Cấu trúc dữ liệu cây nhị phân (Binary Tree)

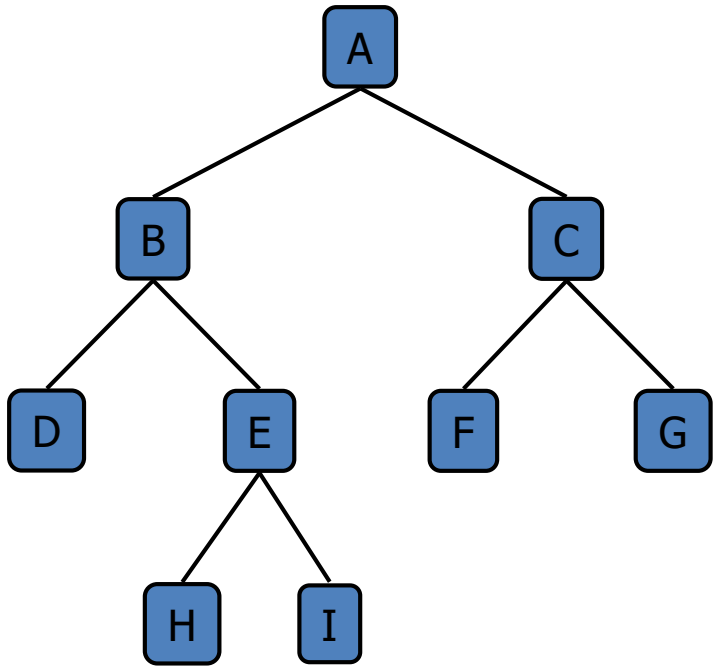
Nội dung

I. Cây nhị phân

II. Cấu trúc liên kết cho Cây nhị phân

III. Bài tập

I. Cây nhị phân (Binary tree)

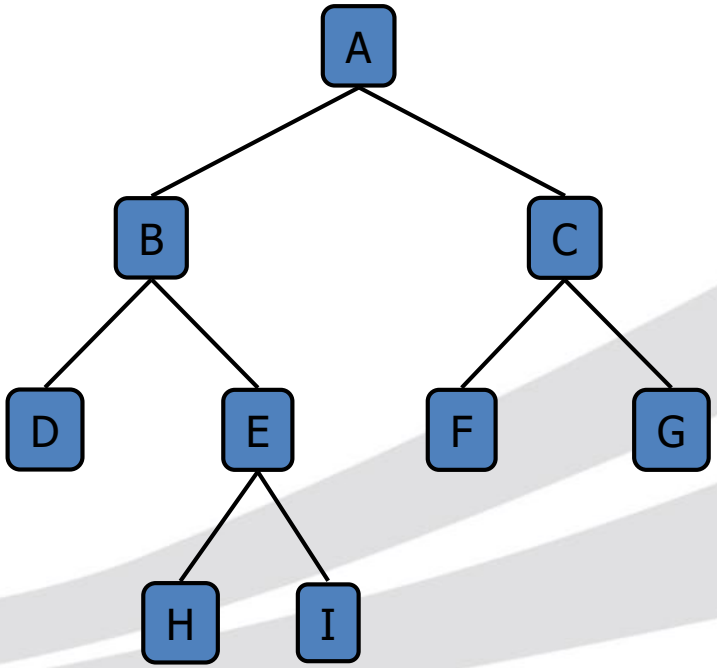


I. Cây nhị phân (Binary tree)

- **Cây nhị phân là một cây có các tính chất sau:**
  - Mỗi một nút trong có nhiều nhất 2 nút con
  - Các nút con của một nút là một cặp có thứ tự
- **Chúng ta gọi con của một nút trong là con trái và con phải**
- **Định nghĩa cây nhị phân bằng đệ qui:**

