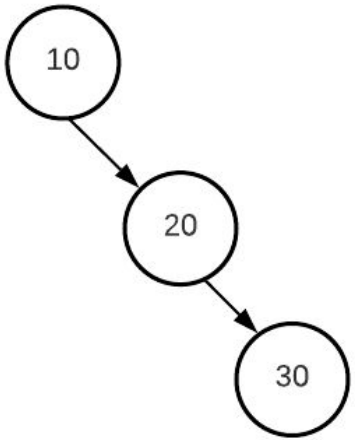
Data Structures

Imbalance Types

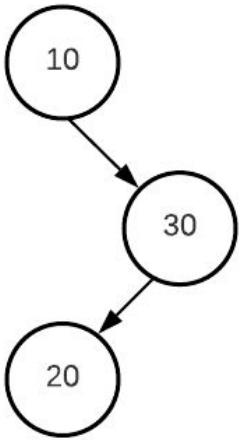
Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

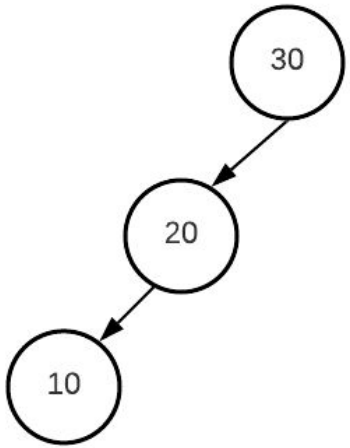
4 imbalance cases (|BF| > 1)



