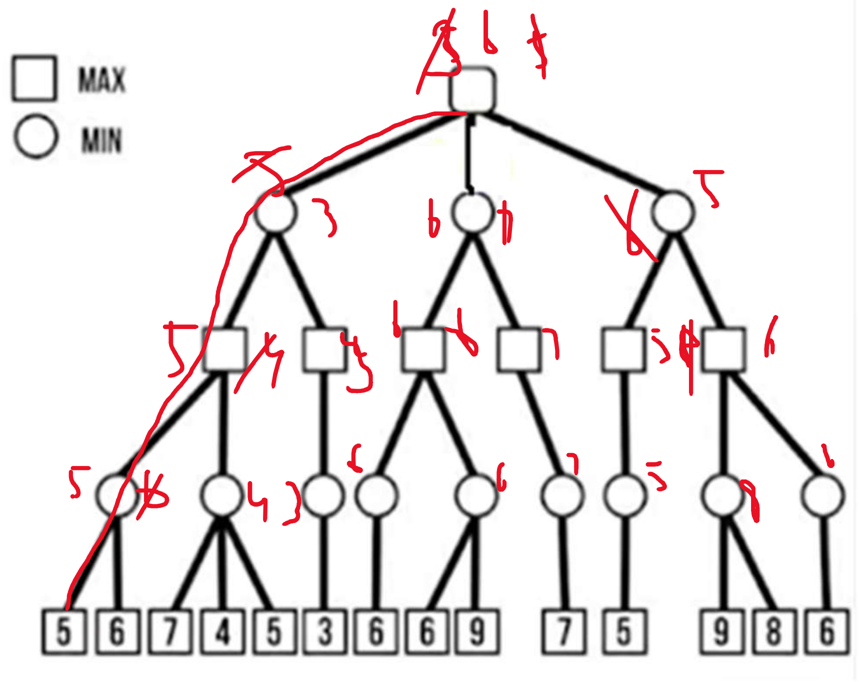
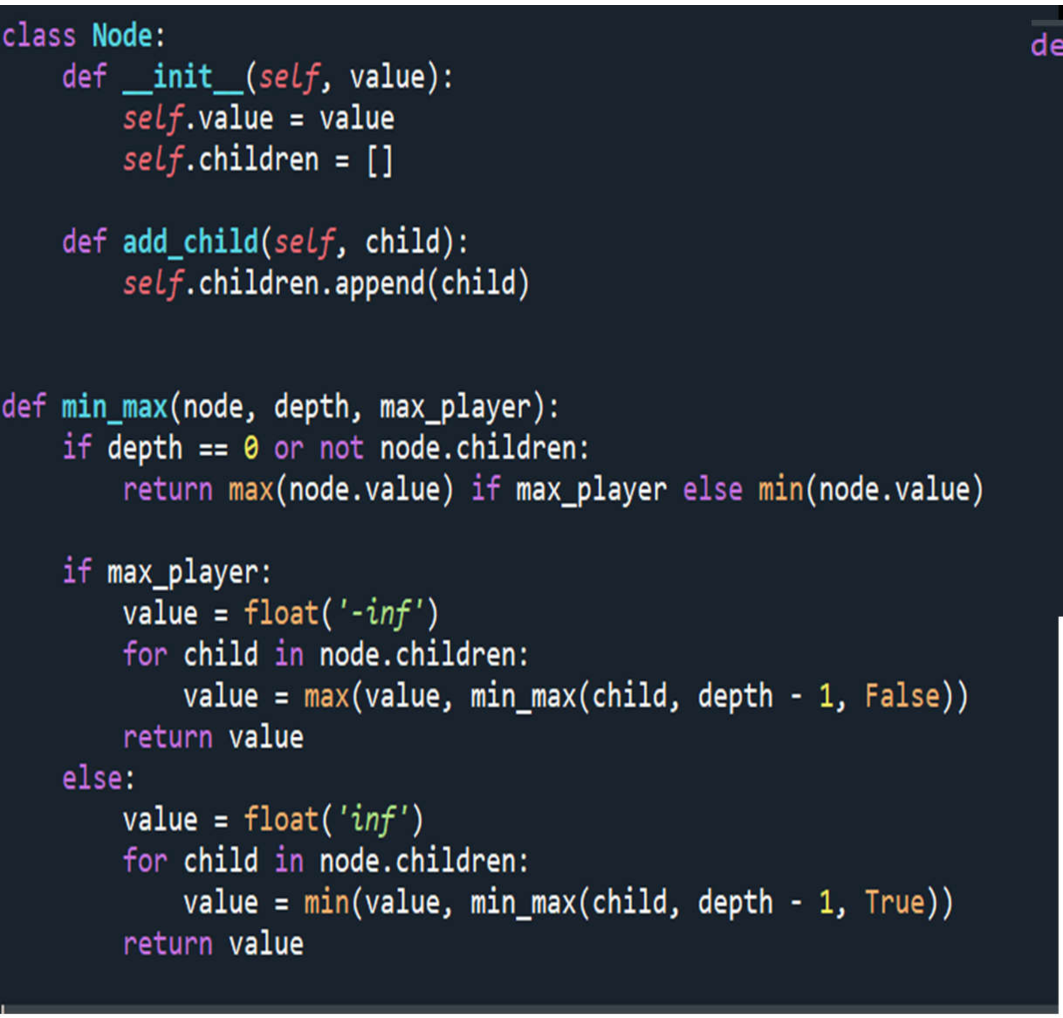
AI LAB WEEK 7