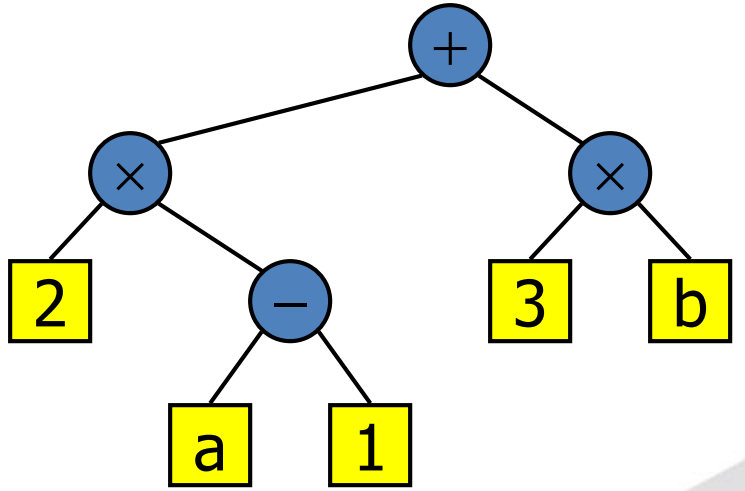
  Cây nhị phân là:
  - Một cây chỉ có một nút hoặc
  - Là cây mà nút gốc của nó có cặp nút con có thứ tự, mỗi một nút con là gốc của một cây nhị phân

- Ứng dụng:
  - Biểu diễn các biểu thức toán học
  - Quá trình quyết định
  - Tìm kiếm



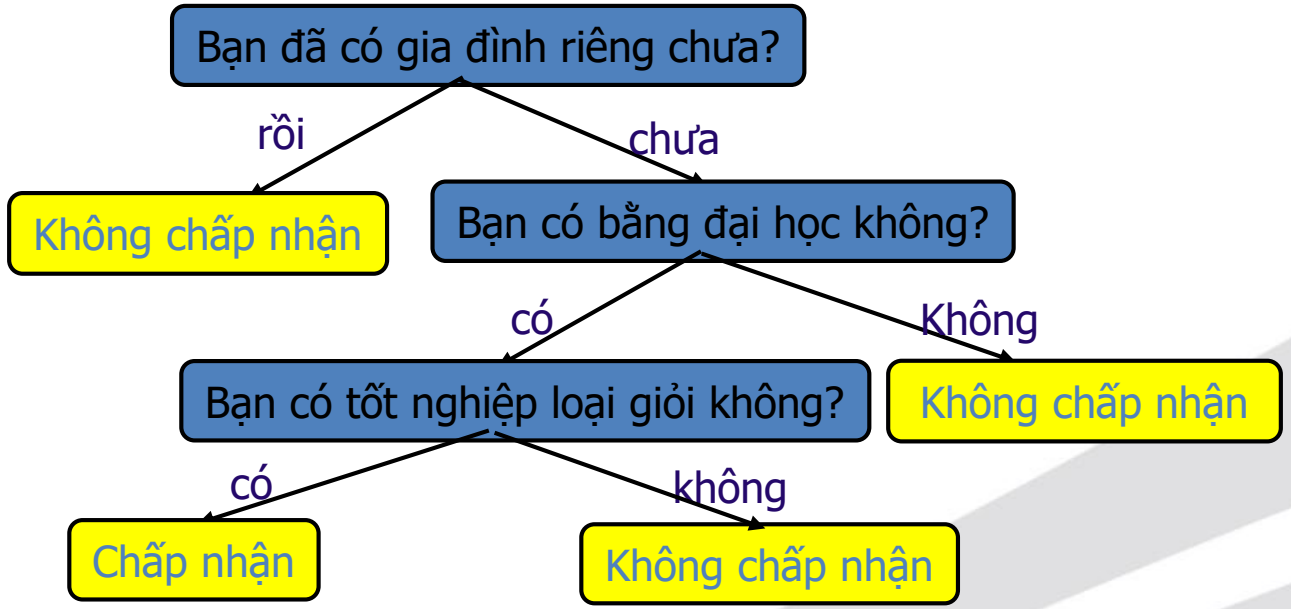
Cây biểu thức

- **Cây nhị phân biểu diễn một biểu thức toán học**
  - Các nút trong: là các toán tử (operators)
  - Các nút ngoài: các toán hạng (operands)
- **Ví dụ: Cây biểu thức cho biểu thức (2 × (a – 1) + (3 × b))**



