

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. Example 1 Input: nums = [-10,-3,0,5,9] Output: [0,-3,9,-10,null,5]

PROGRAM:-

```
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

    def helper(left, right):
        if left > right:
            return None

        # Always choose the middle element to maintain balance
        mid = (left + right) // 2
        root = TreeNode(nums[mid])

        # Recursively form the left and right subtrees
        root.left = helper(left, mid - 1)
        root.right = helper(mid + 1, right)

        return root

    return helper(0, len(nums) - 1)

# Helper function to print the tree in level order (for verification)
def print_level_order(root):
    if not root:
        return []

    result, current_level = [], [root]
    while current_level:
        level_values = []
        next_level = []
        for node in current_level:
            if node:
                level_values.append(node.val)
                next_level.append(node.left)
                next_level.append(node.right)
            else:
                level_values.append(None)
        result.append(level_values)
        current_level = [child for child in next_level if child is not None]
```

```
return result
```

```
# Example usage:
```

```
nums = [-10, -3, 0, 5, 9]
```

```
tree_root = sortedArrayToBST(nums)
```

```
print(print_level_order(tree_root)) # Output: [[0], [-3, 9], [-10, None, 5]]
```

OUTPUT:-

```
[[0], [-10, 5], [-3, 9]]
```

```
=== Code Execution Successful ===
```

TIME COMPLEXITY:-O(n)