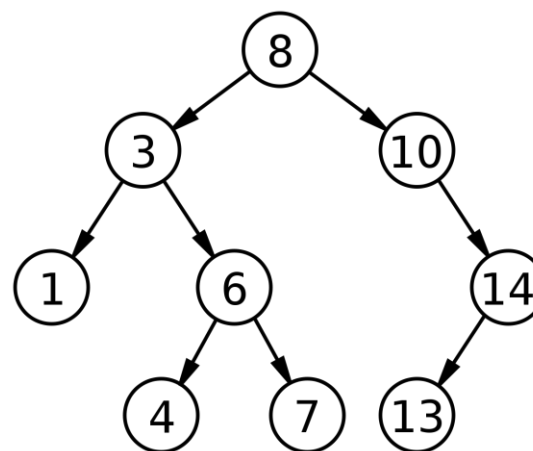


Bài 6.2: Cây nhị phân tìm kiếm

- ✓ Định nghĩa
- ✓ Các thao tác với cây nhị phân
- ✓ Tạo node
- ✓ Thêm node vào cây
- ✓ Duyệt node in-order
- ✓ Ví dụ minh họa & bài tập

Định nghĩa

- Cây nhị phân tìm kiếm(Binary Search Tree – BST) là cây trong đó các node của nó có các đặc điểm sau:
 - Giá trị các node của cây con bên trái nhỏ hơn giá trị của node cha của nó.
 - Giá trị các node của cây con bên phải lớn hơn hoặc bằng giá trị của node cha của nó.
 - Cây con bên trái và cây con bên phải cũng là cây nhị phân.



Các thao tác

- Tạo cây nhị phân tìm kiếm.
- Thêm một node vào cây nhị phân.
- Duyệt cây nhị phân: pre-order, in-order, post-order.
- Tìm kiếm một node trong cây nhị phân.
- Xóa một node khỏi cây nhị phân.
- Đếm số node lá của cây.
- Xác định độ sâu của một node.
- Tìm độ của cây...

Tạo node

Tạo class Node với 3 trường dữ liệu:

- left: con trỏ trỏ tới cây con bên trái.
- right: con trỏ trỏ tới cây con bên phải.
- Data: dữ liệu của node.

```
template<class T> class Node {  
public:  
    Node<T>* left; // con trỏ trỏ tới cây con bên trái  
    Node<T>* right; // con trỏ trỏ tới cây con bên phải  
    T data;        // dữ liệu của node  
  
    Node(T data) {  
        this->data = data;  
        this->left = nullptr;  
        this->right = nullptr;  
    }  
};
```

Tạo cây nhị phân

Tạo class BinarySearchTree với 1 con trỏ root:

```
template<class T> class BinarySearchTree {  
    Node<T>* root; // node gốc của cây  
public:  
    BinarySearchTree() { ... }  
    // hàm chèn thêm phần tử vào cây  
    void add(T value) { ... }  
    // hàm chèn thêm phần tử vào cây  
    Node<T>* add(Node<T>* r, T value) { ... }  
    // hàm duyệt cây  
    void inOrder() { ... }  
    // duyệt cây  
    void inOrder(Node<T>* r) { ... }  
};
```

Thêm node vào cây

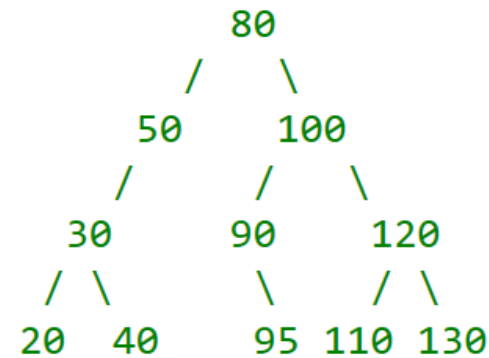
Giả sử cần chèn node p vào cây nhị phân:

- TH1: nếu `root == null`; gán `root = p`;
- TH2: nếu `root.data < p.data`; chèn p sang phải root;
- TH3: nếu `root.data > p.data`; chèn p sang trái root;

```
// hàm chèn thêm phần tử vào cây
void add(T value) {
    root = add(root, value);
}
// hàm chèn thêm phần tử vào cây
Node<T>* add(Node<T>* r, T value) {
    if (r == nullptr) { // nếu r null
        return new Node(value); // tạo node mới
    }
    if (value > r->data) { // thêm ở cây con phải
        r->right = add(r->right, value);
    }
    else { // thêm ở cây con trái
        r->left = add(r->left, value);
    }
    return r; // trả về node r sau khi thêm mới
}
```


Hiển thị các node

- Đoạn code sau hiển thị các phần tử theo thứ tự node nhỏ-> lớn:
hình ảnh cây nhị phân tìm kiếm trong ví dụ



```
// hàm duyệt cây
void inOrder() {
    inOrder(root);
}

// duyệt cây
void inOrder(Node<T>* r) {
    if (r != nullptr) {
        inOrder(r->left);
        cout << r->data << ' ';
        inOrder(r->right);
    }
}
```

Nội dung tiếp theo

Duyệt cây nhị phân tìm kiếm