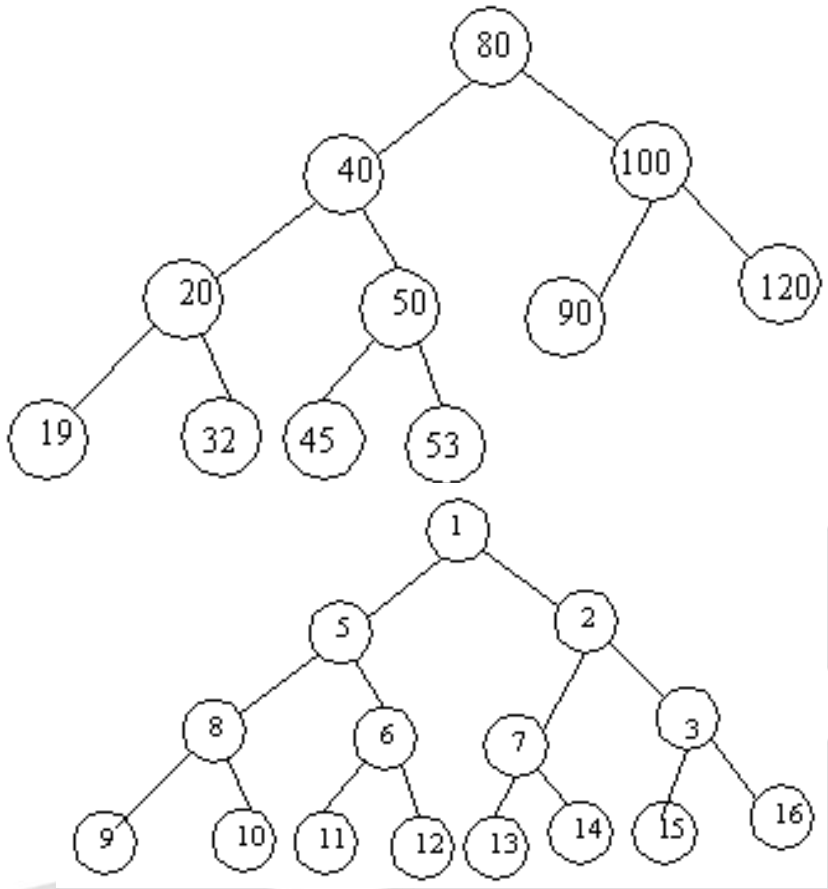
Cây quyết định (Decision tree)

- **Cây kết hợp với một quá trình quyết định**
  - Các nút trong: Các câu hỏi với câu trả lời **yes/no**
  - Các nút ngoài: các quyết định
- **Ví dụ: Cây quyết định tuyển nhân viên**



Một số định nghĩa

❖ Cây nhị phân hoàn chỉnh: Là cây nhị phân mà tất cả các nút trong của nó đều có đủ hai nút con



❖ **Cây nhị phân đầy đủ: là cây nhị phân hoàn chỉnh và tất cả các lá đều ở cùng mức**

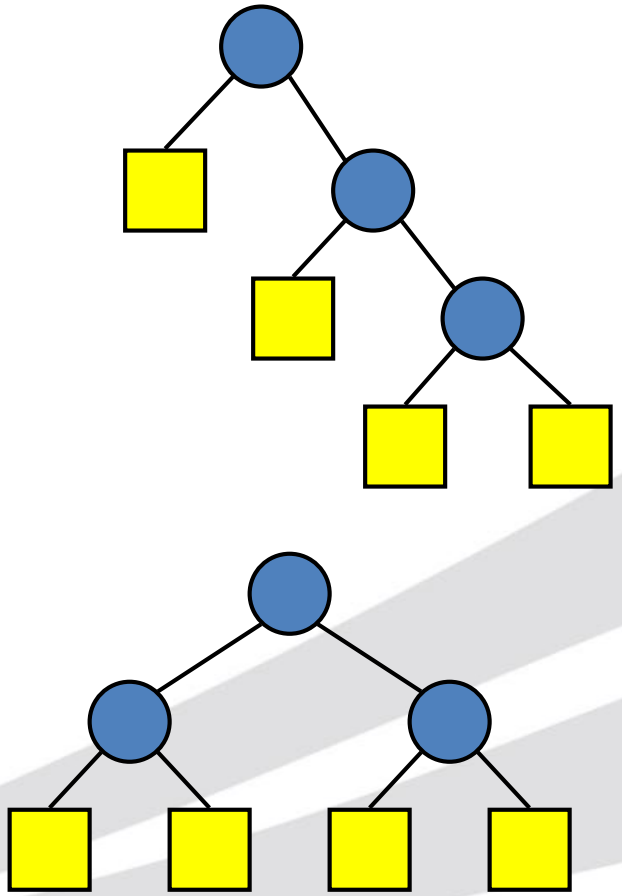
Các tính chất của cây nhị phân hoàn chỉnh

