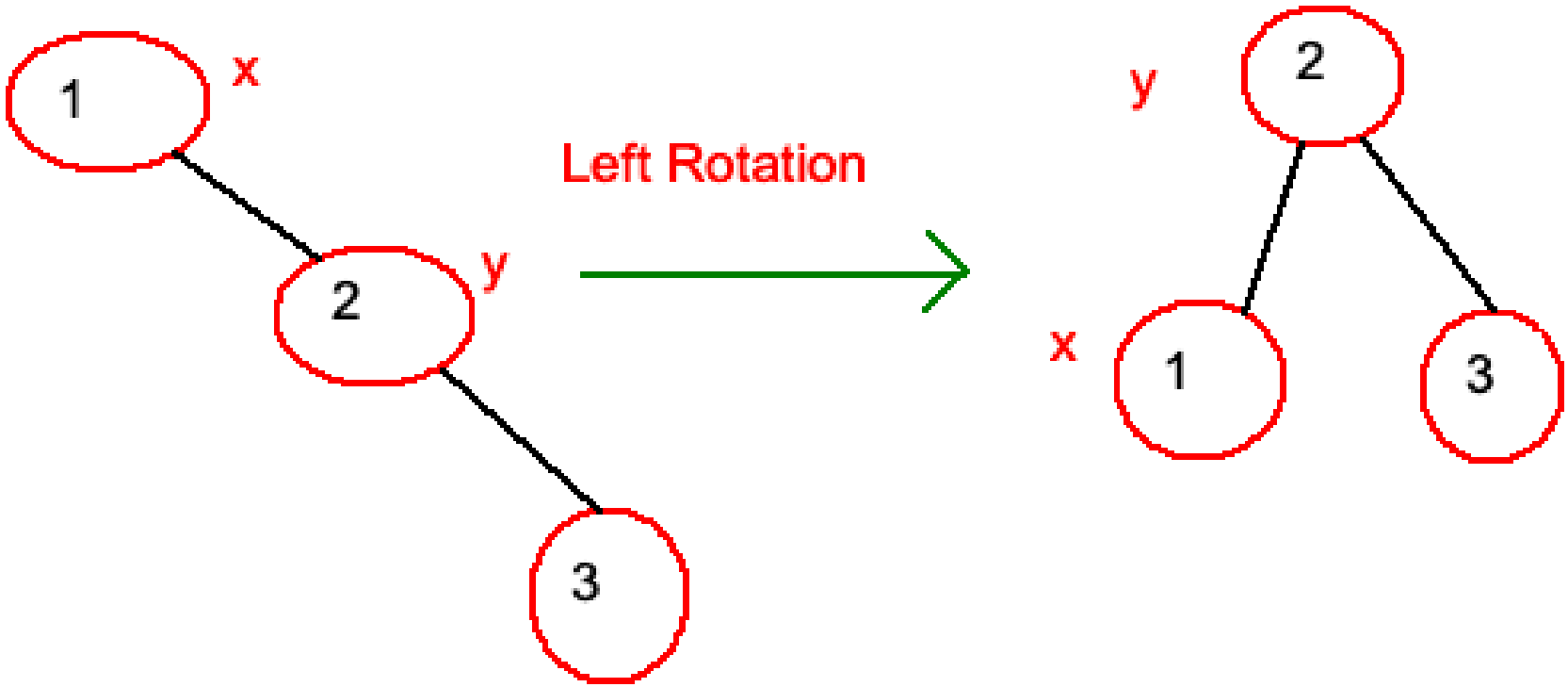

Figure 7: AVL: Left rotation

Rotations: Right-Left

When we do rotations, focus on 2 nodes, x and y

Insert 1,3,2

