Cây

Mục tiêu

- Giới thiệu khái niệm cấu trúc cây.
- Cấu trúc dữ liệu cây nhị phân tìm kiếm: tổ chức, các thuật toán, ứng dụng.
- Giới thiệu cấu trúc dữ liệu cây nhị phân tìm kiếm cân bằng.

Cấu trúc cây

Cây- một số định nghĩa

- **Định nghĩa 1:** cây là một tập hợp T các phần tử (gọi là nút của cây) trong đó có 1 nút đặc biệt được gọi là gốc, các nút còn lại được chia thành những tập rời nhau T1, T2 , ... , Tn theo quan hệ phân cấp trong đó Ti cũng là một cây. Mỗi nút ở cấp i sẽ quản lý một số nút ở cấp i+1. Quan hệ này người ta còn gọi là quan hệ cha-con.
- **Định nghĩa 2:** cấu trúc cây với kiểu cơ sở T là một nút cấu trúc rỗng được gọi là cây rỗng (NULL). Một nút mà thông tin chính của nó có kiểu T, nó liên kết với một số hữu hạn các cấu trúc cây khác cũng có kiểu cơ sở T. Các cấu trúc này được gọi là những cây con của cây đang xét.

Cây - Một số khái niệm cơ bản

- Bậc 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.
- Nút gốc: là nút không có nút cha.
- Nút lá: là nút có bậc bằng 0 .
- Nút nhánh: là nút có bậc khác 0 và không phải là gốc .
- Mức của một nút:
- Mức (gốc (T)) = 0.
- Gọi T1, T2, T3, ... , Tn là các cây con của T0
- Mức (T1) = Mức (T2) = ... = Mức (Tn) = Mức (T0) + 1.

- Độ dài đường đi từ gốc đến nút x: là số nhánh cần đi qua kể từ gốc đến x.
- Độ dài đường đi tổng của cây :

$$P_T = \sum_{X \in T} P_X$$

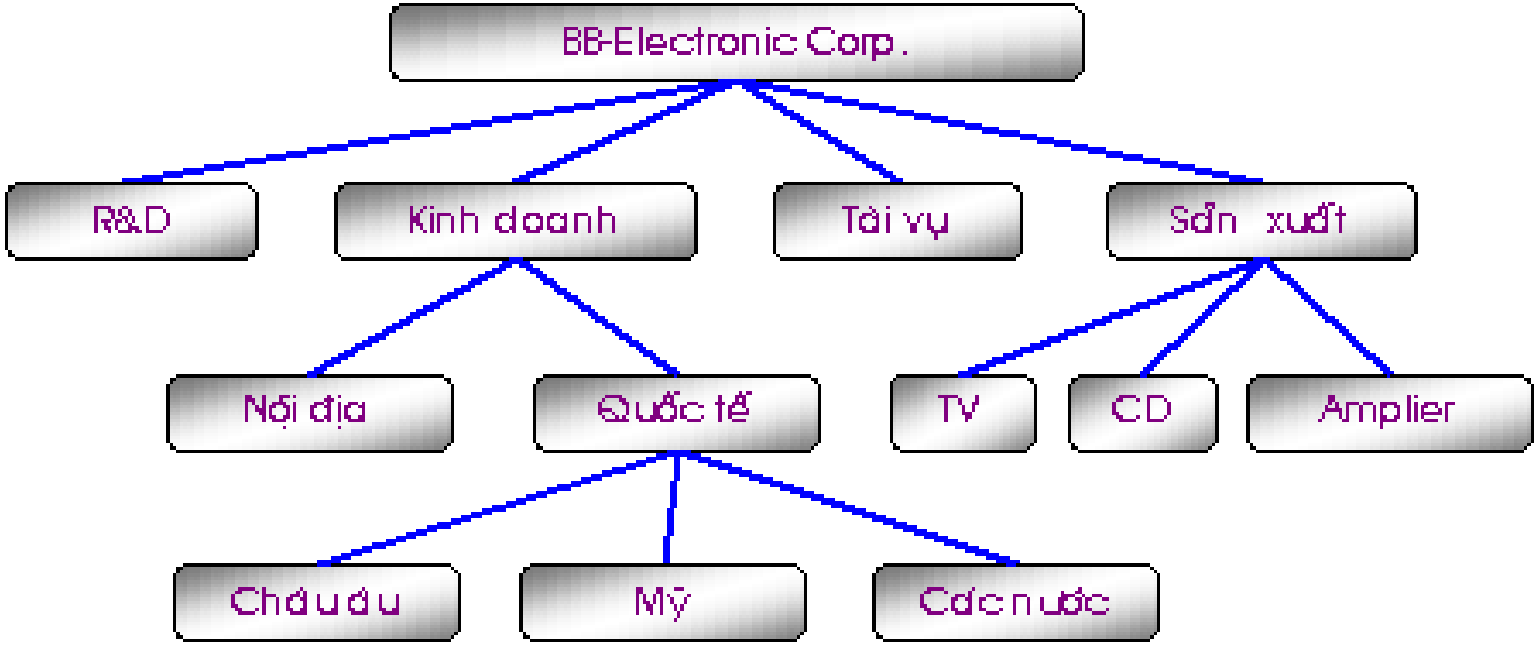
trong đó Px là độ dài đường đi từ gốc đến X.

