

Chương 5. CÂY

ThS. Nguyễn Chí Hiếu

2017

Giới thiệu cấu trúc cây

Giới thiệu cây nhị phân

Các khái niệm về cây nhị phân

Một số tính chất của cây nhị phân

Cách lưu trữ cây nhị phân

Các thao tác duyệt cây nhị phân

Giới thiệu cấu trúc cây

duy h. me



Giới thiệu cấu trúc cây

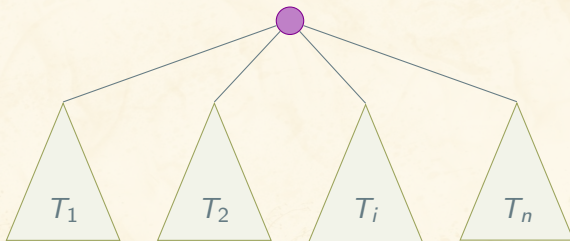
Các khái niệm cơ bản

- ▶ Cây (*tree*) là một tập hợp T gồm các nút (*phần tử*) của cây.
- ▶ Trong đó, có một nút đặc biệt được gọi là nút gốc (*root*), các nút còn lại được chia thành những tập rời nhau T_1, T_2, \dots, T_n theo quan hệ phân cấp. Mỗi T_i cũng là một cây.

Giới thiệu cấu trúc cây

Các khái niệm cơ bản

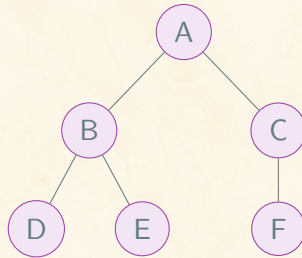
- ▶ Cây (*tree*) là một tập hợp T gồm các nút (*phần tử*) của cây.
- ▶ Trong đó, có một nút đặc biệt được gọi là nút gốc (*root*), các nút còn lại được chia thành những tập rời nhau T_1, T_2, \dots, T_n theo quan hệ phân cấp. Mỗi T_i cũng là một cây.



Giới thiệu cấu trúc cây

Ví dụ 1

Cho cây biểu diễn dãy ký tự A, B, C, D, E, F.



Giới thiệu cấu trúc cây

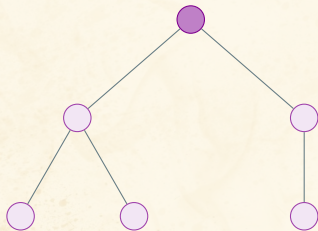
Các khái niệm cơ bản

- ▶ **Bậc** (*degree*) của một nút: là số cây con của nút đó.
- ▶ **Bậc** của một cây: là bậc **lớn nhất** của các nút trong cây (*số cây con tối đa của một nút thuộc cây*). Cây có bậc n thì gọi là cây n -phân.

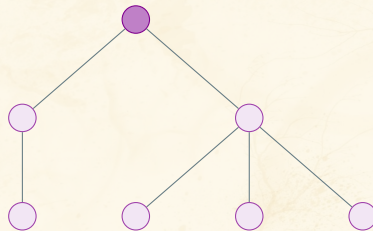
Giới thiệu cấu trúc cây

Các khái niệm cơ bản

- ▶ **Bậc (degree)** của một nút: là số cây con của nút đó.
- ▶ **Bậc của một cây**: là bậc **lớn nhất** của các nút trong cây (*số cây con tối đa của một nút thuộc cây*). Cây có bậc n thì gọi là cây n -phân.



Hình 1: Bậc của cây T_1 là 2 (cây nhị phân).

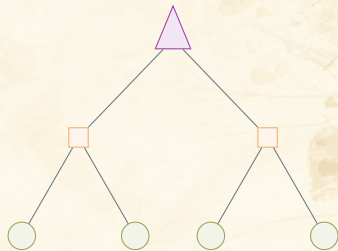


Hình 2: Bậc của cây T_2 là 3.

Giới thiệu cấu trúc cây

Các khái niệm cơ bản

- ▶ Nút (*node*): là một phần tử trong cây.
- ▶ Nút cha (*parent*).
- ▶ Nút con (*child*).
- ▶ Nút anh em (*sibling*): là những nút có cùng nút cha.
- ▶ Nút gốc (*root*): là nút không có nút cha.
- ▶ Nút trong (*internal*): là nút có bậc khác 0 và không phải là nút gốc.
- ▶ Nút lá (*leaf*): là nút có bậc bằng 0.