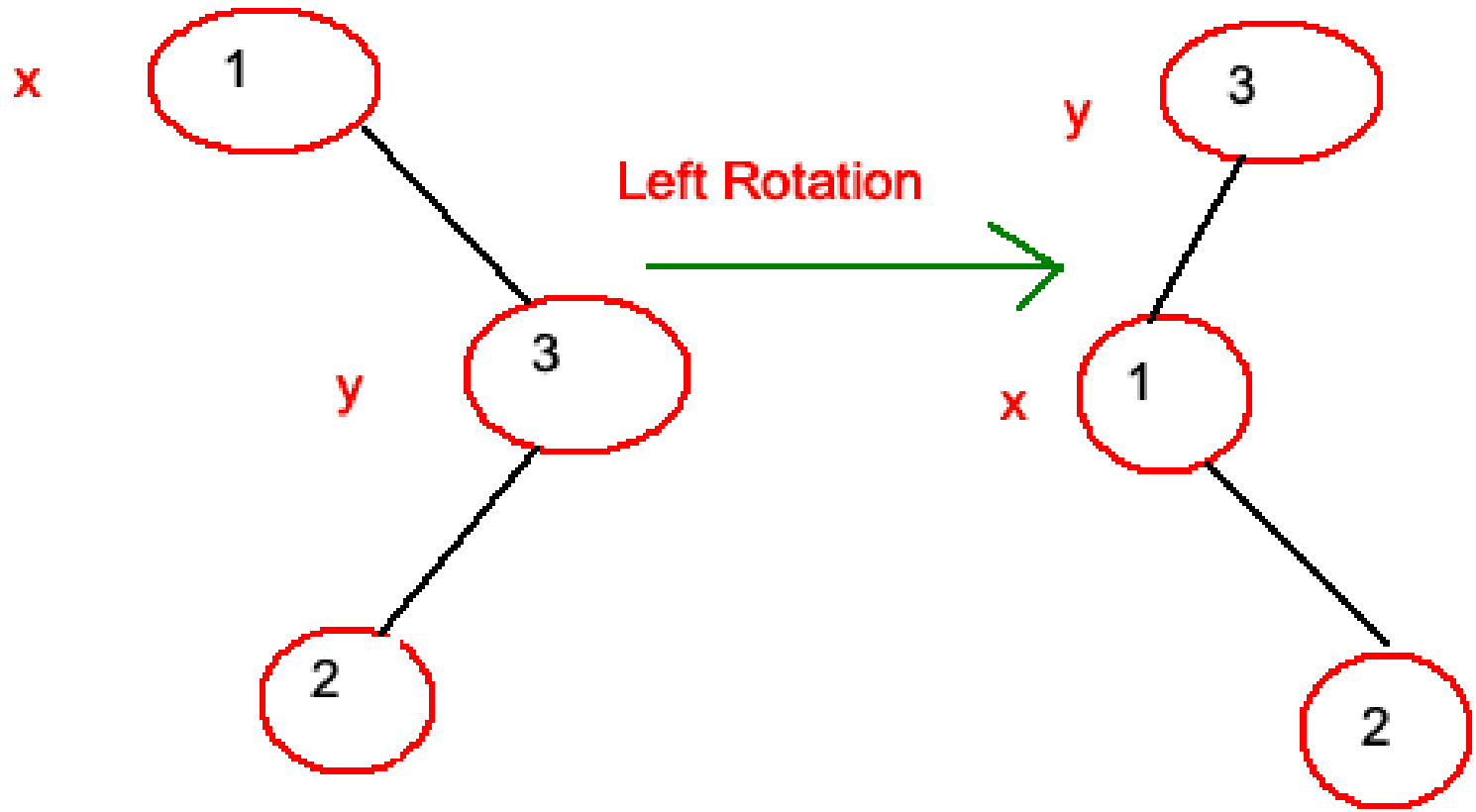

Figure 8: Left rotation:AVL

Rotations: Right-Left

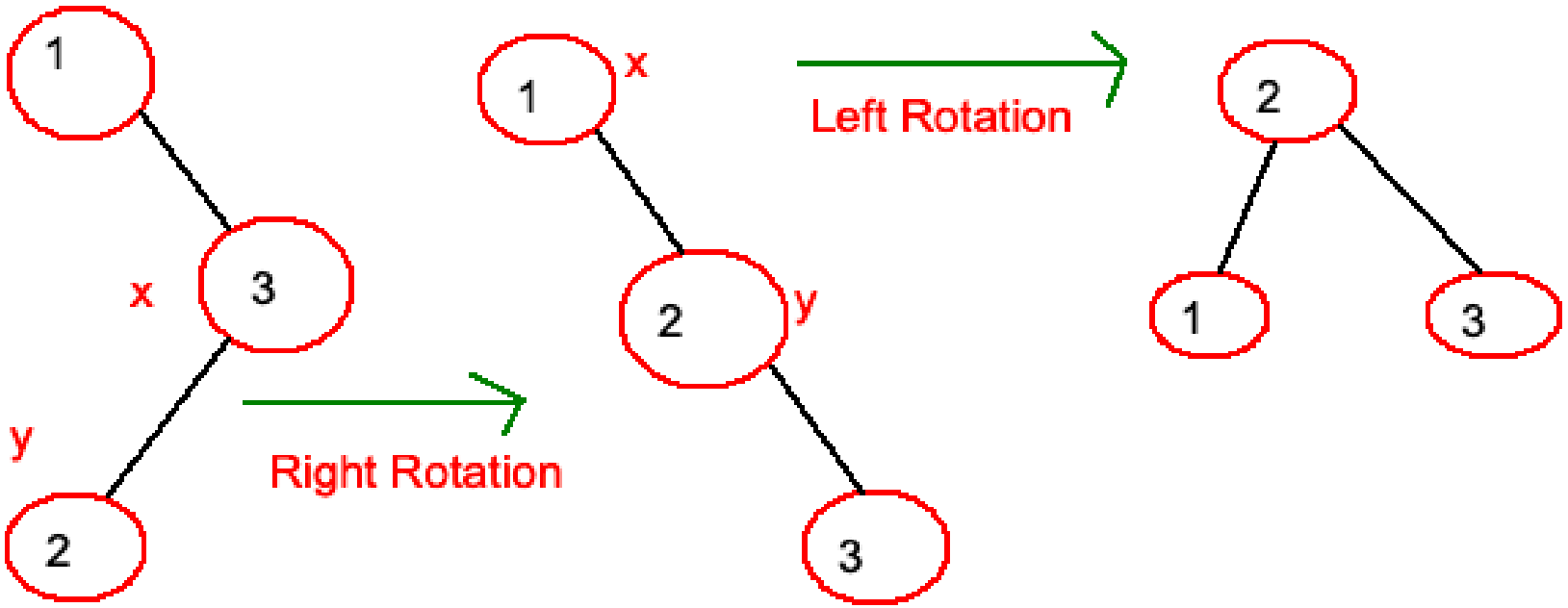

Figure 9: AVL Tree: Right-left Rotation

Balanced Search Trees: 2-3 search trees

Allow 1 or 2 keys per node.

- 2-node: one key, two children.
- 3-node: two keys, three children.