Độ dài đường đi trung bình : PI = PT/n (n là số nút trên cây T).

Rừng cây: là tập hợp nhiều cây trong đó thứ tự các cây là quan trọng.

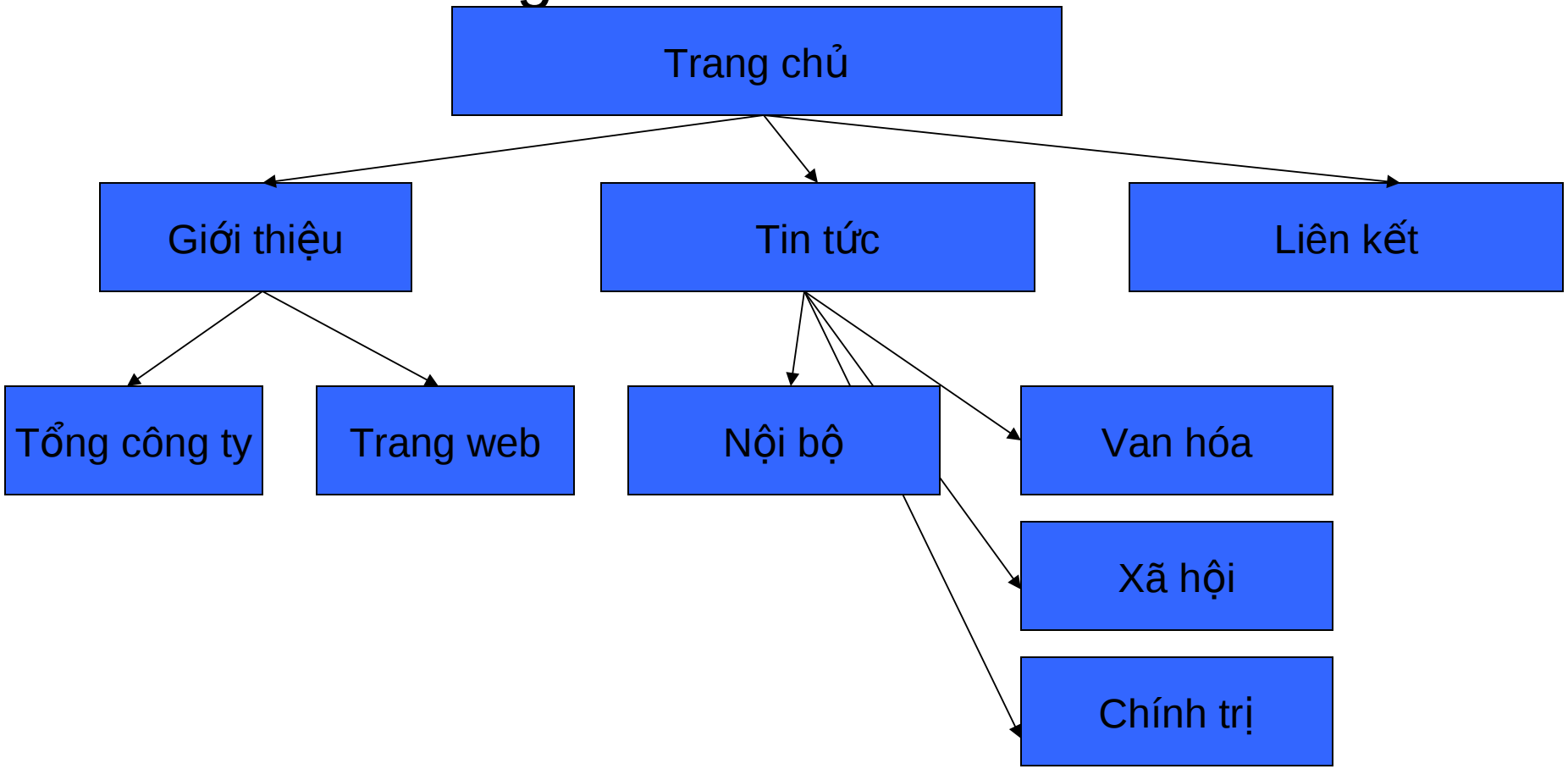
Ví dụ

- Sơ đồ tổ chức công ty



Ví dụ

- cấu trúc trang web

