Binary Tree Inorder Traversal (94)

Recursive solution is trivial.

CODE → T compl. → O(n), Space → O(log n) avg & O(n) worst.

```
Vector <int> v;
vector < int>  inorder (TreeNode* root) {
    if (root == NULL) {
        return v;
    }
    inorder (root → left);
    v. push_back (root → val);
    inorder (root → right);
    return v;
}
```

→ Iterative solution.

(i) Push the current node to stack. and set curr = curr → left until curr is NULL.

(ii) When curr == NULL and stack is not empty, pop the topmost node, print it's value and set curr = curr → right.

CODE

```
vector <int> v;
vector <int> inorder ( TreeNode* root)

    stack <TreeNode*> s;

    while (root != NULL || !s.empty()) {

        while (root != NULL) {

            s.push (root);
            root = root -> left;

        }

        root = s.top();
        s.pop();

        v.push_back (root -> val);
        root = root -> right;

    }

    return v;

}.
```

Time complexity → O(n)

space complexity → O(n).