49. Convert Sorted Array to Binary Search Tree

Given an integer array nums where the elements are sorted in ascending order, convert it

to a height-balanced binary search tree.

Code:

```
class TreeNode:
    def __init__(self, val=0, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right

def sortedArrayToBST(nums):
    if not nums:
        return None

    mid = len(nums) // 2
    root = TreeNode(nums[mid])
    root.left = sortedArrayToBST(nums[:mid])
    root.right = sortedArrayToBST(nums[mid+1:])

    return root
nums = [-10, -3, 0, 5, 9]
bst_root = sortedArrayToBST(nums)

def print_tree(root):
    if not root:
        return
    print_tree(root.left)
    print(root.val, end=' ')
    print_tree(root.right)
```

Output:

```
In-order traversal of the BST:
-10 -3 0 5 9
```

Time Complexity:

- T(n)= O(n)