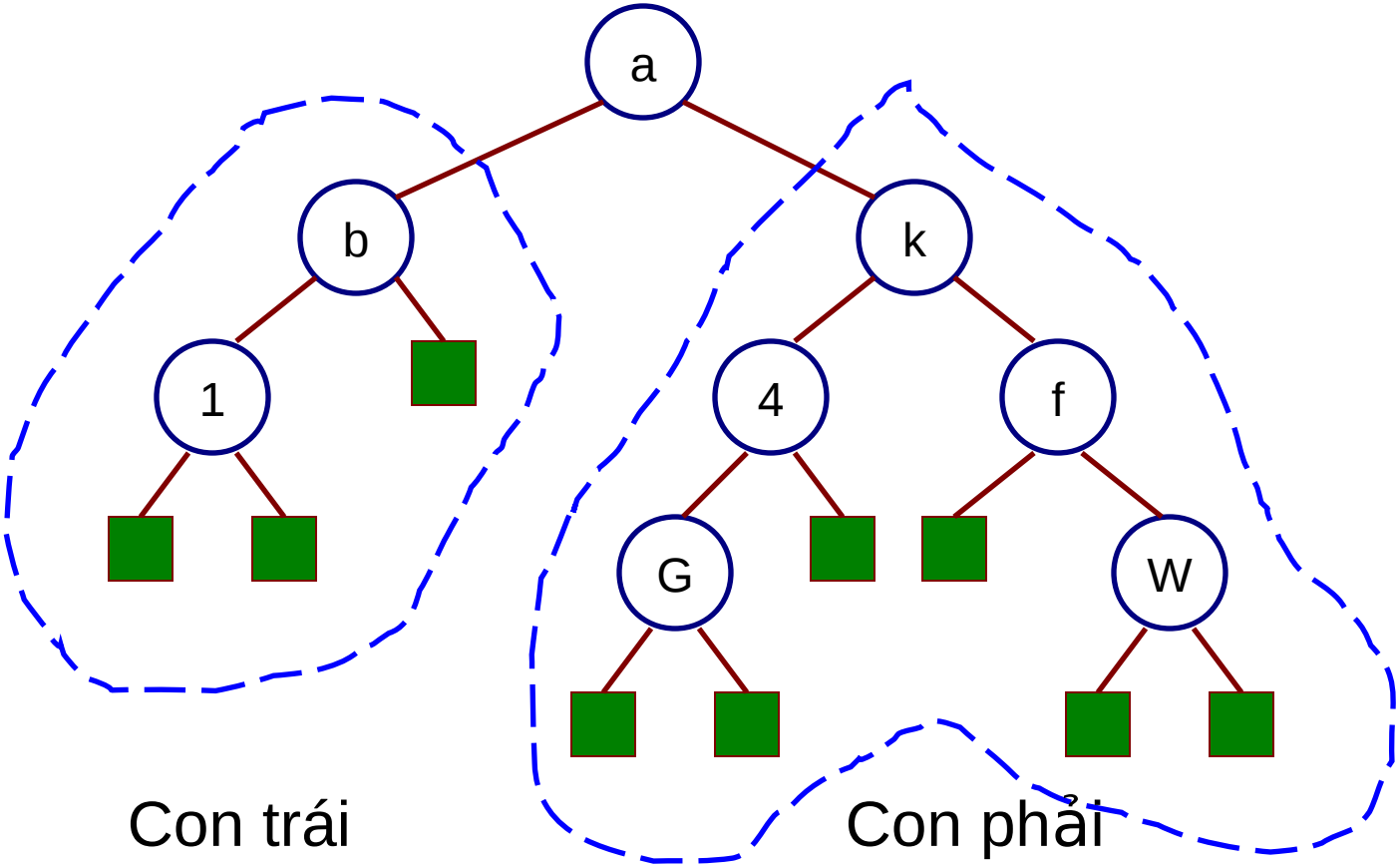


- Mục lục một quyển sách
- Cấu trúc cây thư mục trong DOS/WIN
- Cấu trúc thư viện, …
- **Nhận xét:**
- Trong cấu trúc cây không tồn tại chu trình
- Tổ chức 1 cấu trúc cây cho phép truy cập nhanh đến các phần tử của nó.