Giới thiệu cấu trúc cây

Mức (*level*) hay chiều sâu (*depth*) của một nút

- ▶ Nếu p là nút gốc, thì

$$d(p) = 0.$$

- ▶ Ngược lại,

$$d(p) = d(\text{parent}(p)) + 1.$$

Giới thiệu cấu trúc cây

Chiều cao (*height*) của một nút

Là độ dài đường đi (*path*) hay số cạnh (*edge*) lớn nhất từ nút đó đến nút lá.

- Nếu p là nút lá, thì

$$height(p) = 0.$$

- Ngược lại,

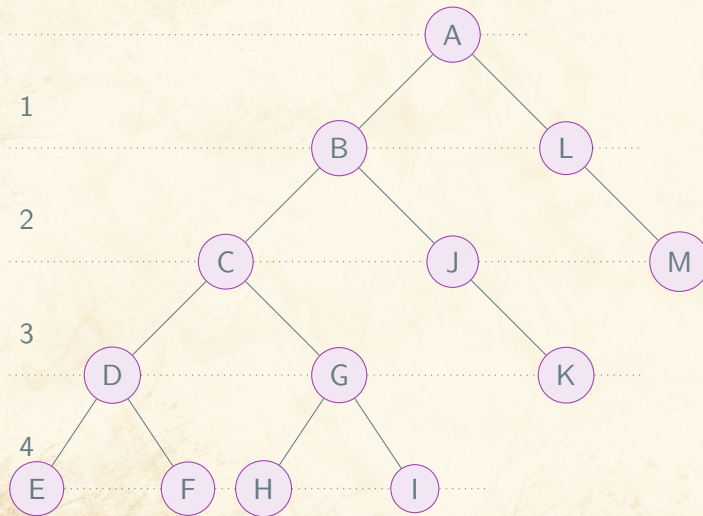
$$height(p) = \max \{height(T_i)\} + 1.$$

với T_i là các cây con của nút p .

Chiều cao của cây chính là chiều cao của *nút gốc*.

Giới thiệu cấu trúc cây

duyetho

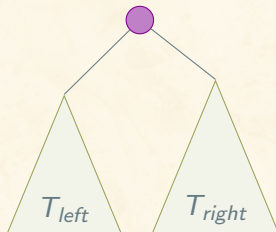


- ▶ Bậc của nút A?
- ▶ Bậc của nút M ?
- ▶ Bậc của cây ?
- ▶ Mức của nút C ?
- ▶ Mức của nút H ?
- ▶ Chiều cao của cây ?

Các khái niệm về cây nhị phân

Cây nhị phân (*binary tree*)

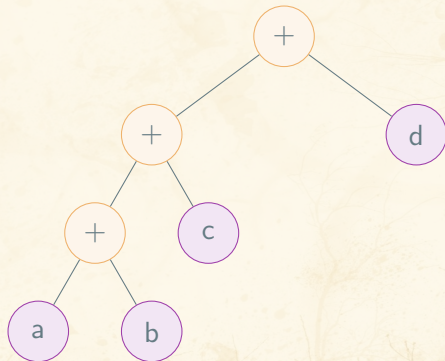
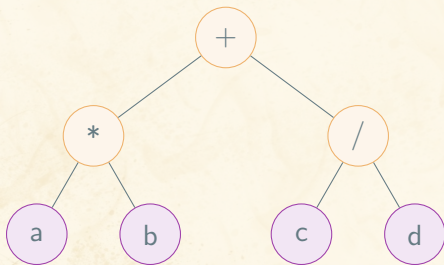
- Là cây mà mỗi nút có **tối đa 2 cây con** (cây con trái và cây con phải).



Các khái niệm về cây nhị phân

Ví dụ 2

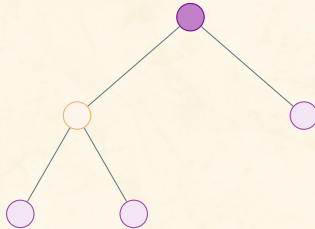
Cây nhị phân mô tả biểu thức $(a * b) + (c / d)$ và $((a + b) + c) + d$.



