```
// LNR: In-order (duyệt tăng dần)
void inorder(Node* root) {
    if (root) {
        inorder(root->left);
        cout << root->data << " ";
        inorder(root->right);
    }
}

// LRN: Post-order
void postorder(Node* root) {
    if (root) {
        postorder(root->left);
        postorder(root->right);
        cout << root->data << " ";
    }
}

int main() {
    Node* root = nullptr;
    root = insert(root, 10);
    insert(root, 5);
    insert(root, 15);
    insert(root, 2);
    insert(root, 7);

    cout << "Duyệt NLR: "; preorder(root); cout << endl;
    cout << "Duyệt LNR: "; inorder(root); cout << endl;
    cout << "Duyệt LRN: "; postorder(root); cout << endl;

    return 0;
}
```

---

3. CÂY CÂN BẰNG AVL

- **AVL Tree** là cây nhị phân tìm kiếm, nhưng luôn giữ cho chiều cao cân bằng.
- Mỗi node có `balance factor = height(left) - height(right)` ∈ {-1, 0, 1}

Các loại xoay:

- Xoay đơn trái / phải.
- Xoay kép trái phải / phải trái.

---

4. CÂY HEAP

- **Heap** là cây nhị phân hoàn chỉnh, thường dùng để xây dựng **Priority Queue**.
- **Min-Heap**: node cha ≤ node con.
- **Max-Heap**: node cha ≥ node con.

Min-Heap dùng mảng:

```
#include <iostream>
#include <vector>
using namespace std;
```