Cây và cây nhị phân

Bởi:

Trần Hạnh Nhi

CẤU TRÚC CÂY

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 T_1, T_2 , ... , 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.

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 T_1, T_2, T_3, ... , Tn là các cây con của T_0

Mức (T_1) = Mức (T_2) = ... = Mức (Tn) = Mức (T_0) + 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 :

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

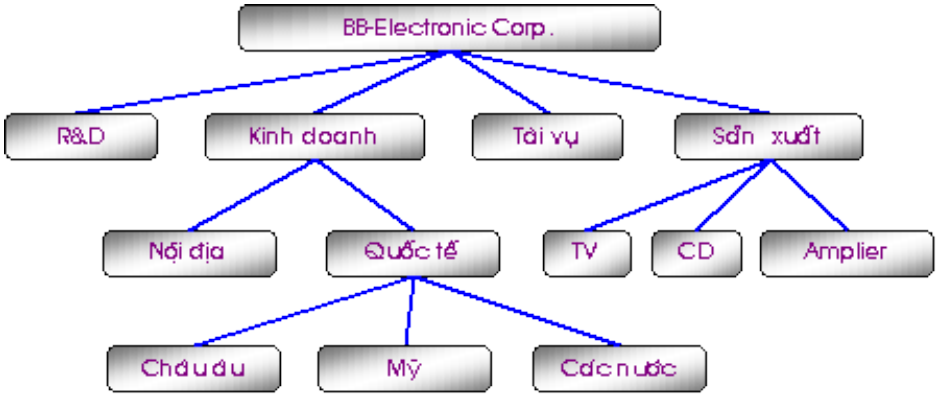
trong đó P_X là độ dài đường đi từ gốc đến X.

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.

Một số ví dụ về đối tượng các cấu trúc dạng cây

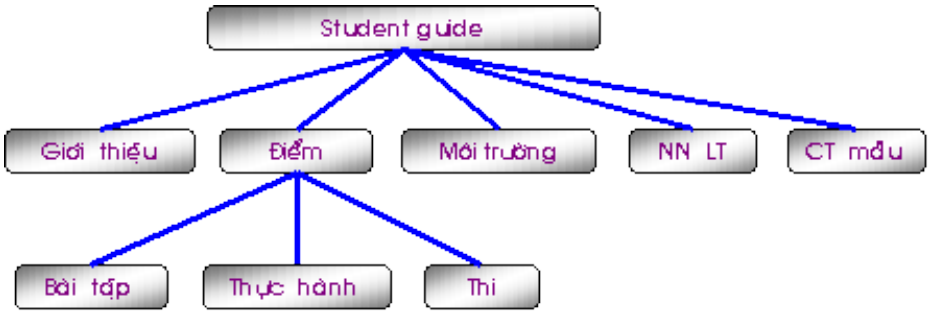
Sơ đồ tổ chức của một công ty



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

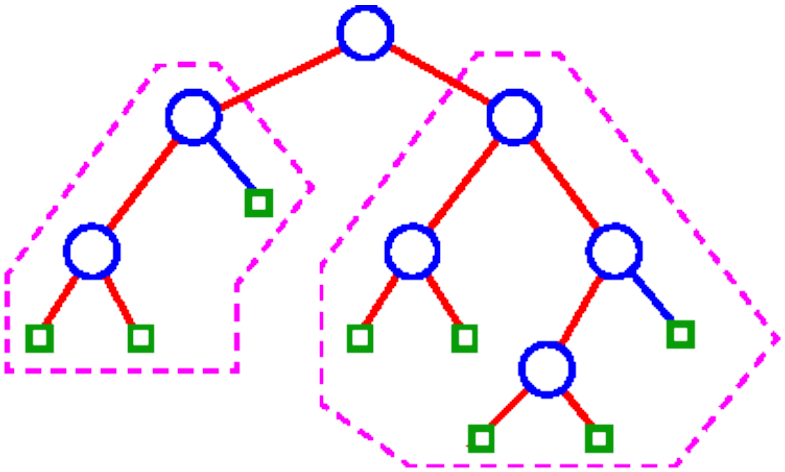
Tổ chức 1 cấu trúc cây cho phép truy cập nhanh đến các phần tử của nó.