Các khái niệm về cây nhị phân

Full binary tree

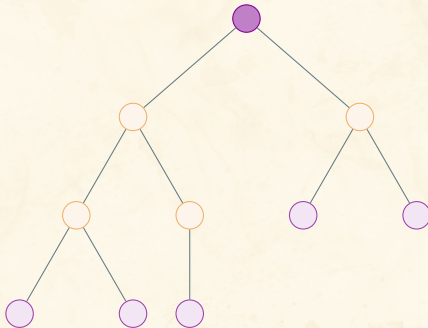
- Là cây nhị phân mà mỗi nút trong (*internal*) có **đầy đủ 2 cây con**.



Các khái niệm về cây nhị phân

Complete binary tree

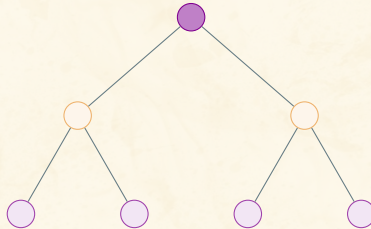
- Là cây nhị phân mà mỗi nút trong (*internal*) ở mức $d - 1$ có **đầy đủ 2 cây con** và các cây con ở mức d được điền từ *trái sang phải*.



Các khái niệm về cây nhị phân

Perfect binary tree

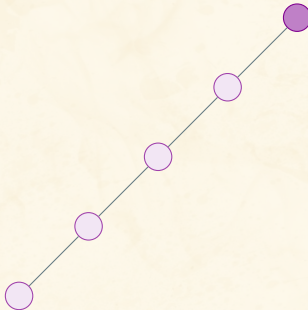
- Là một *complete binary tree* mà tất cả nút lá *có cùng mức*.



Các khái niệm về cây nhị phân

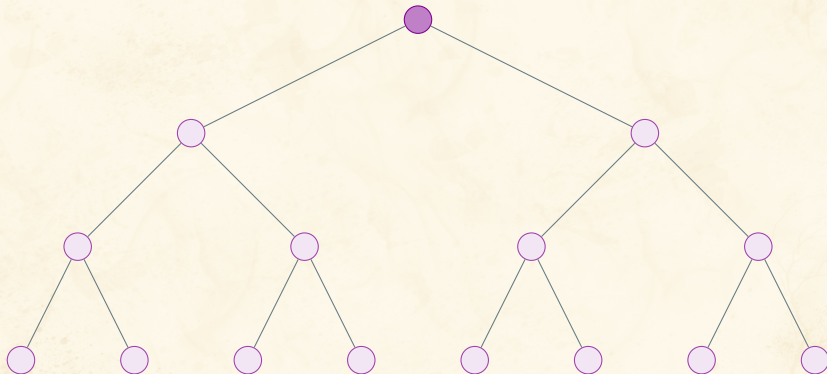
Cây suy biến (*skewed tree*)

- ▶ Là cây nhị phân *bị lệch về một nhánh*.