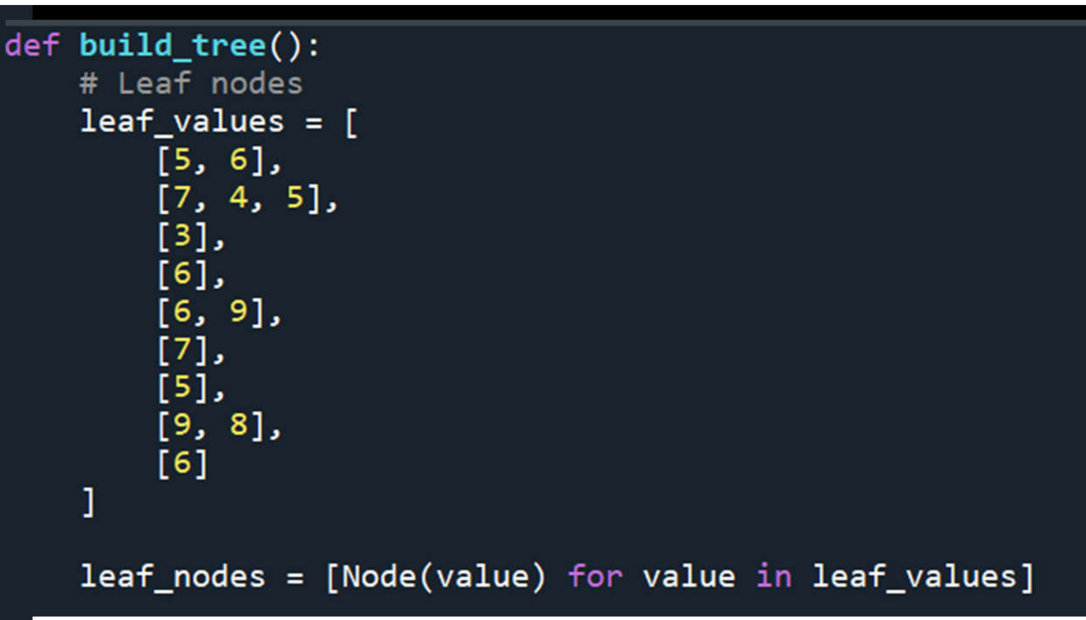
Dr. Mubashir Ahmad



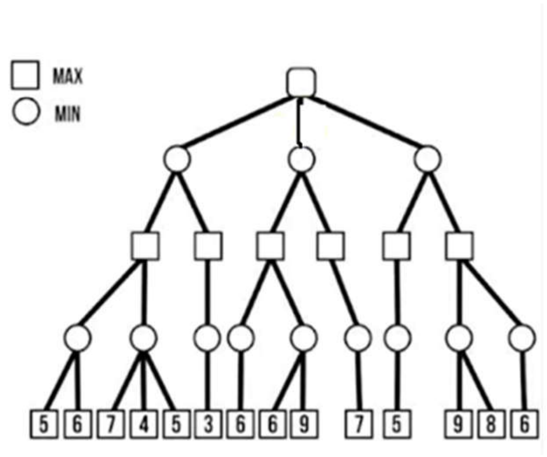
MIN MAX PROBLEM

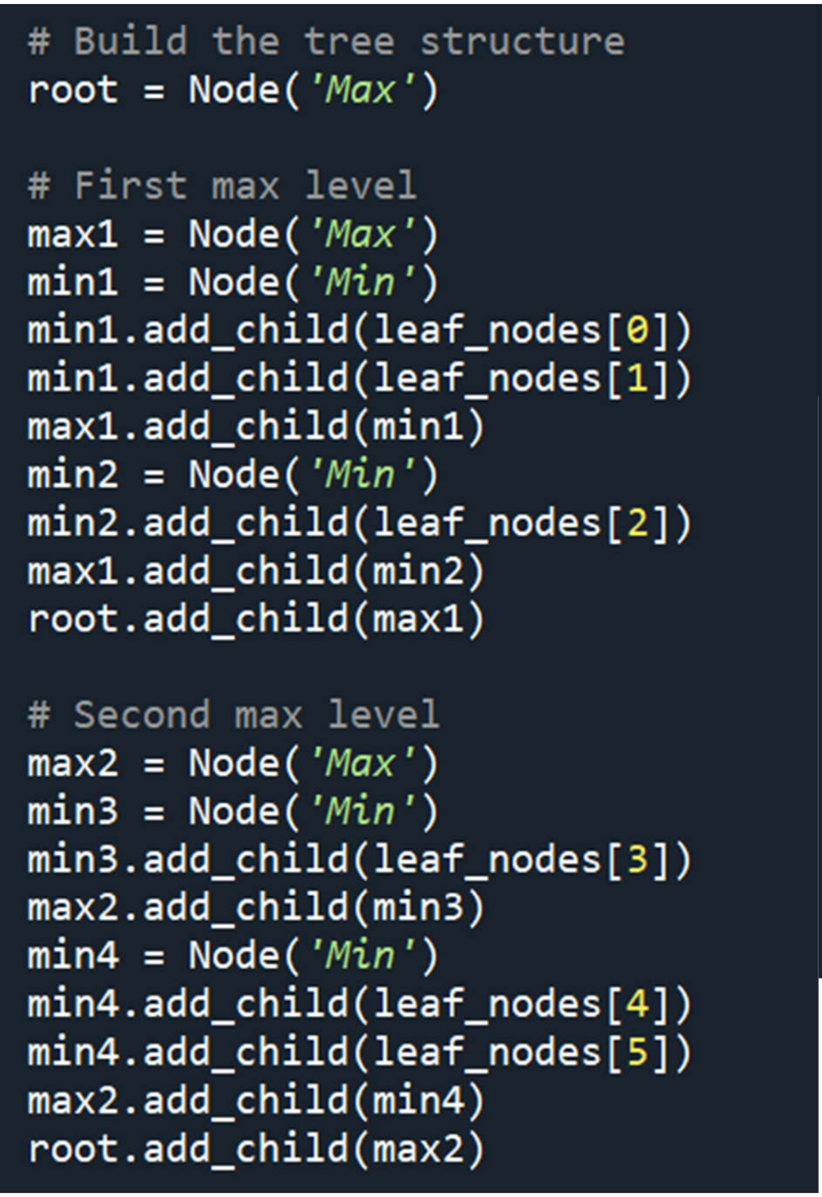


```python
class Node:
    def __init__(self, value):
        self.value = value
        self.children = []

    def add_child(self, child):
        self.children.append(child)


def min_max(node, depth, max_player):
    if depth == 0 or not node.children:
        return max(node.value) if max_player else min(node.value)

    if max_player:
        value = float('-inf')
        for child in node.children:
            value = max(value, min_max(child, depth - 1, False))
        return value
    else:
        value = float('inf')
        for child in node.children:
            value = min(value, min_max(child, depth - 1, True))
        return value
```