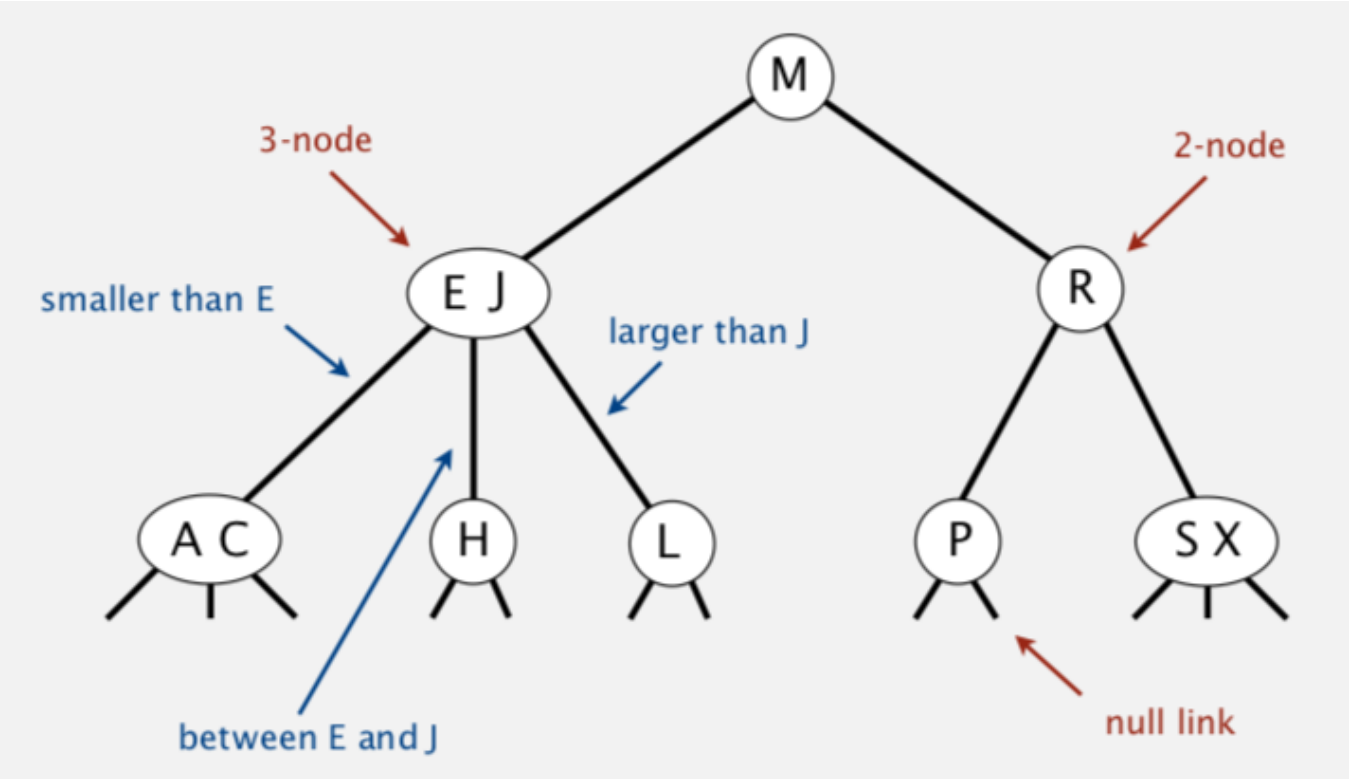

Figure 10: 2-3 search representation

Balanced Search Trees: red-black BSTs

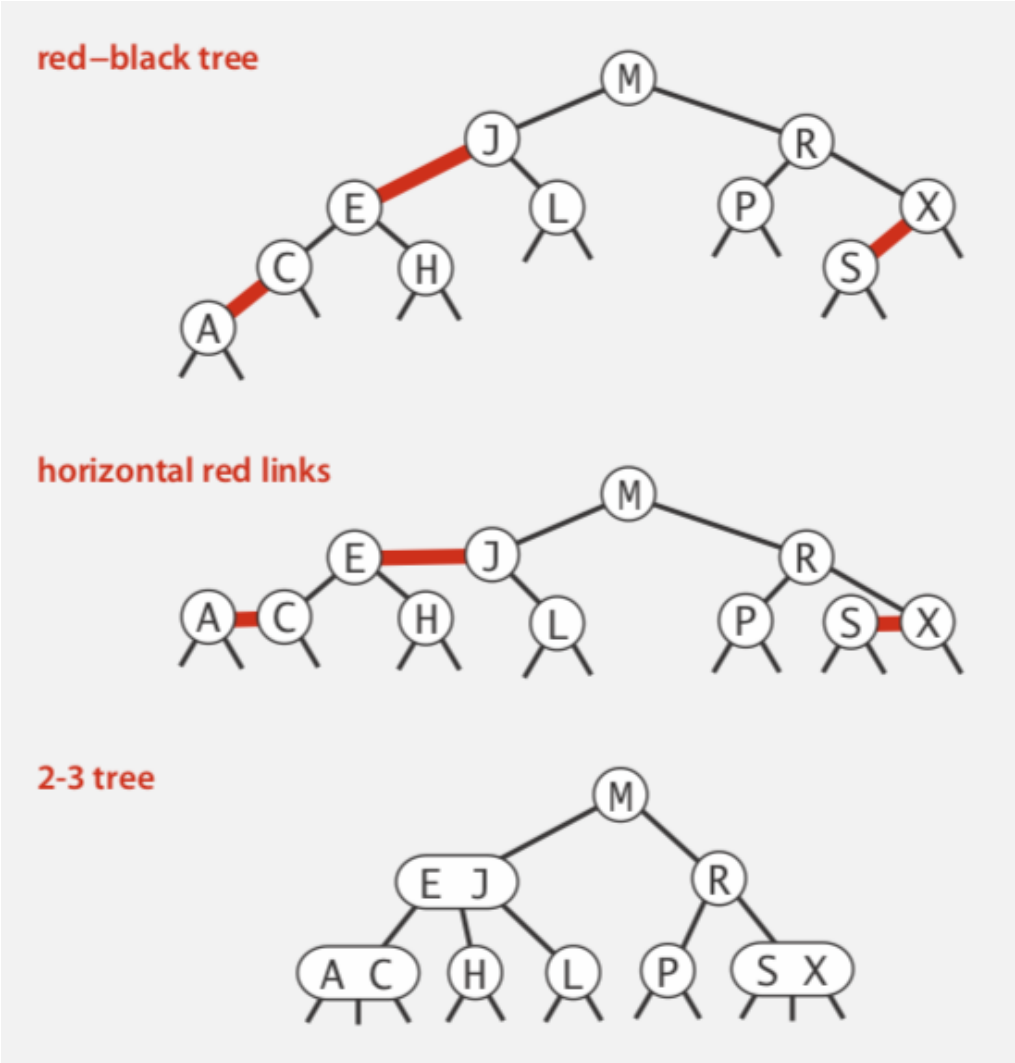

Figure 11: Red-Black Tree