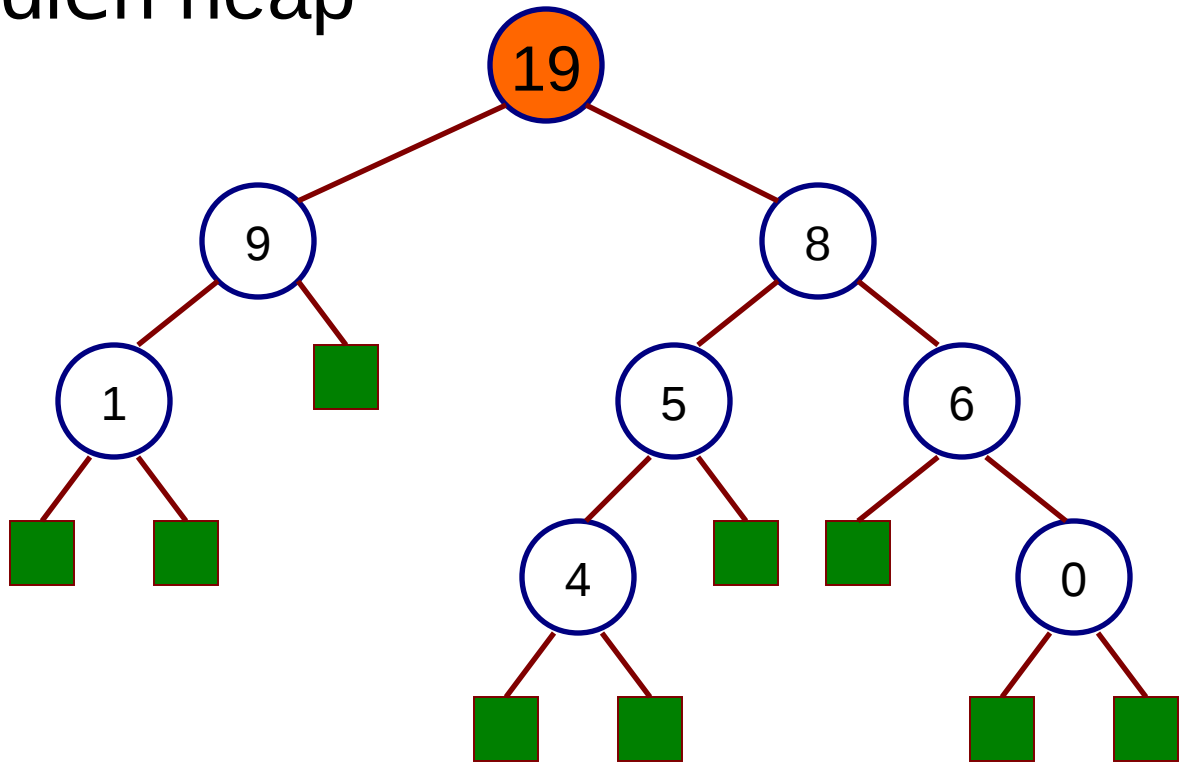
Cây nhị phân

- *Cây nhị phân là cây mà mỗi nút có tối đa 2 cây con*
- Trong thực tế thường gặp các cấu trúc có dạng cây nhị phân. Một cây tổng quát có thể biểu diễn thông qua cây nhị phân.