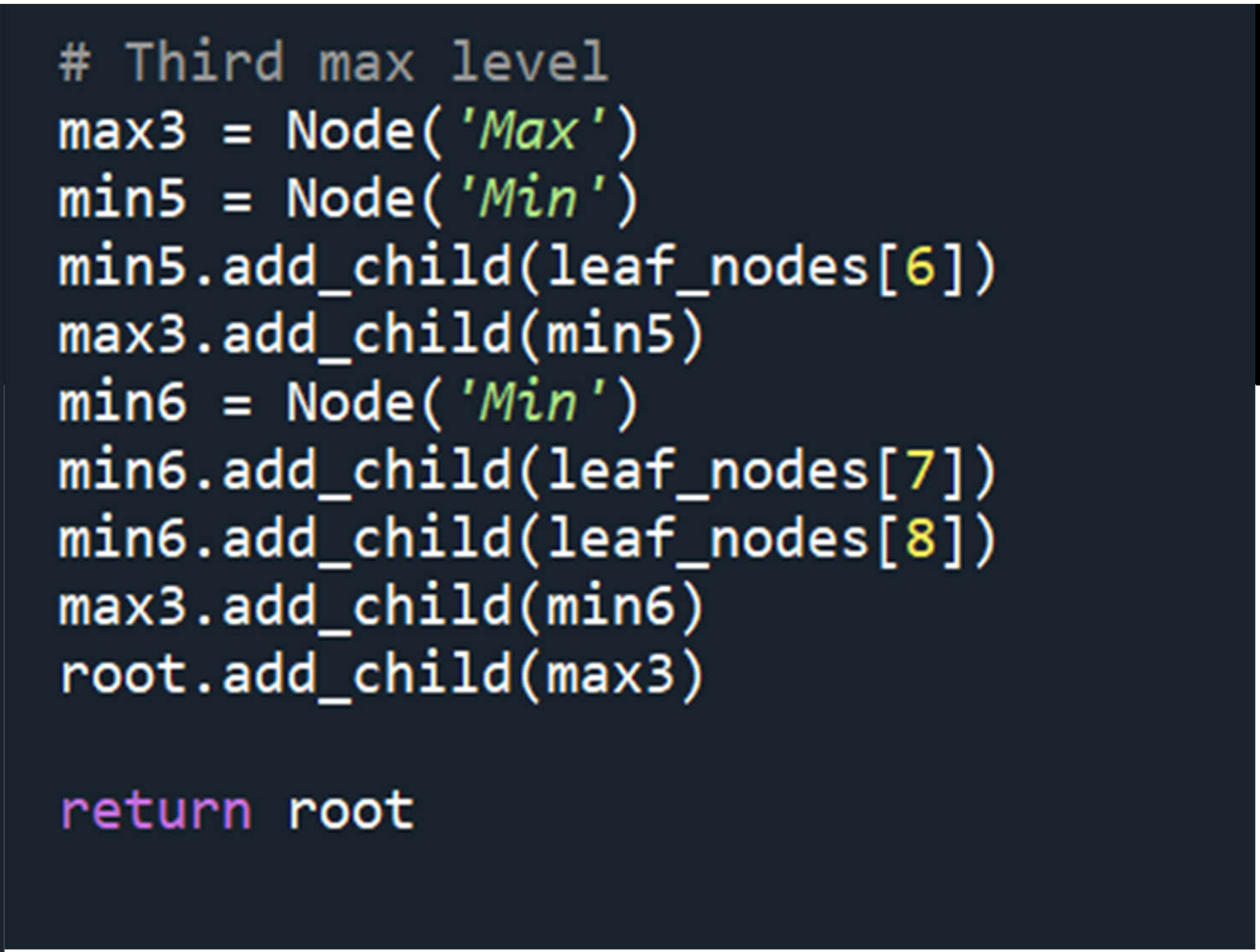


```python
def build_tree():
    # Leaf nodes
    leaf_values = [
        [5, 6],
        [7, 4, 5],
        [3],
        [6],
        [6, 9],
        [7],
        [5],
        [9, 8],
        [6]
    ]

    leaf_nodes = [Node(value) for value in leaf_values]
```