CÂY NHỊ PHÂN

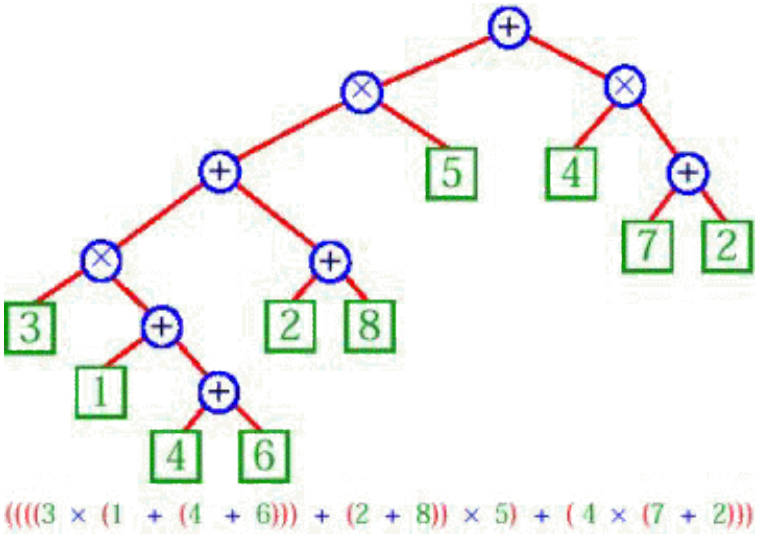
Định nghĩa

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.

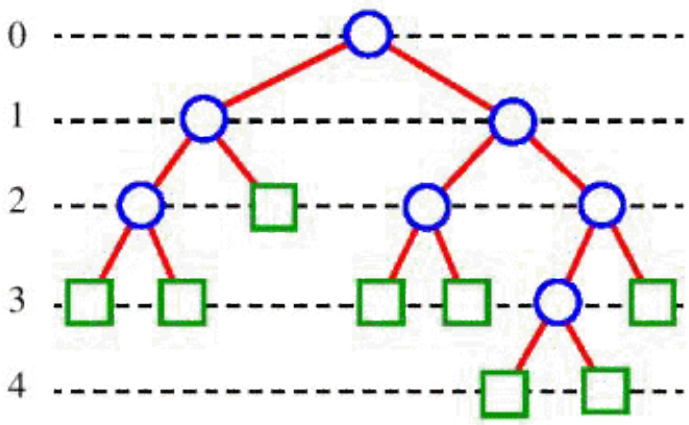


Cây nhị phân có thể ứng dụng trong nhiều bài toán thông dụng. Ví dụ dưới đây cho ta hình ảnh của một biểu thức toán học:



Một số tính chất của cây nhị phân:

- Số nút nằm ở mức I £ 2^I.
- Số nút lá £ 2^{h-1}, với h là chiều cao của cây.
- Chiều cao của cây h ³ $\log_2$(số nút trong cây).
- Số nút trong cây £ 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.

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

```
typedef struct    tagTNODE

{

    Data  Key;//Data là kiểu dữ liệu ứng với thông tin lưu tại nút

    struct tagNODE *pLeft, *pRight;

}TNODE;

typedef TNODE     *TREE;
```

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. Từ đây trở đi, khi nói về cây nhị phân, chúng ta sẽ dùng phương pháp biểu diễn này.

Duyệt cây nhị phân

Nếu như khi khảo sát cấu trúc dữ liệu dạng danh sách liên kết ta không quan tâm nhiều đến bài toán duyệt qua tất cả các phần tử của chúng thì bài toán duyệt cây hết sức quan trọng. Nó là cốt lõi của nhiều thao tác quan trong khác trên cây. Do cây nhị phân là một cấu trúc dữ liệu phi tuyến nên bài toán duyệt cây là bài toán không tầm thường.

Có nhiều kiểu duyệt cây khác nhau, và chúng cũng có những ứng dụng khác nhau. Đối với cây nhị phân, do cấu trúc đệ qui của nó, việc duyệt cây tiếp cận theo kiểu đệ qui là hợp lý và đơn giản nhất. Sau đây chúng ta sẽ xem xét một số kiểu duyệt thông dụng.

