

Cây và cây nhị phân (Trees and Binary Trees)

Bài giảng môn Cấu trúc dữ liệu và giải thuật

Khoa Công nghệ thông tin

Trường Đại học Thủy Lợi

Nội dung

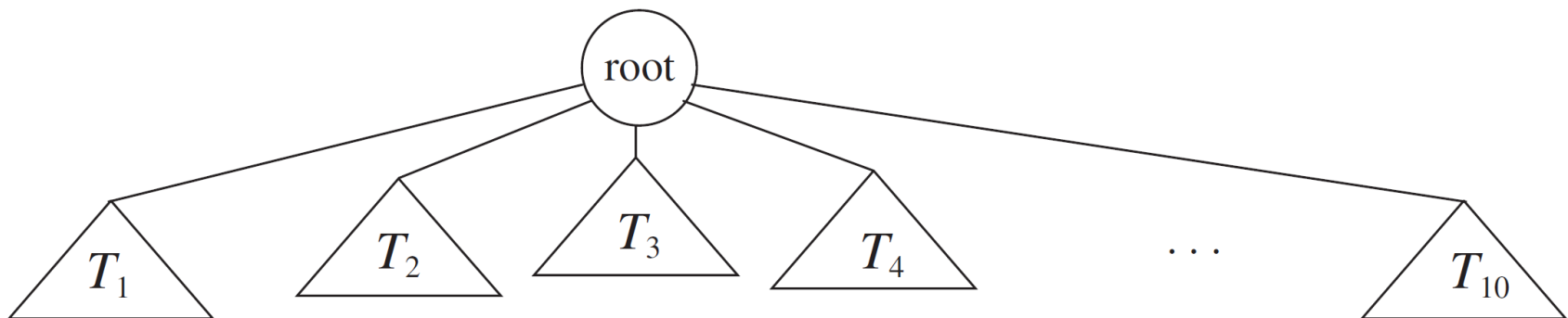
1. Cây
2. Cây nhị phân

1. Cây

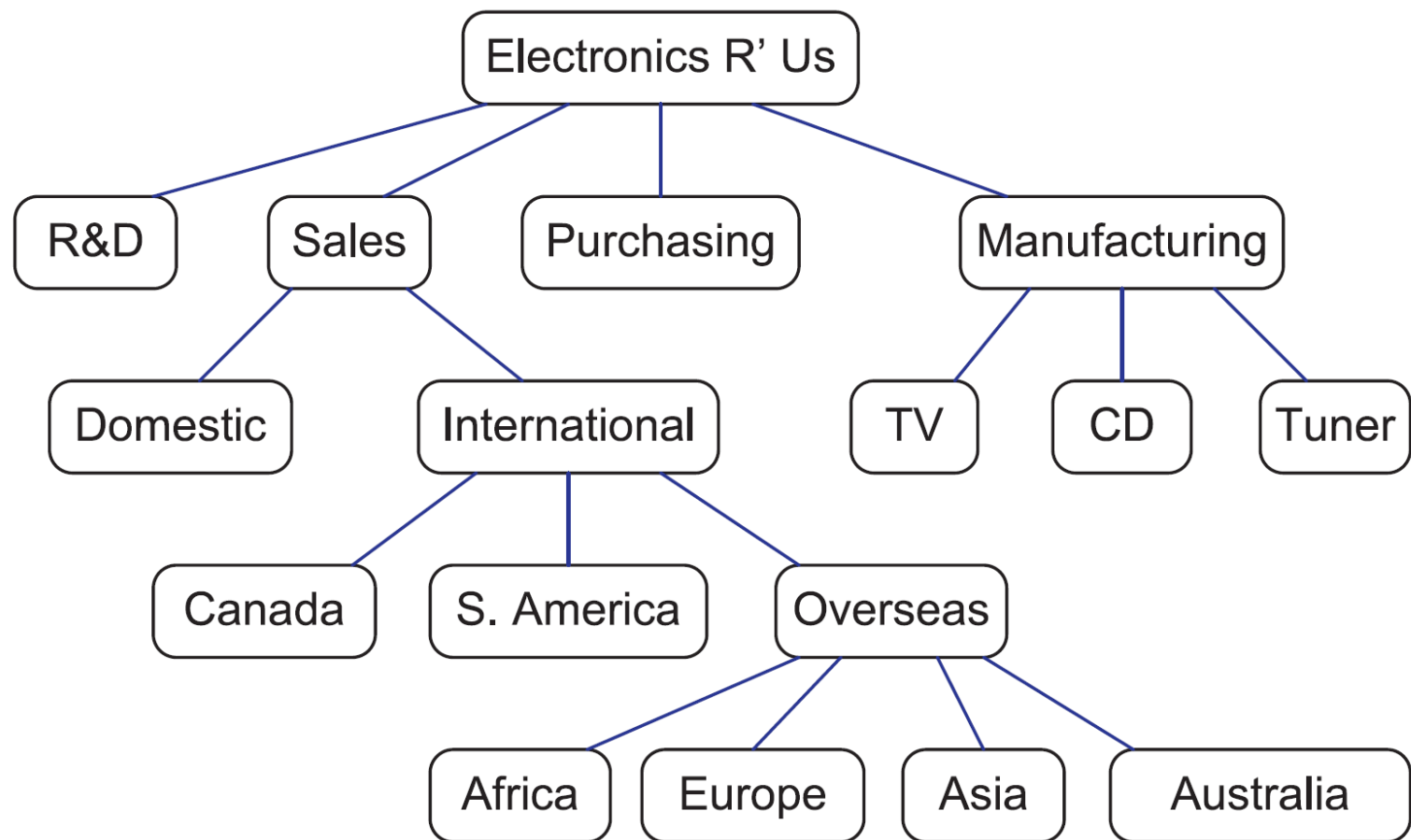
Định nghĩa cây

Cây là một tập nút:

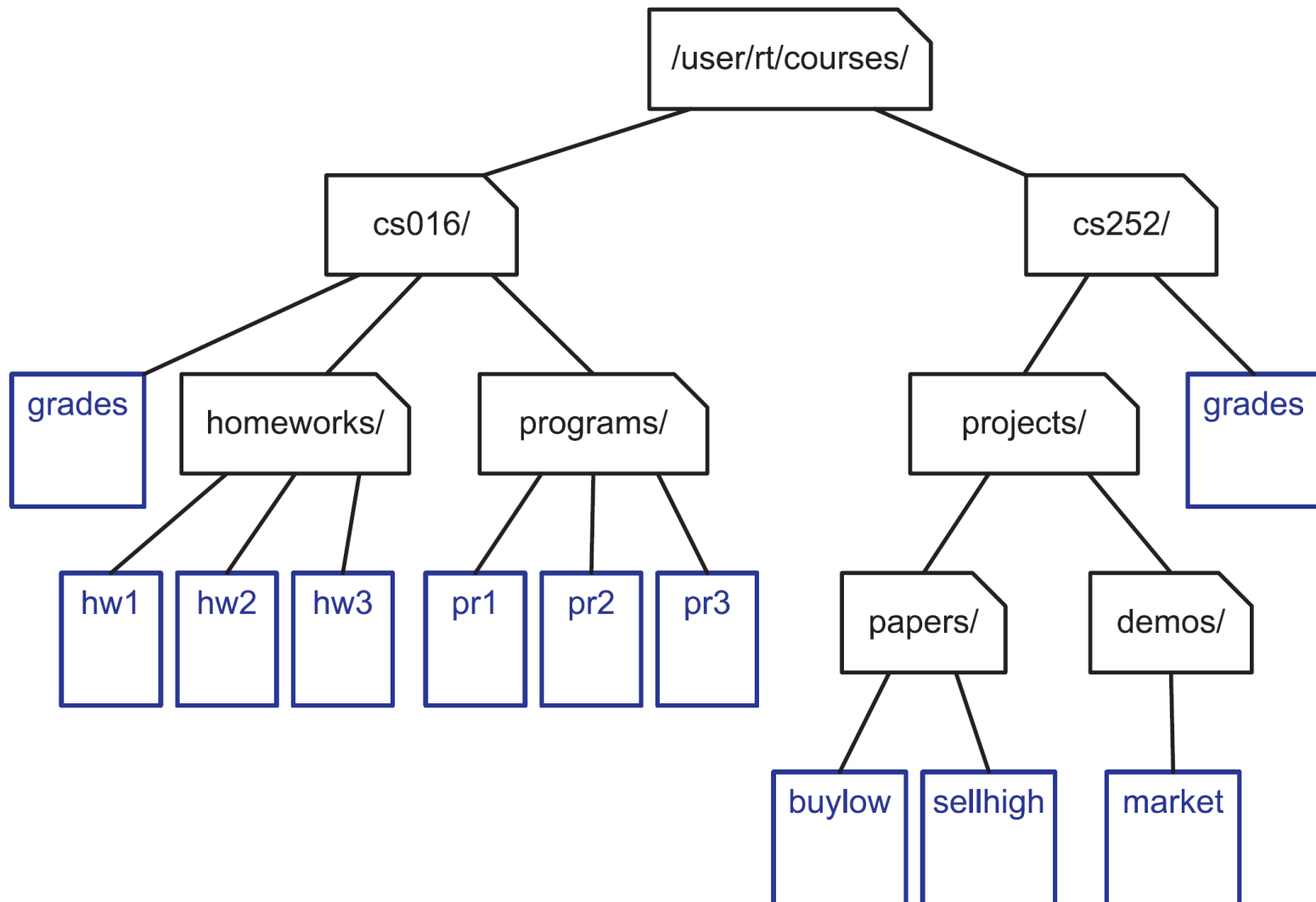
- Nếu tập nút rỗng, đó là cây rỗng.
- Nếu tập nút không rỗng:
 - Có một nút root được gọi là **nút gốc**.
 - Có k **cây con** T_1, T_2, \dots, T_k ($k \geq 0$) sao cho nút gốc của mỗi cây con đó được nối với nút root bằng một **cạnh**.
 - root được gọi là **nút cha**, còn gốc của các cây con T_1, T_2, \dots, T_k được gọi là các **nút con** của root.



