Program 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] Explanation: [0,-10,5,null,-3,null,9] is also accepted:

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
    mid = len(nums) // 2
    return TreeNode(nums[mid], sortedArrayToBST(nums[:mid]),
sortedArrayToBST(nums[mid+1:]))

# Example usage
def printLevelOrder(root):
    result, queue = [], [root]
    while queue:
        node = queue.pop(0)
        if node:
            result.append(node.val)
            queue.extend([node.left, node.right])
        else:
            result.append(None)
    while result and result[-1] is None:
        result.pop()
    return result

nums = [-10, -3, 0, 5, 9]
root = sortedArrayToBST(nums)
print(printLevelOrder(root))  # Output: [0, -3, 9, -10, None, 5]
```

Output::

```
"C:\Program Files\Python312\python.exe" "C:\Work Space\DAA COADS.PYTHON\program 49.py"
[0, -3, 9, -10, None, 5]

Process finished with exit code 0
```

Time complexity:
O(n)