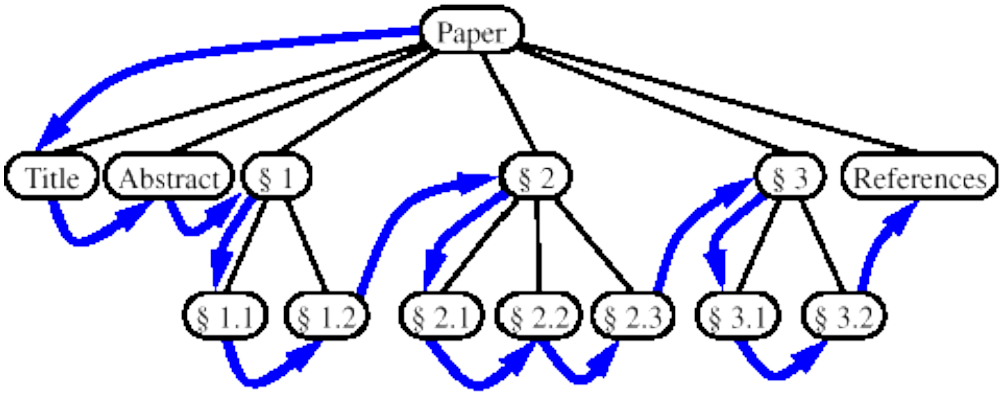
Có 3 kiểu duyệt chính có thể áp dụng trên cây nhị phân: duyệt theo thứ tự trước (NLR), thứ tự giữa (LNR) và thứ tựi sau (LRN). Tên của 3 kiểu duyệt này được đặt dựa trên trình tự của việc thăm nút gốc so với việc thăm 2 cây con.

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(TREE Root)  {

    if (Root != NULL)       {

<Xử lý Root>;                 //Xử lý tương ứng theo nhu cầu NLR(Root->pLeft);
 NLR(Root->pRight);

    }
```

}



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:

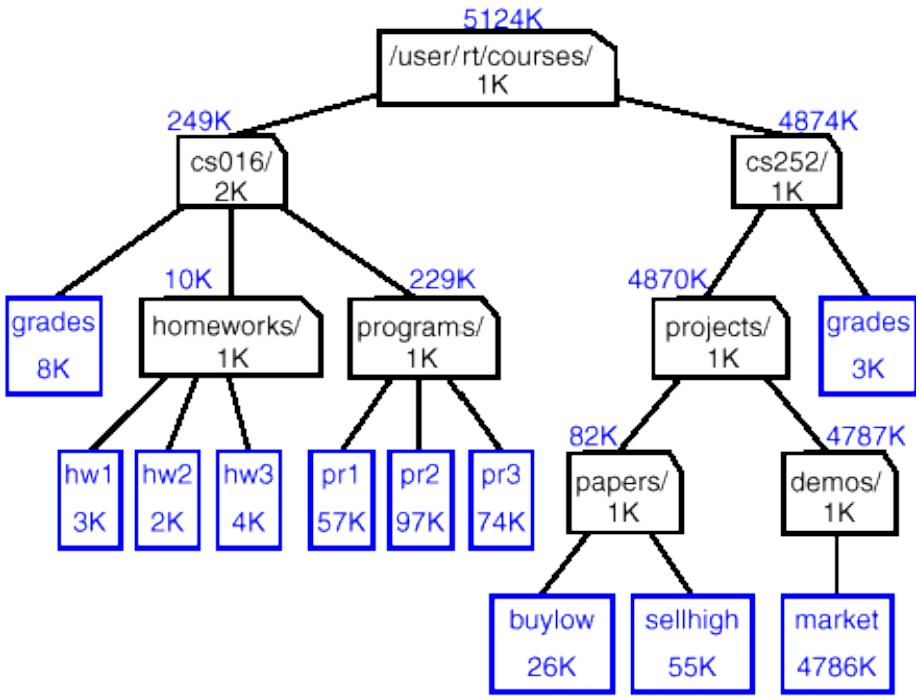
```
void  LNR(TREE Root)  {

    if (Root != NULL)       {

LNR(Root->Left); <Xử lý Root>;              //Xử lý tương ứng theo nhu cầu
LNR(Root->Right);

