ASSIGNMENT

**NAME:** ABAH ONAH GABRIEL

**MATRIC NUMBER:** AMS25CMS0004

**COURSE CODE:** CMP 831

**COURSE TITLE:** ADVANCE DATA STRUCTURE AND ALGORITHMS

From the binary search tree above perform the following

1. In order
2. Post order
3. Pre order
4. Level order

Solution

In order

Going be the principle or method it is LRR

= 15,25,28,30,35,40,45,50,55,60,70

Post order

= 15,28,25,35,30,40,45,70,50,55,60

Pre order

Going by the formula

Root left right

= 40,15,28,30,25,35,50,70,45,60,55

Level order

We mirror from upper value to the bottom level

= 40,30,50,25,35,45,60,15,28,55,70

QUESTION TWO

Write a php or JavaScript program that implement level order of binary search

**Step 1: Build the BST**

We insert the values in the order:

40 → 30 → 50 → 25 → 35 → 45 → 60 → 15 → 28 → 55 → 70

**Step 2: Level Order Traversal (Breadth-First)**

Here’s the JavaScript code to build the BST and perform level order traversal:

javascript

class TreeNode {

constructor(value) {

this.value = value;

this.left = null;

this.right = null;

}

}

class BST {

constructor() {

this.root = null;

}

insert(value) {

const newNode = new TreeNode(value);

if (!this.root) {

this.root = newNode;

return;

}

let current = this.root;

while (true) {

if (value < current.value) {

if (!current.left) {

current.left = newNode;

return;

}

current = current.left;

} else {

if (!current.right) {

current.right = newNode;

return;

}

current = current.right;

}

}

}

levelOrderTraversal() {

const result = [];

const queue = [];

if (this.root) queue.push(this.root);

While (queue.length > 0) {

const current = queue.shift();

result.push(current.value);

if (current.left) queue.push(current.left);

if (current.right) queue.push(current.right);

}

return result;

}

}

// Build BST

const bst = new BST();

[40, 30, 50, 25, 35, 45, 60, 15, 28, 55, 70].forEach(val => bst.insert(val));

// Perform Level Order Traversal

console.log("Level Order Traversal:", bst.levelOrderTraversal());

**Output:**

This will log:

Level Order Traversal: [40, 30, 50, 25, 35, 45, 60, 15, 28, 55, 70]