Một số tính chất của cây nhị phân

Cho cây nhị phân T có chiều cao h



Một số tính chất của cây nhị phân

1. Số nút nằm ở mức d

$$n(d) \leq 2^d.$$

2. Số nút lá

$$n(h) \leq 2^{h-1}.$$

3. Số nút trong

$$n(h) \leq 2^h - 1.$$

4. Tổng số nút

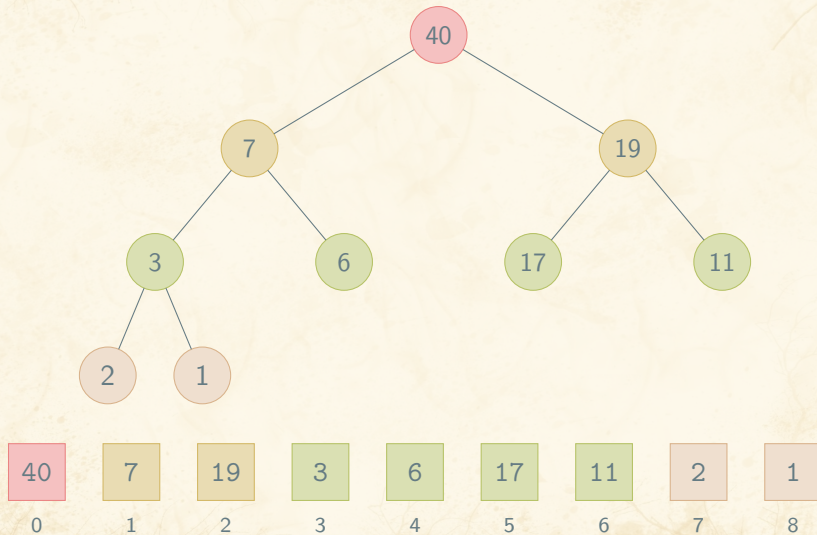
$$n(h) \leq 2^{h+1} - 1.$$

5. Chiều cao của cây

$$h \geq \log_2(n+1) - 1 \approx \log_2 n.$$

Cách lưu trữ cây nhị phân

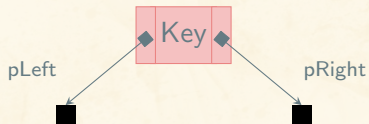
duyetho



Cách lưu trữ cây nhị phân

Cấu trúc dữ liệu của một nút trong cây

- ▶ Thành phần dữ liệu: khóa (*key*) của một nút.
- ▶ Thành phần liên kết: con trỏ *pLeft* liên kết với *cây con trái* và con trỏ *pRight* liên kết với *cây con phải*.



Cấu trúc dữ liệu của một cây

- ▶ Chỉ cần một con trỏ trỏ đến nút gốc của cây.

Cách lưu trữ cây nhị phân

Định nghĩa cấu trúc dữ liệu của một nút

```
1 typedef int Data;  
2 struct Node  
3 {  
4     Data    Key;  
5     Node    *pLeft;  
6     Node    *pRight;  
7 };
```


Các thao tác duyệt cây nhị phân

Duyệt cây

- ▶ Là thuật toán liệt kê danh sách tất cả các nút của một cây, mỗi nút chỉ một lần.
- ▶ Mỗi thuật toán duyệt cây khác nhau ở *thứ tự duyệt nút gốc* của cây con đang xét.
- ▶ Ba phương pháp duyệt cây:
 - ▶ Tiền thứ tự (*preorder*) hay **NLR** (*node-left-right*).
 - ▶ Trung thứ tự (*inoder*) hay **LNR** (*left-node-right*).
 - ▶ Hậu thứ tự (*postorder*) hay **LRN** (*left-right-node*).

Các thao tác duyệt cây nhị phân

Thao tác duyệt

- ▶ Hàm duyệt cây chứa *hai lời gọi hàm đệ quy* tương ứng cây con trái và cây con phải.
- ▶ Ngoài ra, hàm duyệt sẽ chứa các thao tác xử lý tùy theo yêu cầu như: tìm kiếm, in, ... một nút.

Hàm 1: Traverse(pRoot)

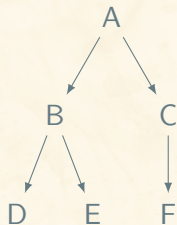
- Đầu vào: cây nhị phân với gốc pRoot.
- Đầu ra:

```
1  if cây con đang xét khác rỗng
2      Traverse(pRoot->pLeft) // Duyệt cây con trái
3      Traverse(pRoot->pRight) // Duyệt cây con phải
```

Các thao tác duyệt cây nhị phân

Ví dụ 3

Duyệt cây T theo phương pháp tiền thứ tự, trung thứ tự và hậu thứ tự.

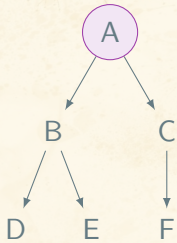


Hình 3: Cây T gồm 6 đỉnh A, B, C, D, E, F .

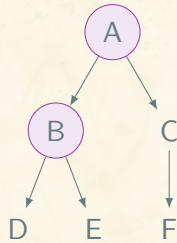
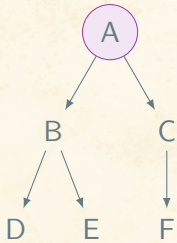
Duyệt tiền thứ tự (NLR)

```
1 void NLR(Node *pRoot)
2 {
3     if (pRoot != NULL)
4     {
5         // Thao tác xử lý nút đang xét ...
6         NLR(pRoot->pLeft);
7         NLR(pRoot->pRight);
8     }
9 }
```

Duyệt tiên thứ tự (NLR)