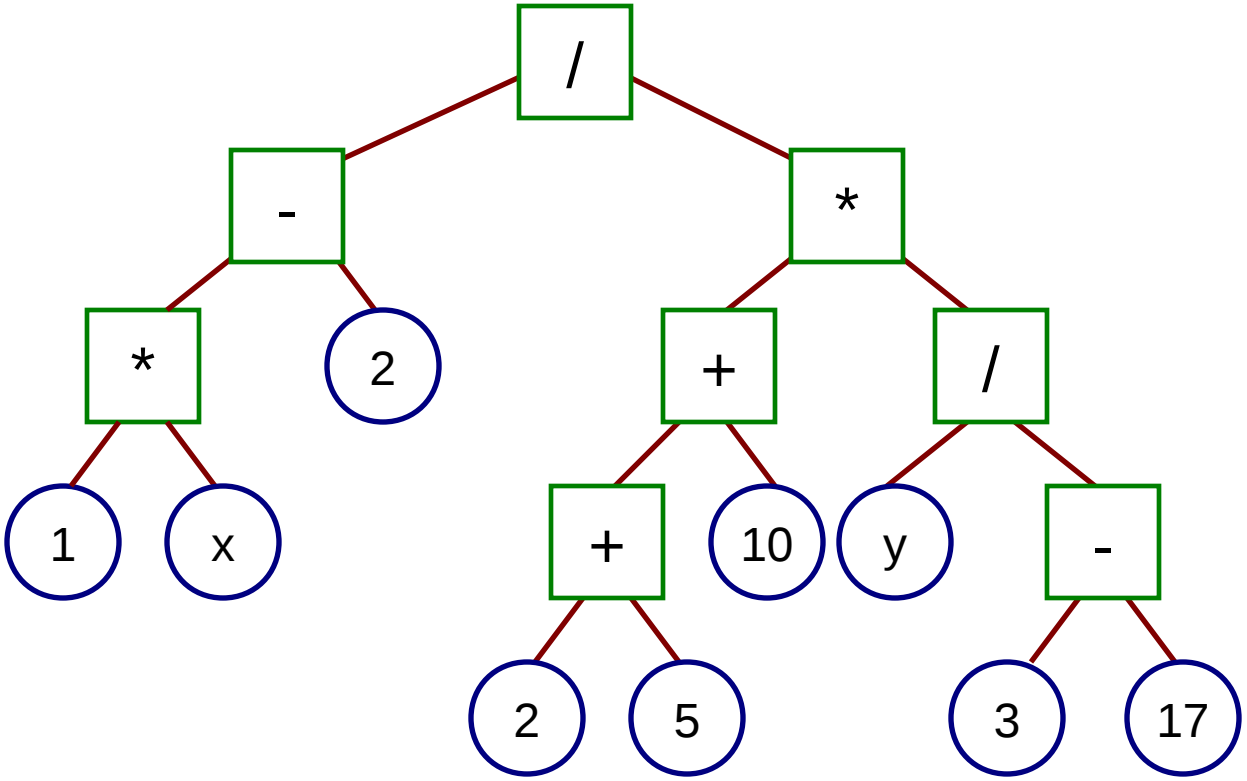
Hình ảnh cây nhị phân



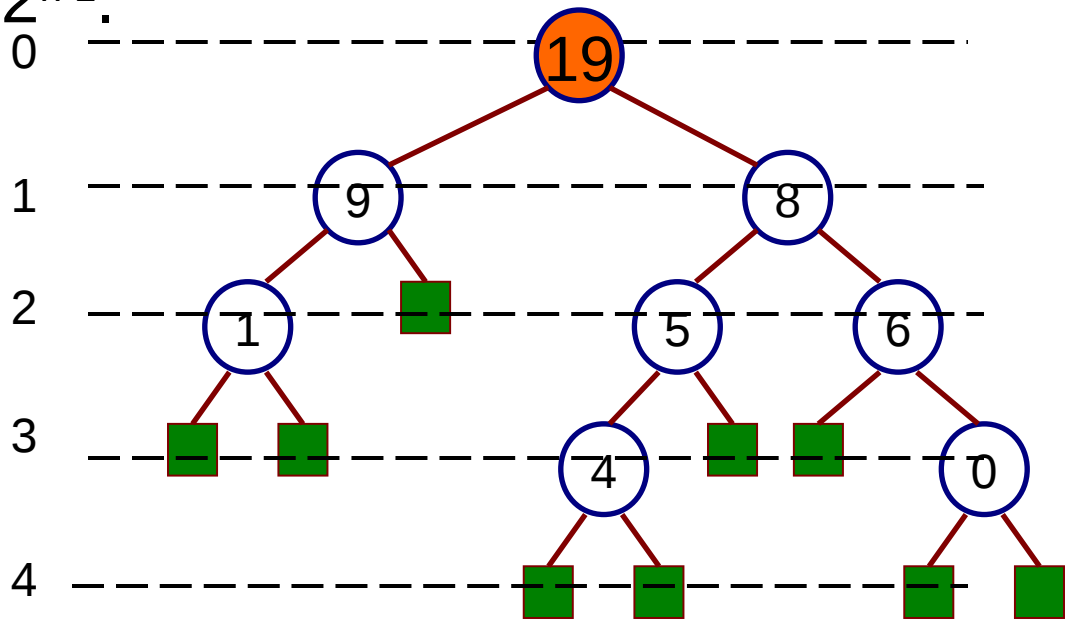
- Biểu diễn heap



- Biểu diễn tính toán biểu thức
- ((1*x)-2)/((2+5+10)*(y/(3-17)))



- **Một số tính chất của cây nhị phân:**
- Số nút nằm ở mức I nhỏ hơn 2^I.
- Số nút lá $\leq 2^{h-1}$, với h là chiều cao của cây.
- Chiều cao của cây h $\geq \log_2$(số nút trong cây).
- Số nút trong cây $\leq 2^{h-1}$.



- **Biểu diễn cây nhị phân T**
- Cây nhị phân là một cấu trúc bao gồm các phần tử (nút) được kết nối với nhau theo quan hệ “cha-con” với mỗi cha có tối đa 2 con. Để biểu diễn cây nhị phân ta chọn phương pháp cấp phát liên kết. Ứng với một nút, ta dùng một biến động lưu trữ các thông tin:
- Thông tin lưu trữ tại nút.
- Địa chỉ nút gốc của cây con trái trong bộ nhớ.
- Địa chỉ nút gốc của cây con phải trong bộ nhớ.

C

Khai báo tương ứng trong ngôn ngữ C có thể như sau:

```
struct CTreeNode
{
    Data        Key;//Data là kiểu dữ liệu ứng với thông tin lưu tại nút
     CTreeNode *TLeft, *TRight;
…
}
CtreeNode *root;
```

Do tính chất mềm dẻo của cách biểu diễn bằng cấp phát liên kết, phương pháp này được dùng chủ yếu trong biểu diễn cây nhị phân.