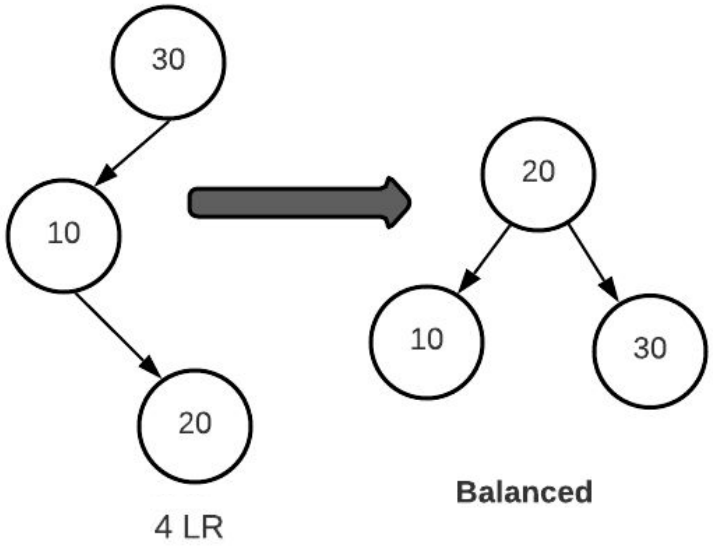
1 RR



2 RL



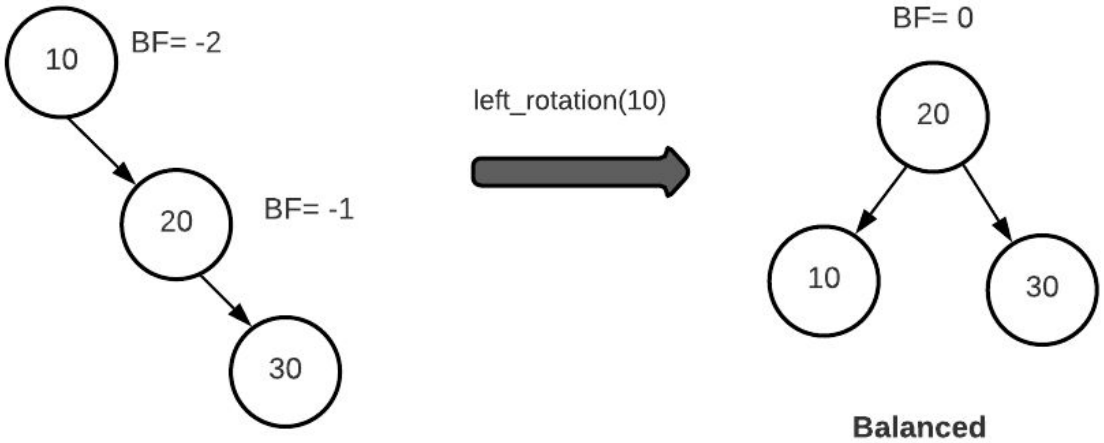
3 LL



4 LR

Case 1: Right-Right imbalanced tree

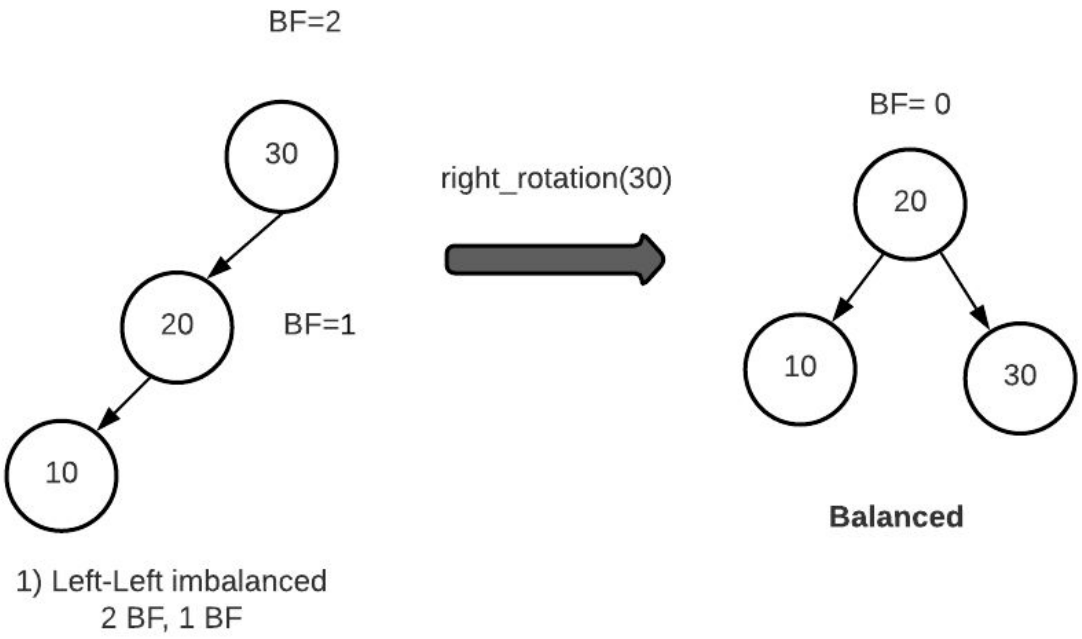
- Happens when we a node is eventually inserted on right side
  - BF are -2, -1
- Apply left rotation on root
- node = left_rotation(node);
- Note for simplicity we dropped subtrees for nodes 10, 20, 30



