Experiment 3

Student Name: Mohit Garg   **UID:22BCS50051**

Branch: BE+ME CSE   **Section/Group: NTPP_IOT-602(A)**

Semester:6TH   **Date of Performance:12/02/25**

Subject Name: AP LAB-2   **Subject Code:22CSP-351**

1. **Aim: Divide and Conquer**
   a)   **Max Subarray**
   b)   **Reverse Bit**
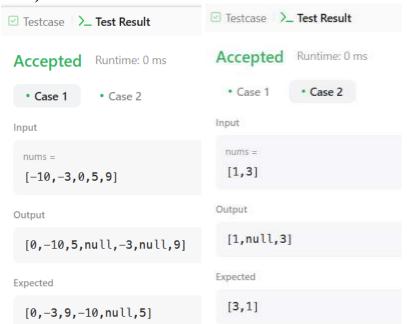   c)   **Number of 1 Bits**

2. **Code:**
   a)
```
class TreeNode {
   int val;
   TreeNode left;
   TreeNode right;
   TreeNode(int x) { val = x; }
}
public class Solution {
   public TreeNode sortedArrayToBST(int[] nums) {
      if (nums.length == 0) return null;
      return buildBST(nums, 0, nums.length - 1);
   }
   private TreeNode buildBST(int[] nums, int start, int end) {
      if (start > end) return null;
      int mid = start + (end - start) / 2; // Avoid potential overflow
      TreeNode root = new TreeNode(nums[mid]);
      root.left = buildBST(nums, start, mid - 1);
      root.right = buildBST(nums, mid + 1, end);
      return root;
   }
}
```

   b)
```
class Solution {
public:
```

```cpp
    bool isValidBST(TreeNode* root) {
        return valid(root, LONG_MIN, LONG_MAX);
    }
  private:
    bool valid(TreeNode* node, long minimum, long maximum) {
        if (!node) return true;
        if (!(node->val > minimum && node->val < maximum)) return false;
        return valid(node->left, minimum, node->val) && valid(node->right, node->val, maximum);
    }
};
```

c)
```cpp
class Solution {
public:
 int maxDepth(TreeNode* root) {
  if (root == nullptr)
    return 0;
  return 1 + max(maxDepth(root->left), maxDepth(root->right));
 }
};
```

3. **Output:**

a)

☑ Testcase >_ Test Result	☑ Testcase >_ Test Result
Accepted Runtime: 0 ms	**Accepted** Runtime: 0 ms
• Case 1 • Case 2	• Case 1 • Case 2
Input	Input
nums = [−10,−3,0,5,9]	nums = [1,3]
Output	Output
[0,−10,5,null,−3,null,9]	[1,null,3]
Expected	Expected
[0,−3,9,−10,null,5]	[3,1]

b)

Testcase >_ Test Result

Accepted Runtime: 0 ms

- Case 1   - Case 2

Input

root =
[2,1,3]

Output

true

Expected

true

Testcase >_ Test Result

Accepted Runtime: 0 ms

- Case 1   - Case 2

Input

root =
[5,1,4,null,null,3,6]

Output

false

Expected

false

---

c)

Testcase >_ Test Result

Accepted Runtime: 0 ms

- Case 1   - Case 2

Input

root =
[3,9,20,null,null,15,7]

Output

3

Expected

3

Testcase >_ Test Result

Accepted Runtime: 0 ms

- Case 1   - Case 2

Input

root =
[1,null,2]

Output

2

Expected

2