Ký hiệu

- n số các nút
- e số các nút ngoài
- i số các nút trong
- h chiều cao

Các tính chất:

- $e = i + 1$
- $n = 2e - 1$
- $h \le i$
- $h \le (n - 1)/2$
- $e \le 2^h$
- $h \ge \log_2 e$
- $h \ge \log_2 (n + 1) - 1$



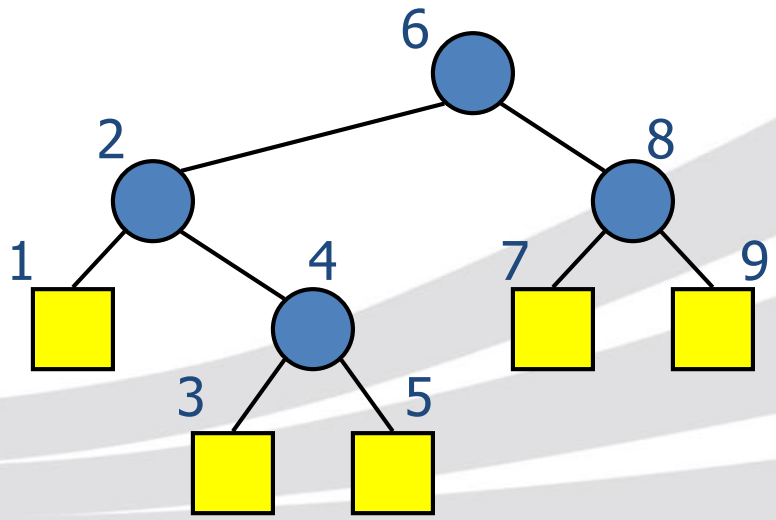
Cấu trúc dữ liệu Cây nhị phân (Binary tree ADT)

- **ADT cây nhị phân là sự mở rộng của ADT cây, tức là, nó kế thừa các phương thức của ADT cây**
- **Thêm vào các phương thức:**
  - Địa chỉ left(p) // trả lại địa chỉ của nút con trái
  - Địa chỉ right(p) // trả lại địa chỉ của nút con phải
  - int hasLeft(p) //Cho biết nút có con trái không
  - int hasRight(p) //Cho biết nút có con phải không