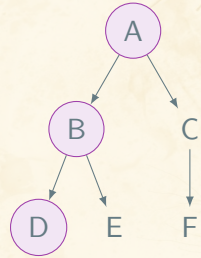
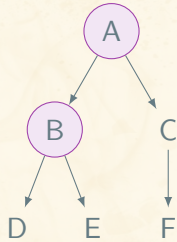
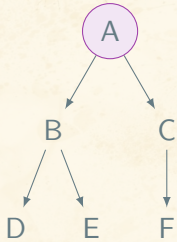


Duyệt tiên thứ tự (NLR)



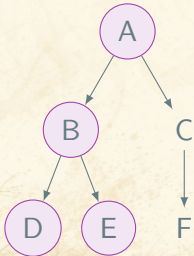
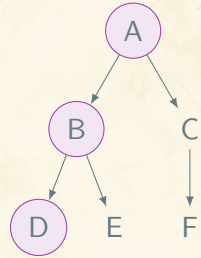
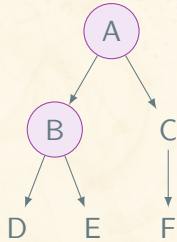
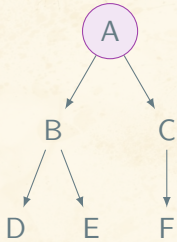
Duyệt tiên thứ tự (NLR)

duyên me



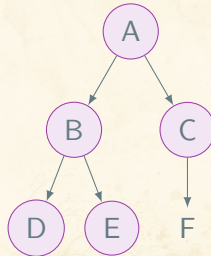
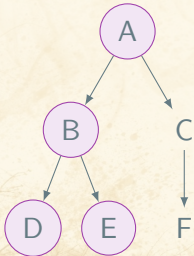
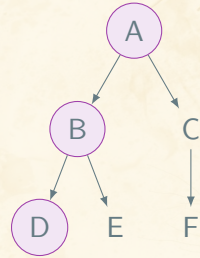
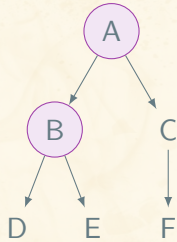
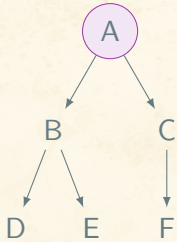
Duyệt tiên thứ tự (NLR)

duyệt



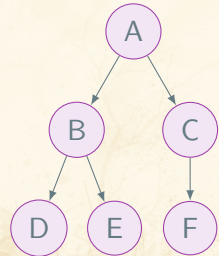
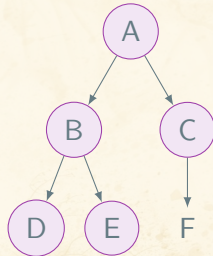
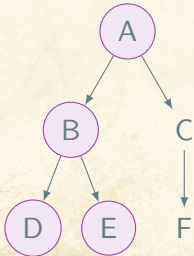
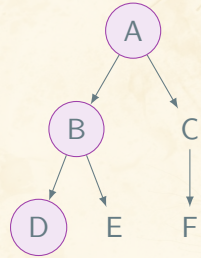
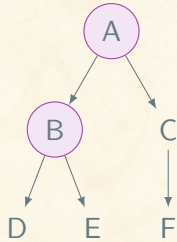
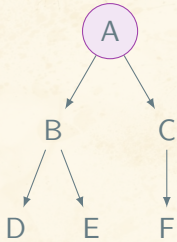
Duyệt tiên thứ tự (NLR)

duyệt



Duyệt tiên thứ tự (NLR)

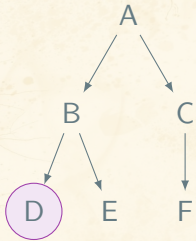
duyệt



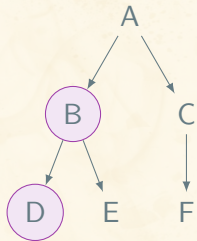
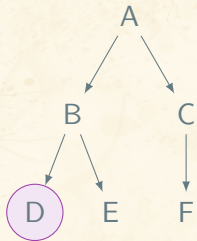
Duyệt trung thứ tự (LNR)

```
1 void LNR(Node *pRoot)
2 {
3     if(pRoot != NULL)
4     {
5         LNR(pRoot->pLeft);
6         // Thao tác xử lý nút đang xét ...
7         LNR(pRoot->pRight);
8     }
9 }
```

Duyệt trung thứ tự (LNR)