C#

Khai báo tương ứng trong ngôn ngữ C có thể như sau:

```
class  CTreeNode
{
   Data        Key;//Data là kiểu dữ liệu ứng với thông tin lưu tại nút
    CTreeNode TLeft, TRight;
…
}
Class root:CtreeNode{};
```

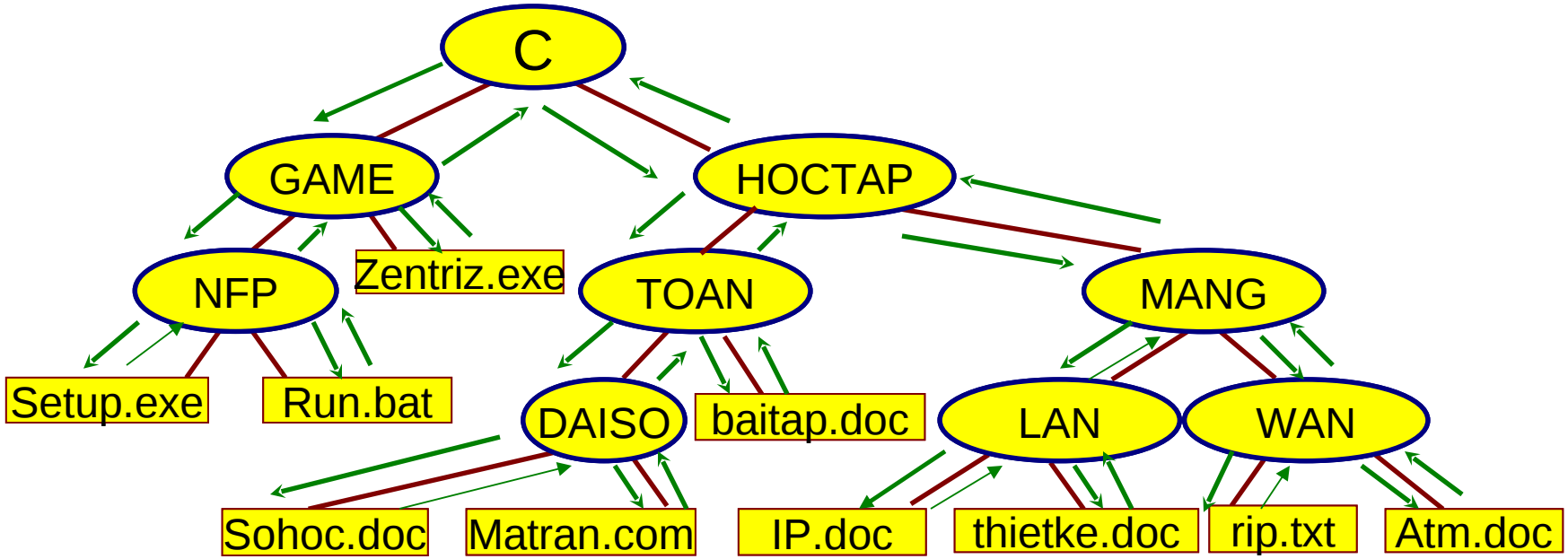
Do tính chất mềm dẻo của cách biểu diễn bằng cấp phát liên kết, phương pháp này được dùng chủ yếu trong biểu diễn cây nhị phân.

- Có 3 kiểu duyệt chính có thể áp dụng trên cây nhị phân:
- duyệt theo thứ tự trước (NLR),
- thứ tự giữa (LNR)
- thứ tự sau (LRN)

- **Duyệt theo thứ tự trước (Node-Left-Right)**
- Kiểu duyệt này trước tiên thăm nút gốc sau đó thăm các nút của cây con trái rồi đến cây con phải. Thủ tục duyệt có thể trình bày đơn giản như sau:

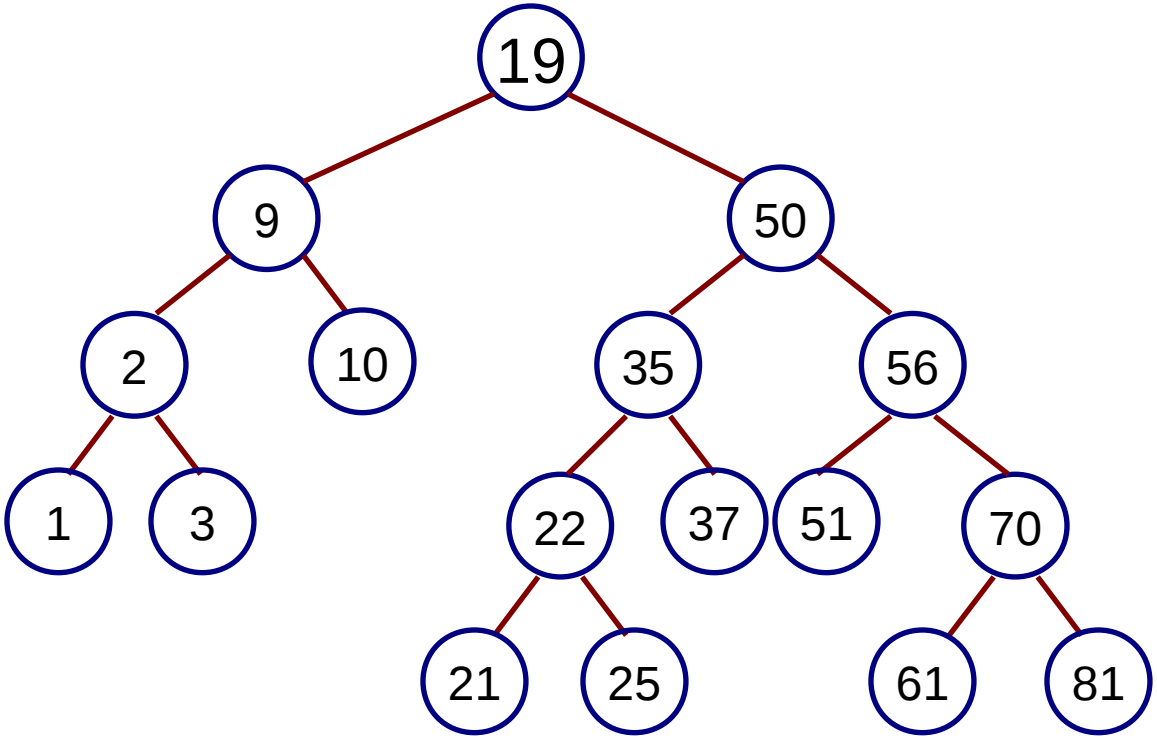
```
void      NLR(CTReeNode *a)
    {
    if (a != NULL)
    {
<Xử lý a.key>;  //Xử lý tương ứng theo nhu cầu
    NLR(a->TLeft);
    NLR(a->TRight);
    }
}
```