Balanced Search Trees: B-trees(Bayer-McCreight, 1972)

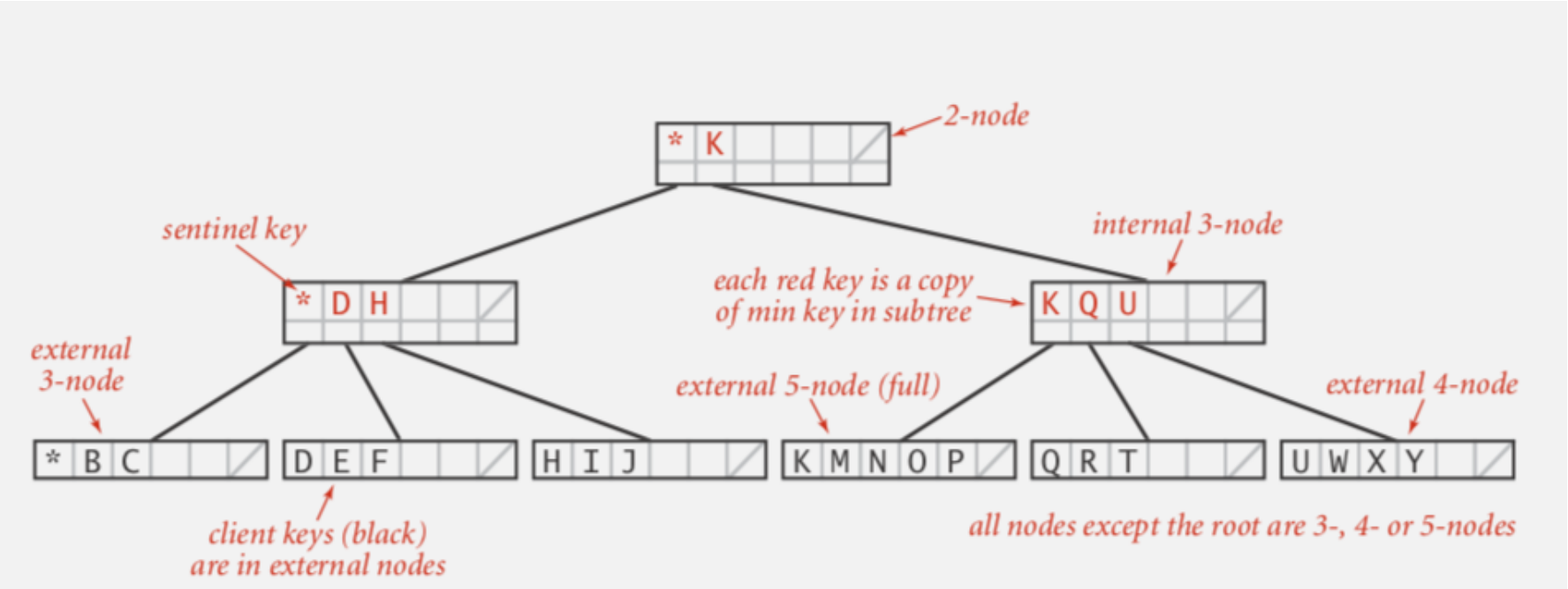

Figure 12: B tree

Balanced Search Trees: Performance

Bottom line. Guaranteed logarithmic performance for search and insert. O(Log n)