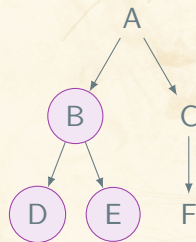
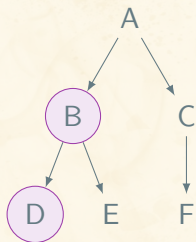
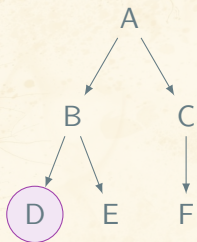


Duyệt trung thứ tự (LNR)



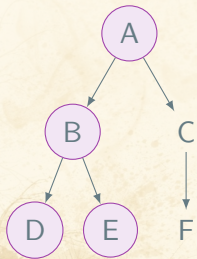
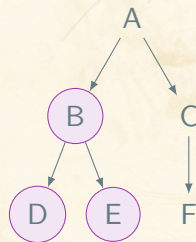
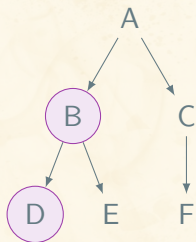
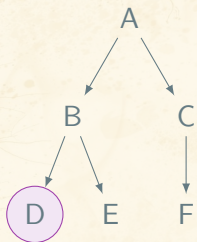
Duyệt trung thứ tự (LNR)

duyệt



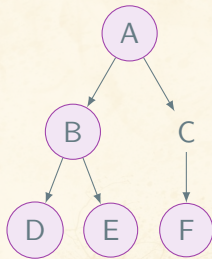
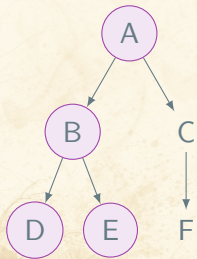
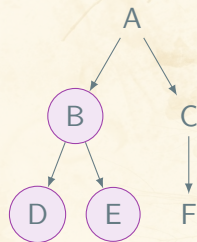
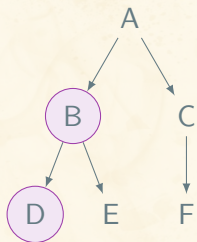
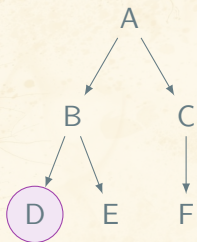
Duyệt trung thứ tự (LNR)

duyệt



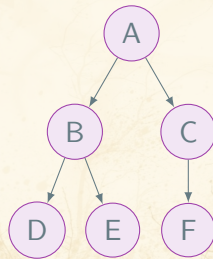
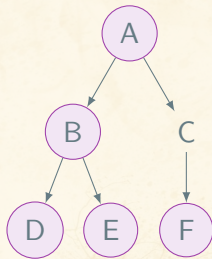
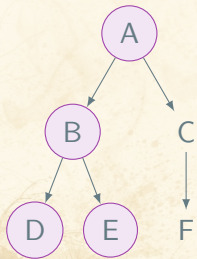
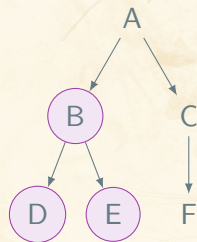
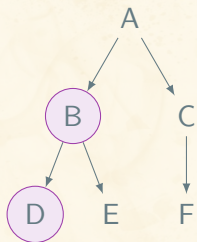
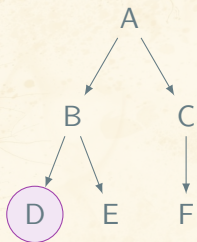
Duyệt trung thứ tự (LNR)

duyệt



Duyệt trung thứ tự (LNR)

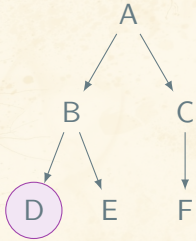
duyệt



Duyệt hậu thứ tự (LRN)

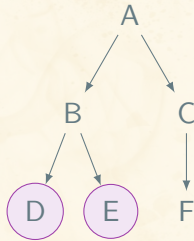
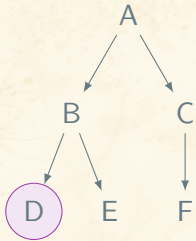
```
1 void LRN(Node *pRoot)
2 {
3     if(pRoot != NULL)
4     {
5         LRN(pRoot->pLeft);
6         LRN(pRoot->pRight);
7         // Thao tac xu ly nut dang xet ...
8     }
9 }
```