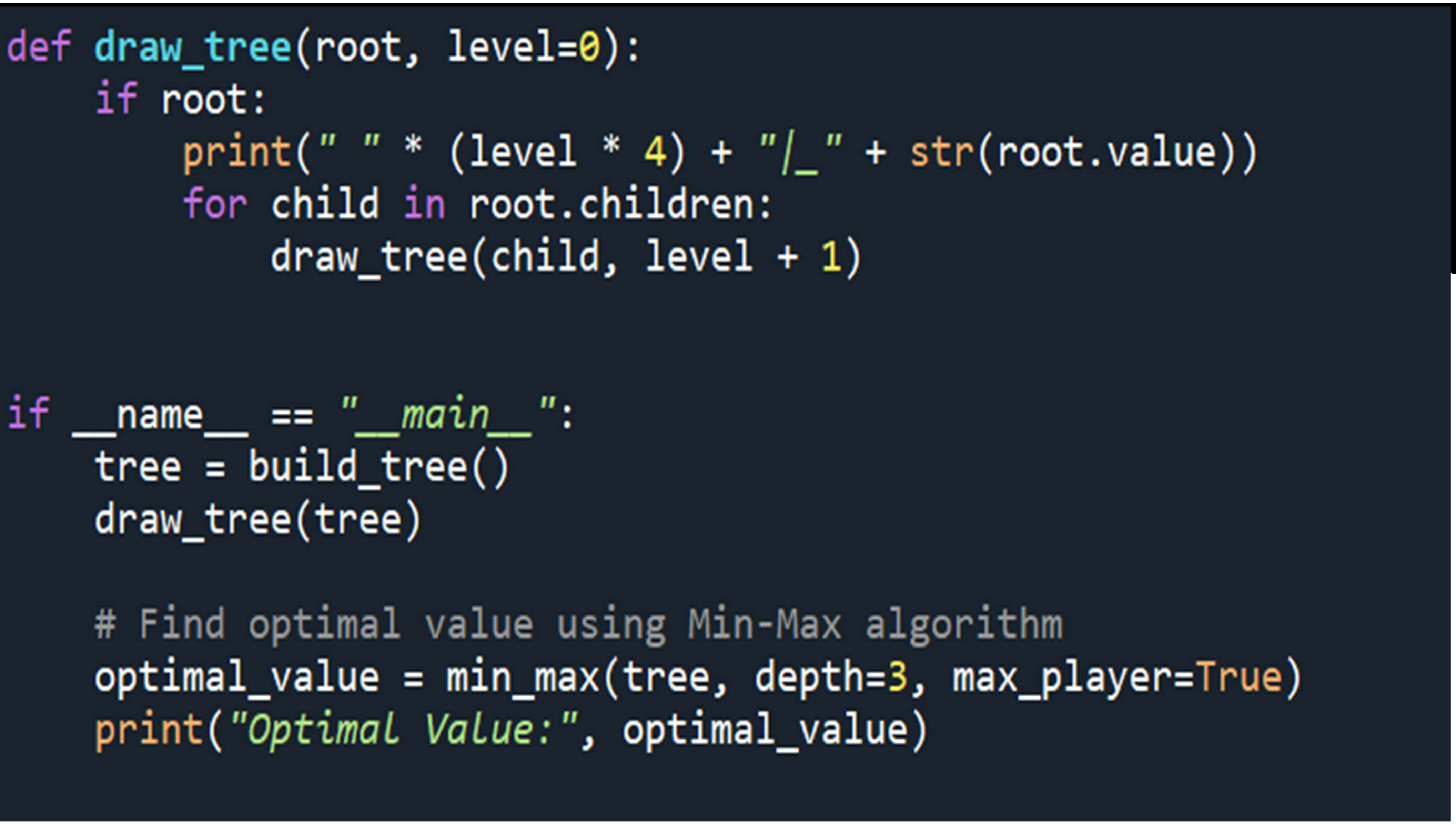




```python
# Build the tree structure
root = Node('Max')

# First max level
max1 = Node('Max')
min1 = Node('Min')
min1.add_child(leaf_nodes[0])
min1.add_child(leaf_nodes[1])
max1.add_child(min1)
min2 = Node('Min')
min2.add_child(leaf_nodes[2])
max1.add_child(min2)
root.add_child(max1)

# Second max level
max2 = Node('Max')
min3 = Node('Min')
min3.add_child(leaf_nodes[3])
max2.add_child(min3)
min4 = Node('Min')
min4.add_child(leaf_nodes[4])
min4.add_child(leaf_nodes[5])
max2.add_child(min4)
root.add_child(max2)
```



```python
# Third max level
max3 = Node('Max')
min5 = Node('Min')
min5.add_child(leaf_nodes[6])
max3.add_child(min5)
min6 = Node('Min')
min6.add_child(leaf_nodes[7])
min6.add_child(leaf_nodes[8])
max3.add_child(min6)
root.add_child(max3)

return root
```



```python
def draw_tree(root, level=0):
    if root:
        print(" " * (level * 4) + "|_" + str(root.value))
        for child in root.children:
            draw_tree(child, level + 1)


if __name__ == "__main__":
    tree = build_tree()
    draw_tree(tree)

    # Find optimal value using Min-Max algorithm
    optimal_value = min_max(tree, depth=3, max_player=True)
    print("Optimal Value:", optimal_value)
```

TASKS

#1: Build leaf tree

#2: Build the tree structure

#3: Draw a tree

#4: Code for min-max

#5: Write an optimal value