- Ví dụ:
- Duyệt thông tin của ổ đĩa C

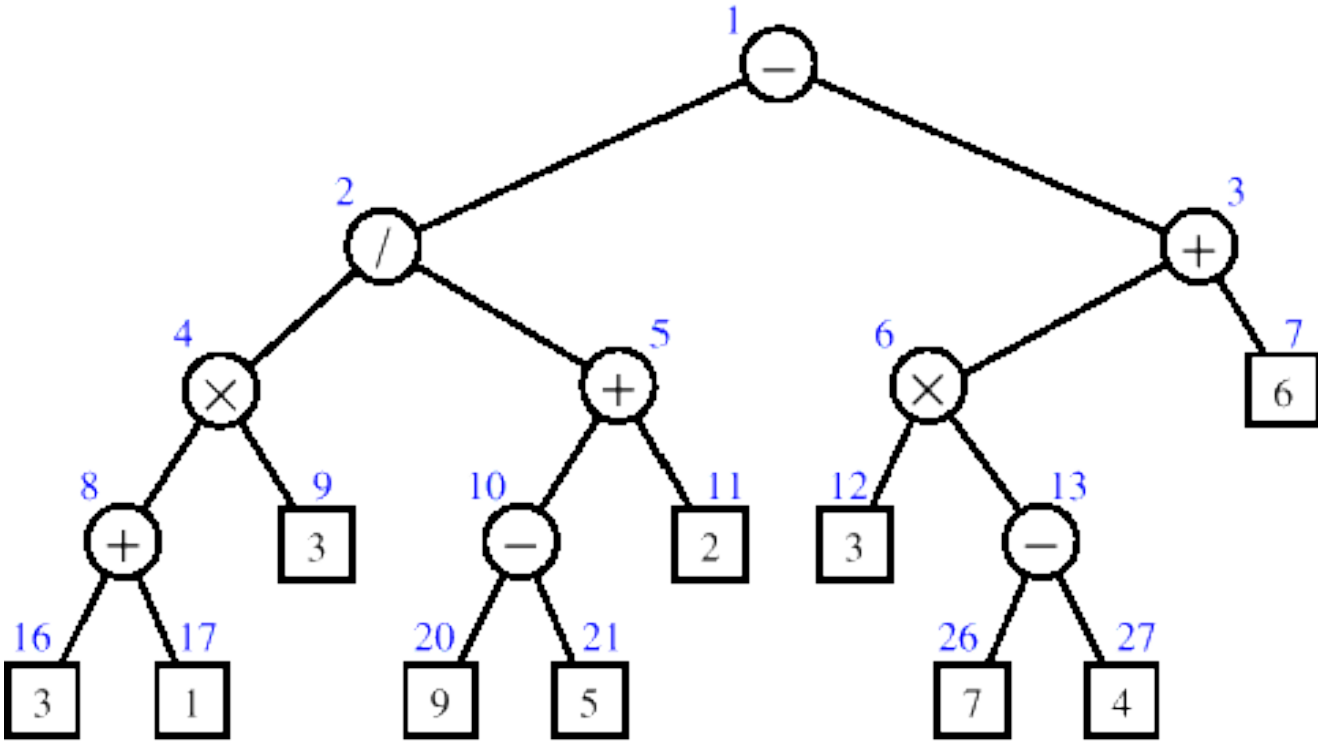


- **Duyệt theo thứ tự giữa (Left- Node-Right)**
- Kiểu duyệt này trước tiên thăm các nút của cây con trái sau đó thăm nút gốc rồi đến cây con phải. Thủ tục duyệt có thể trình bày đơn giản như sau:
- void LNR(CTREEnode *a)
  {
- if (a != NULL)
  {
- LNR(a.Left);
  <Xử lý a.key>; //Xử lý tương ứng theo nhu cầu
  LNR(a.Right);
- }

- từ cây nhị phân→ mảng sắp xếp

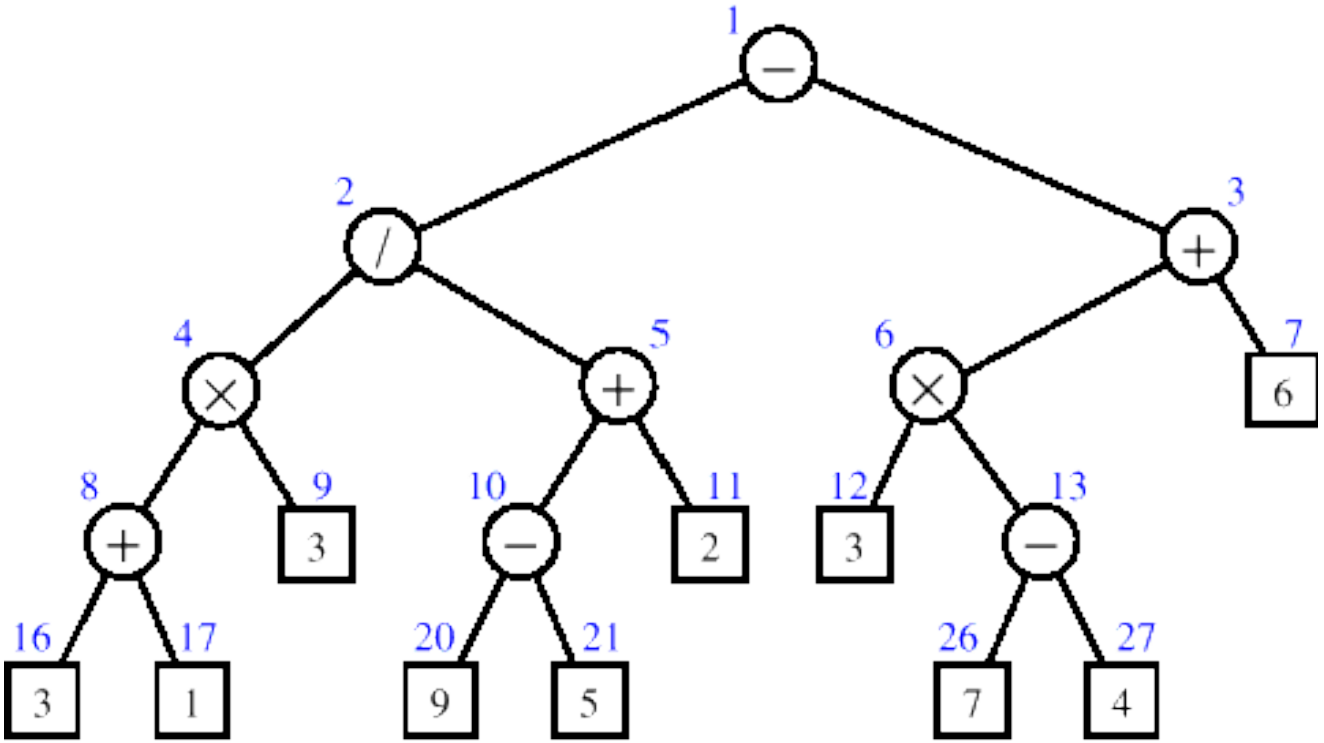


1,2,3,9,10,19,21,22,25,35,37,50,51,56,61,70,81



$$(3 + 1)\times 3/(9 -– 5 + 2) – (3\times (7 – 4) + 6) = –13$$

- **Duyệt theo thứ tự sau (Left-Right-Node)**
- Kiểu duyệt này trước tiên thăm các nút của cây con trái sau đó thăm đến cây con phải rồi cuốio cùng mới thăm nút gốc. Thủ tục duyệt có thể trình bày đơn giản như sau:
- void LRN(TREE Root)
  {
- if (Root != NULL)
  {
- LRN(Root->Left);
  LRN(Root->Right);
  <Xử lý Root>; //Xử lý tương ứng theo nhu cầu
- }
- }



(3 1+ 3 x 9 5-2+/ 3 7 4 – x6+-