[Week 12] Practice

AVL Tree

[Code] advanced/main.py

Advanced

- Rebalace 함수 완성
- merge_avl 함수 완성 (AVL트리 병합)

```python
# --- 심화문제 함수1 ---
'''
    [1] rebalance() 함수를 완성하라.
'''
def rebalance(self, node):
    self.update_height(node)
    balance = self.balance_factor(node)
    if balance > 1:
        if self.balance_factor(______) < 0:
            ______ = ____________
        return ________________________
    if balance < -1:
        if self.balance_factor(______) > 0:
            ______ = ____________
        return ________________________
    return node
```

```python
# --- 심화문제 함수2 ---
'''
    [2] merge_avl() 함수를 완성하라.
'''
def merge_avl(tree1: AVLTree, tree2: AVLTree) -> AVLTree:
    if not tree1.root:
        return tree2
    if not tree2.root:
        return tree1

    node = tree1.root
    while node.right:
        node = node.right
    max_key = _____

    tree1.delete(_____)

    new_root = AVLNode(_____)
    new_root.left  = _____
    new_root.right = _____

    new_tree = AVLTree()
    new_tree.root = new_root
    new_tree.root = new_tree.rebalance(_____)

    return new_tree
```

Advanced

- split_avl 함수 완성
  (특정 key 기준 AVL트리 분할)

[Code] advanced/main.py

```python
# --- 심화문제 함수3 ---
'''
    [3] split_avl() 함수를 완성하라.
'''
def split_avl(tree: AVLTree, key: int) -> Tuple[AVLTree, AVLTree]:
    def _split(node: AVLNode, key: int):
        if not node:
            return None, None

        if node.key <= key:
            left_sub, right_sub = _split(_____, _____)
            node.right = _____
            node = tree.rebalance(node)
            return node, right_sub
        else:
            left_sub, right_sub = _split(_____, _____)
            node.left = _____
            node = tree.rebalance(node)
            return left_sub, node

    left_root, right_root = _split(_____, _____)
    left_tree, right_tree = AVLTree(), AVLTree()
    left_tree.root, right_tree.root = left_root, right_root
    return left_tree, right_tree
```

Problem Output

advanced

```
Preorder: 30 20 10 29 40 50
Inorder:  10 20 29 30 40 50
Postorder: 10 29 20 50 40 30
Leaf nodes count: 3
Merged Inorder:  10 20 29 30 40 50 60 70 80
Merged Preorder: 50 30 20 10 29 40 70 60 80
Merged postorder: 10 29 20 40 30 60 80 70 50
Merged leaf count: 5
spl1 Inorder:  10 20 29 30 40
spl2 Inorder:  50 60 70 80
```