Duyệt hậu thứ tự (LRN)



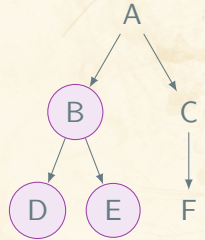
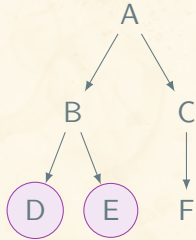
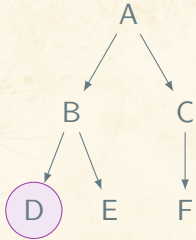
Duyệt hậu thứ tự (LRN)

duyệt



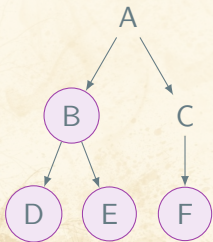
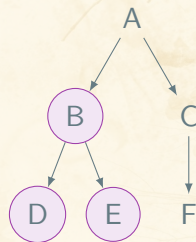
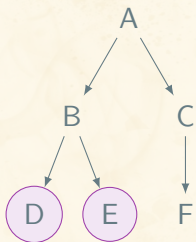
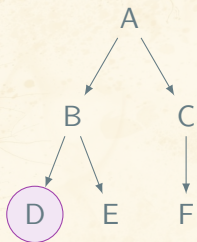
Duyệt hậu thứ tự (LRN)

duyệt



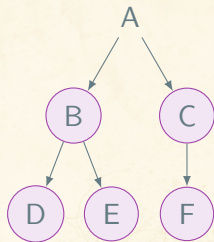
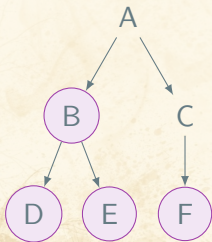
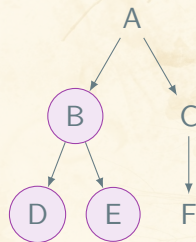
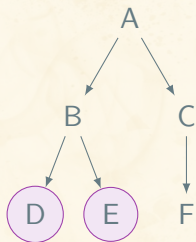
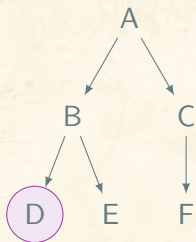
Duyệt hậu thứ tự (LRN)

duyệt



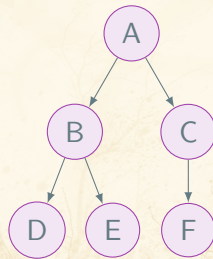
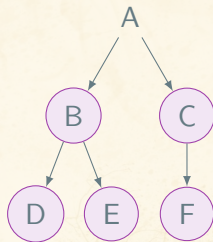
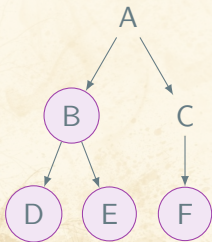
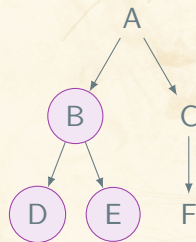
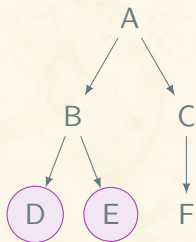
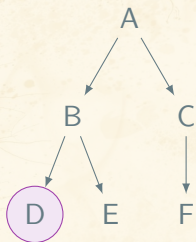
Duyệt hậu thứ tự (LRN)

duyệt



Duyệt hậu thứ tự (LRN)

duyệt



Cài đặt các hàm xử lý yêu cầu sau:

1. Tìm nút có khóa là k trong cây nhị phân..
2. Tính tổng giá trị các nút trong cây nhị phân.
3. Xác định số nút lá cây nhị phân.
4. Xác định chiều cao của cây nhị phân.



Dương Anh Đức, Trần Hạnh Nhi.

Nhập môn Cấu trúc dữ liệu và Thuật toán.

Đại học Khoa học tự nhiên TP Hồ Chí Minh, 2003.



Donald E. Knuth.

The Art of Computer Programming, Volume 3.

Addison-Wesley, 1998.



Niklaus Wirth.

Algorithms + Data Structures = Programs.

Prentice-Hall, 1976.



Robert Sedgewick.

Algorithms in C.

Addison-Wesley, 1990.