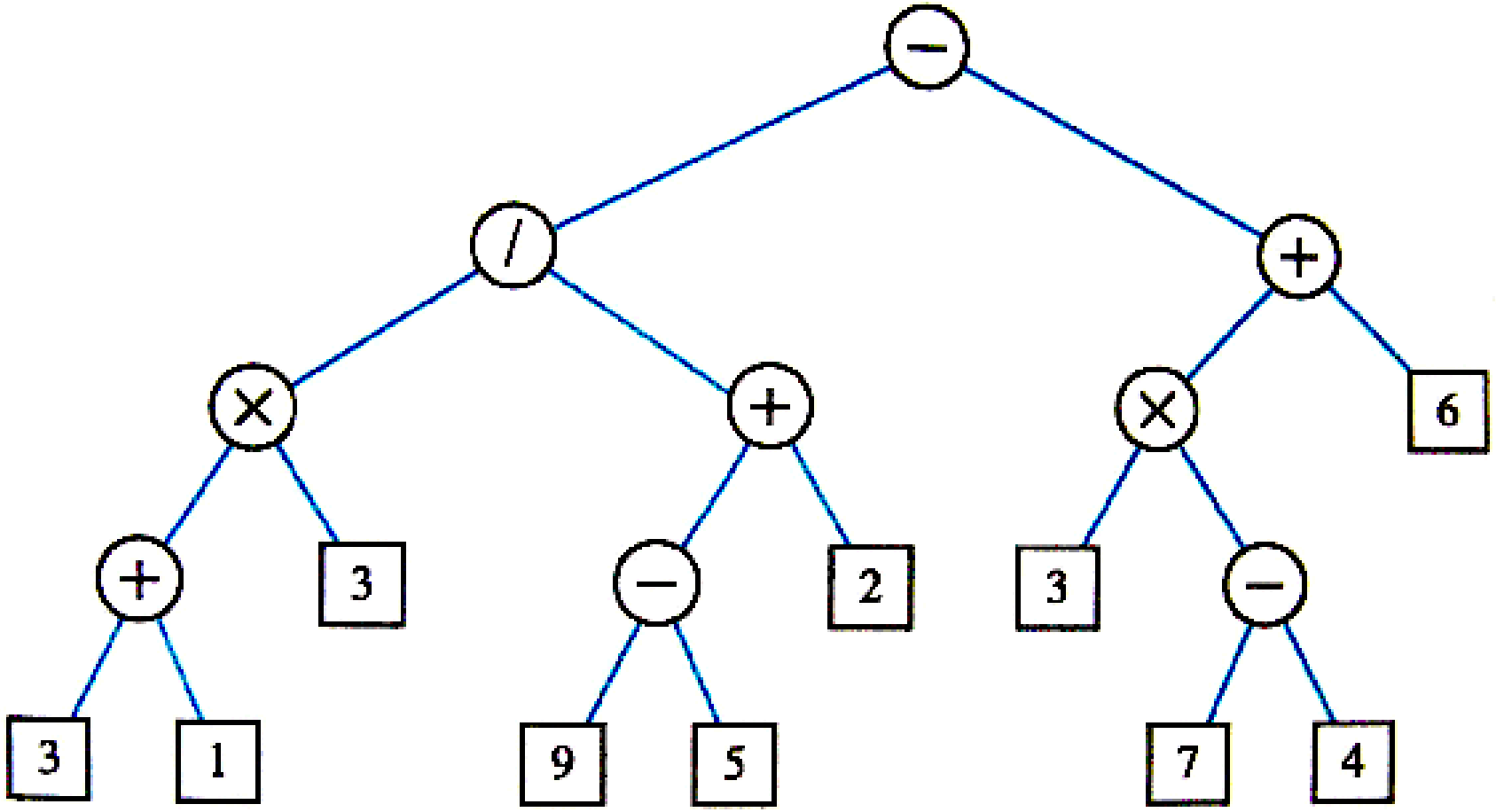
Duyệt theo thứ tự giữa - Inorder Traversal

- **Duyệt theo thứ tự giữa:**
  - Thăm cây con bên trái theo thứ tự giữa (nếu có)
  - Thăm nút cha
  - Thăm cây con bên phải theo thứ tự giữa (nếu có)
- **Ứng dụng: vẽ cây nhị phân**
  - x(v) = Thứ tự thăm của v
  - y(v) = độ sâu của v

```
Algorithm inOrder(v)
    if hasLeft (v)
        inOrder (left (v))
    visit(v)
    if hasRight (v)
        inOrder (right (v))
```