    }

}
```

Duyệt theo thứ tự sau (Left-Right-Node)

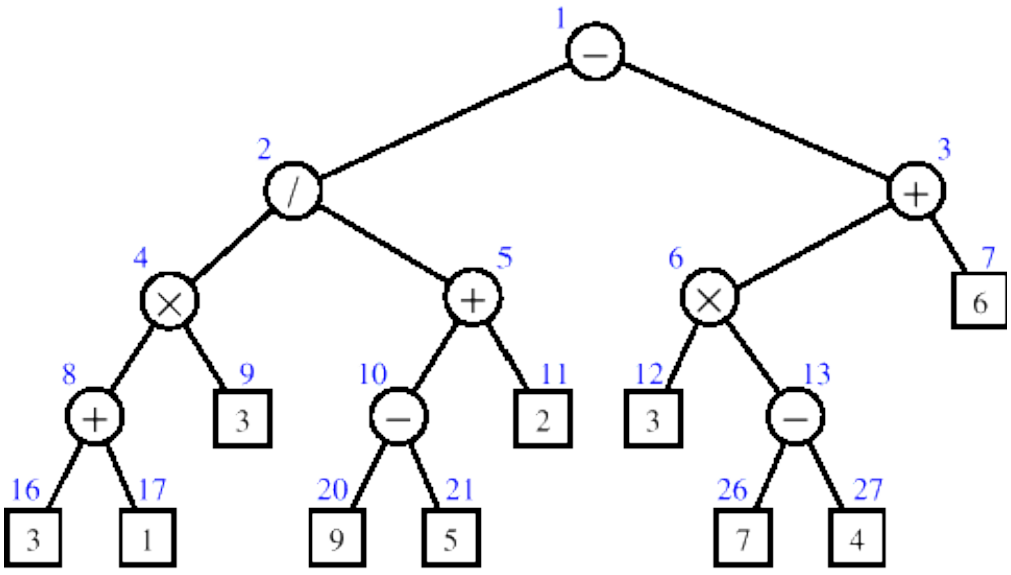
Kiểu duyệt này trước tiên thăm các nút của cây con trái sau đó thăm đến cây con phải rồi cuốio cùng mới thăm nút gốc. Thủ tục duyệt có thể trình bày đơn giản như sau:

```
void  LRN(TREE Root)  {

    if (Root != NULL)       {

LRN(Root->Left); LRN(Root->Right); <Xử lý Root>;   //Xử lý tương ứng theo nhu cầu