1) Right Right unbalanced
-2 BF, -1 BF

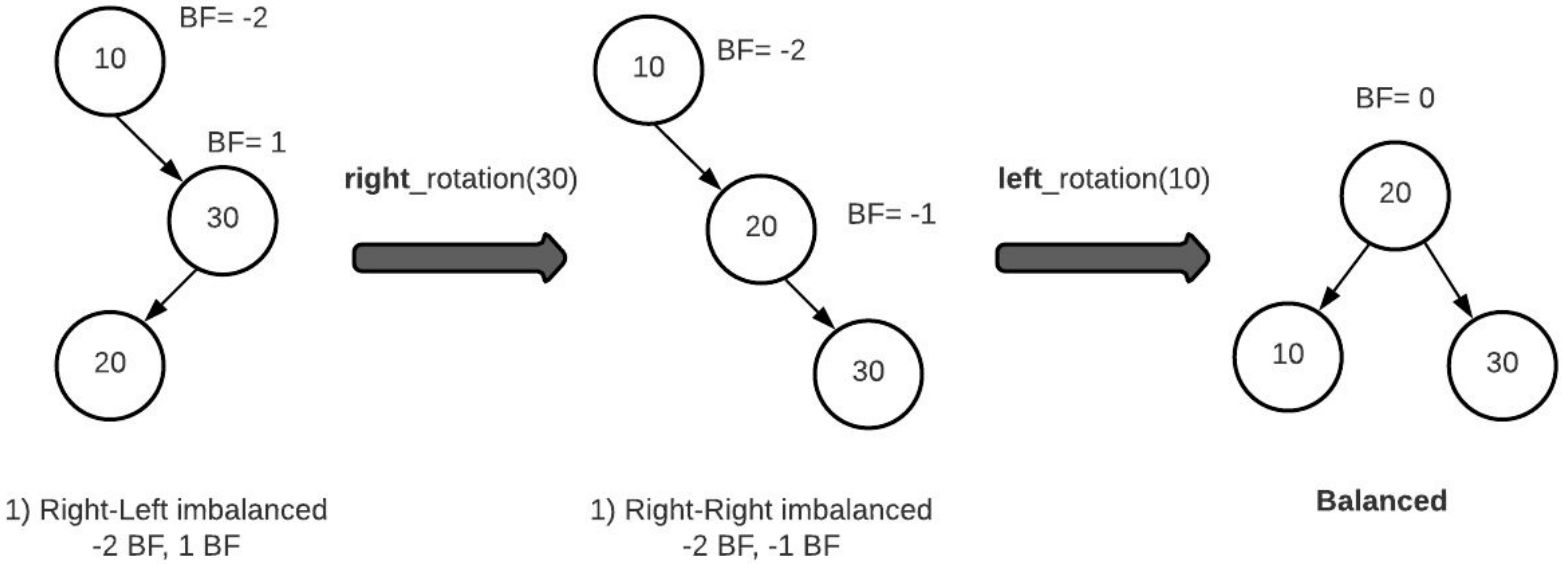
Case 2: Left-Left imbalanced tree

- The complete opposite case
  - BF: 2 1
- Do right rotation on root
- node = right_rotation(node);



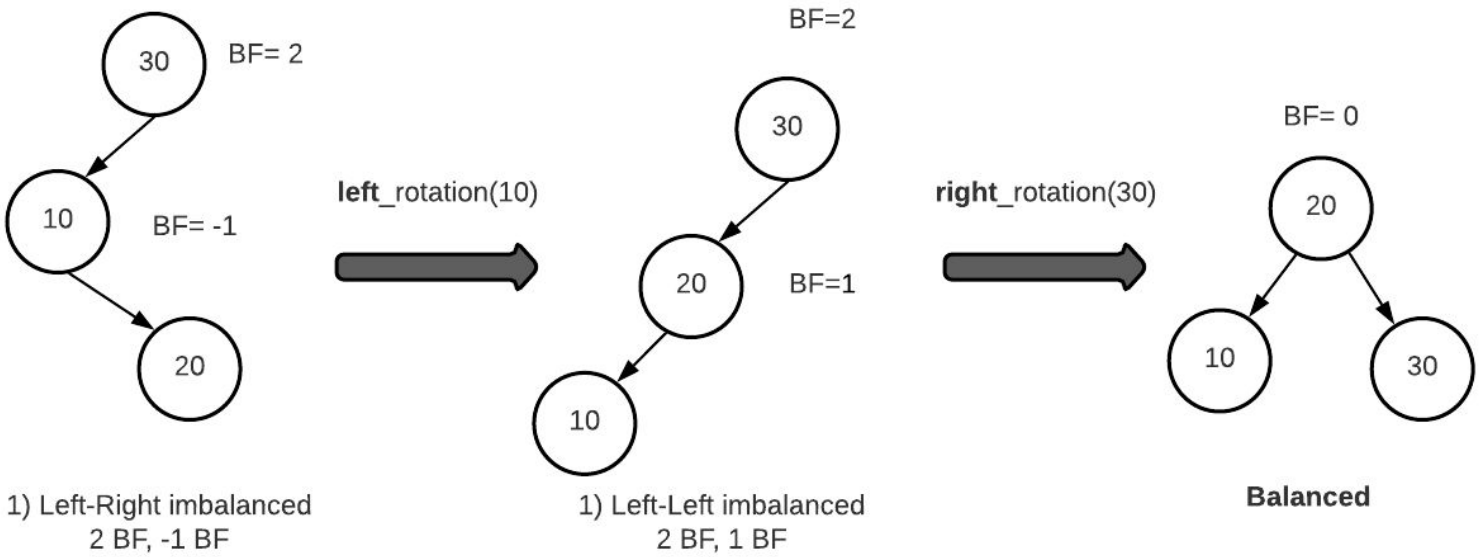
Case 3: Right-Left imbalanced tree

- The trick here to convert it first to right-right imbalance tree case
- node->right = right_rotation(node->right);
- node = left_rotation(node);



Case 4: Left-Right imbalanced tree

- The trick here to convert it first to left-left imbalance tree case
- node->left = left_rotation(node->left);
- node = right_rotation(node);



4 cases handling

```
AVLTree* balance(AVLTree* node) {
    if (node->balance_factor() == 2) {           // Left
        if (node->left->balance_factor() == -1) // Left Right?
            node->left = left_rotation(node->left); // To Left Left

        node = right_rotation(node);    // Balance Left Left
    } else if (node->balance_factor() == -2) {
        if (node->right->balance_factor() == 1)
            node->right = right_rotation(node->right);

        node = left_rotation(node);
    }
    return node;
}
```

Notes

- All introduced functions are O(1)
- In a general BST, balance factor can be any number!
- But balance() function assumes only: {-2, -1, 0, 1, 2}
- **Why?**
- Because AVL follows a **change-then-fix** approach
- The tree will always be balanced {-1, 0, 1}
- Once we make an insert/delete ⇒ may be **1 step** imbalanced up or down
  - That is: {-2, -1, 0, 1, 2}
- Then we fix immediately in bottom-up style any corruption ⇒ {-1, 0, 1}

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."