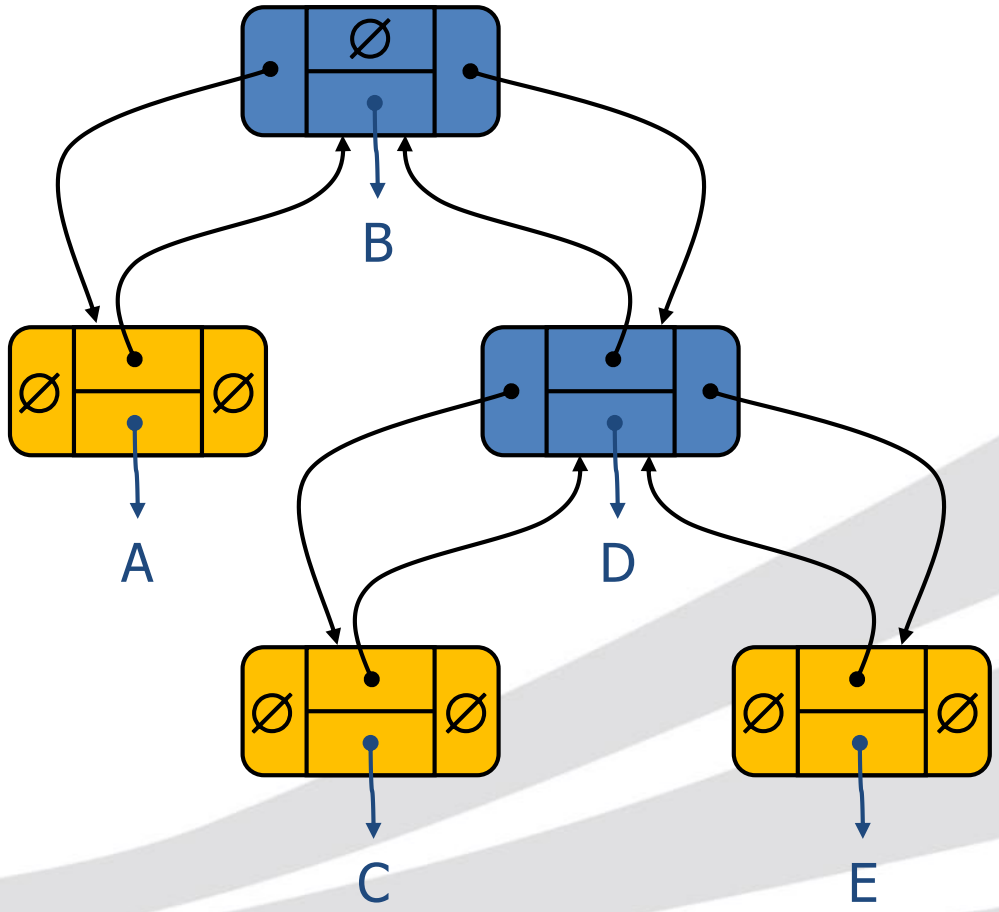


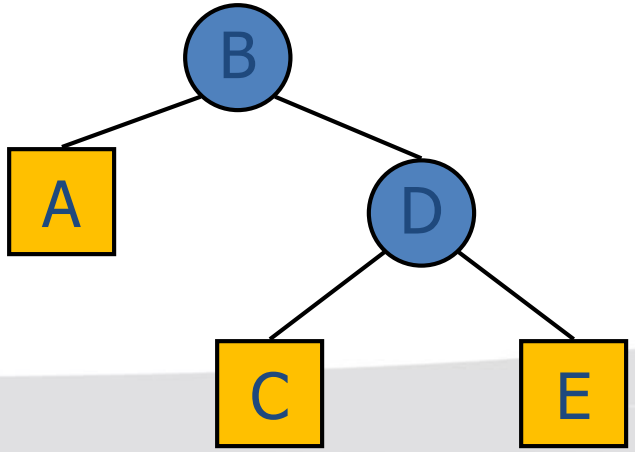
Bài tập: Hãy chỉ ra thứ tự các nút của cây dưới đây bằng phương pháp duyệt Inorder?



II. Cấu trúc liên kết cho cây nhị phân

- **Một nút là một đối tượng, đang lưu trữ:**
  - Phần tử (Element)
  - Nút cha (Parent node)
  - Nút con trái
  - Nút con phải
- **Mỗi nút thể hiện một ví trí trong ADT cây**





Cấu trúc BTreeNode biểu diễn cây nhị phân

- Thuộc tính
  - Object elem
  - BTreeNode *Parent
  - BTreeNode *Left
  - BTreeNode *Right

- Phương thức
  - BTreeNode *getParent()
  - BTreeNode *getLeft()
  - BTreeNode *getRight()
  - void setLeft(BTreeNode *)
  - void setRight(BTreeNode *)
  - void setParent(BTreeNode *)
  - int hasLeft()
  - int hasRight()
  - Object getElem()
  - void setElem(Object o)

Cấu trúc dữ liệu cây nhị phân

❑ **Thuộc tính**

- BTreeNode * root

❖ Các phương thức truy cập:

- BTreeNode *getroot()

❑ **Phương thức**

- int size()
- int isEmpty()
- int isInternal(BTreeNode *)
- int isExternal(BTreeNode *)
- int isRoot(BTreeNode *)
- void preOrder(BTreeNode *, void (*visit)(BTreeNode *))
- void inOrder(BTreeNode *, void (*visit)(BTreeNode *))
- void postOrder(BTreeNode *, void (*visit)(BTreeNode *))
- BTreeNode* insert(BTreeNode *parent, element)
- void remove(BTreeNode *);

III. Bài tập

1. Xây dựng lớp biểu diễn Cây nhị phân
2. Cài đặt các thuật toán duyệt cây.
3. Xây dựng lớp ứng dụng tạo cây, duyệt cây, tìm kiếm phần tử trên cây.

IV. Bài tập

- Bài toán: Độ sâu các nút trên cây BTS

Truy cập: http://laptrinhonline.club/problem/tichpxcaytknpdepth

- Code tham khảo: https://ideone.com/4BGjHs

Hết