    }

}
```

Một ví dụ quen thuộc trong tin học về ứng dụng của duyệt theo thứ tự sau là việc xác định tổng kích thước của một thư mục trên đĩa như hình sau:



Một ứng dụng quan trọng khác của phép duyệt cây theo thứ tự sau là việc tính toán giá trị của biểu thức dựa trên cây biểu thức như hình dưới:



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

Biểu diễn cây tổng quát bằng cây nhị phân

Nhược điểm của các cấu trúc cây tổng quát là bậc của các nút trên cây có thể dao động trong một biên độ lớn Þ việc biểu diễn gặp nhiều khó khăn và lãng phí. Hơn nữa, việc xây dựng các thao tác trên cây tổng quát phức tạp hơn trên cây nhị phân nhiều. Vì vậy, thường nếu không quá cần thiết phải sử dụng cây tổng quát, người ta chuyển cây tổng quát thành cây nhị phân.

Ta có thể biến đổi một cây bất kỳ thành một cây nhị phân theo qui tắc sau:

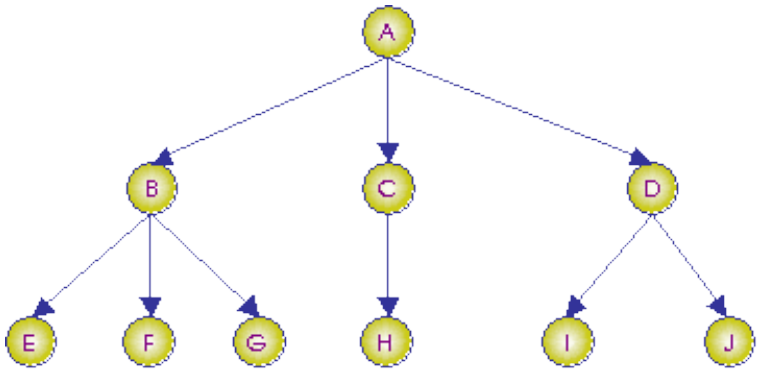
Giữ lại nút con trái nhất làm nút con trái.

Các nút con còn lại chuyển thành nút con phải.

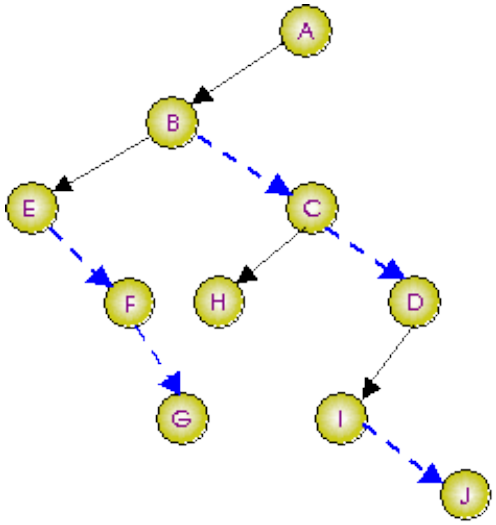
Như vậy, trong cây nhị phân mới, con trái thể hiện quan hệ cha con và con phải thể hiện quan hệ anh em trong cây tổng quát ban đầu.

Ta có thể xem ví dụ dưới đây để thấy rõ hơn qui trình.

Giả sử có cây tổng quát như hình bên dưới:



Cây nhị phân tương ứng sẽ như sau:



Một cách biểu diễn cây nhị phân khác

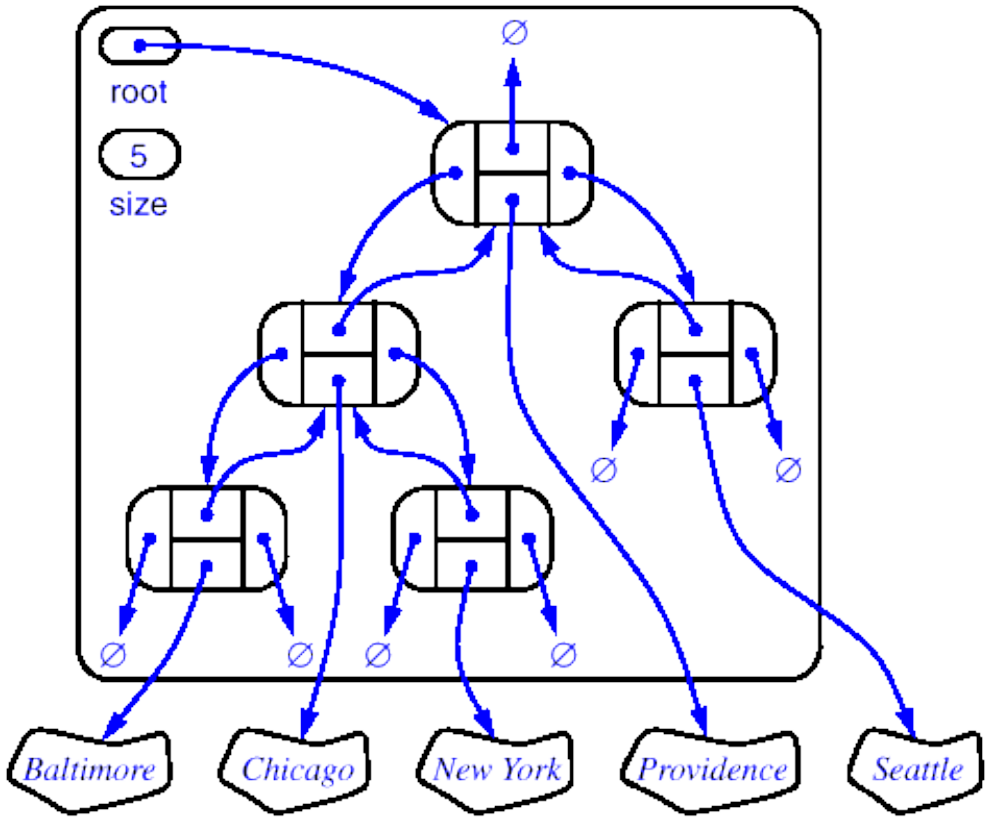
Đôi khi, khi định nghĩa cây nhị phân, người ta quan tâm đến cả quan hệ 2 chiều cha con chứ không chỉ một chiều như định nghĩa ở phần trên. Lúc đó, cấu trúc cây nhị phân có thể định nghĩa lại như sau:

typedef struct tagTNode {

DataType Key; struct tagTNode* pParent; struct tagTNode* pLeft; struct tagTNode* pRight;

}TNODE;

typedef TNODE *TREE;


root
5
size
∅
∅
∅
∅
∅
∅
∅
∅
Baltimore
Chicago
New York
Providence
Seattle