Ví dụ 1: Cấu trúc tổ chức của một công ty



Ví dụ 2: Cấu trúc hệ thống file

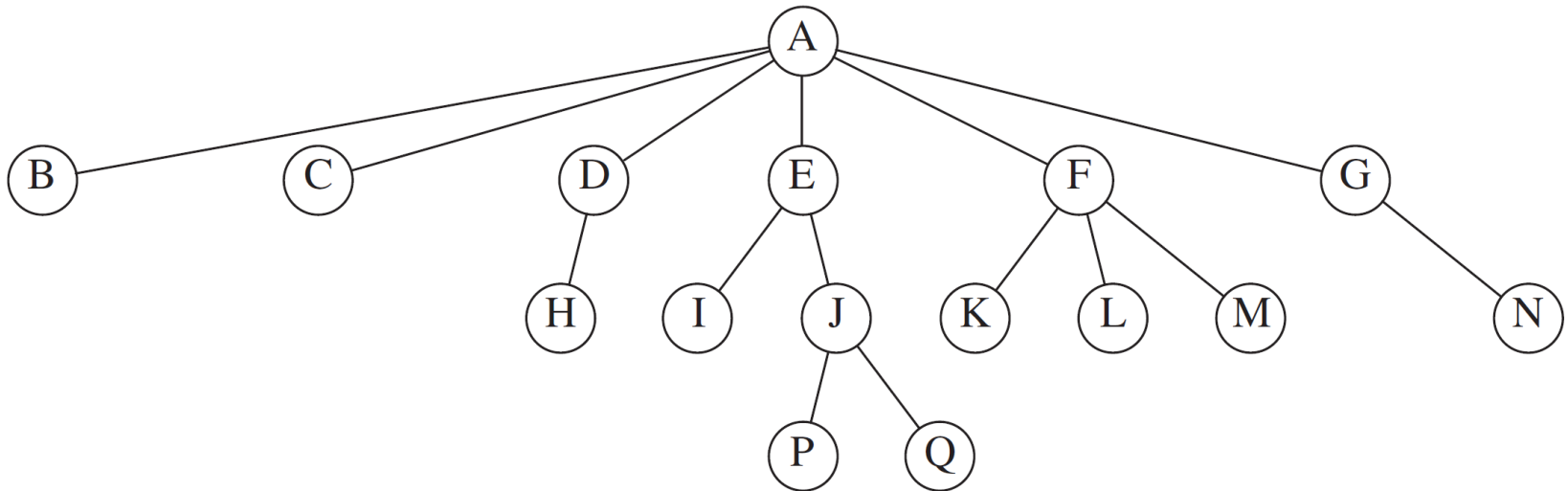


Các khái niệm về cây (1)

Xét một cây có n nút:

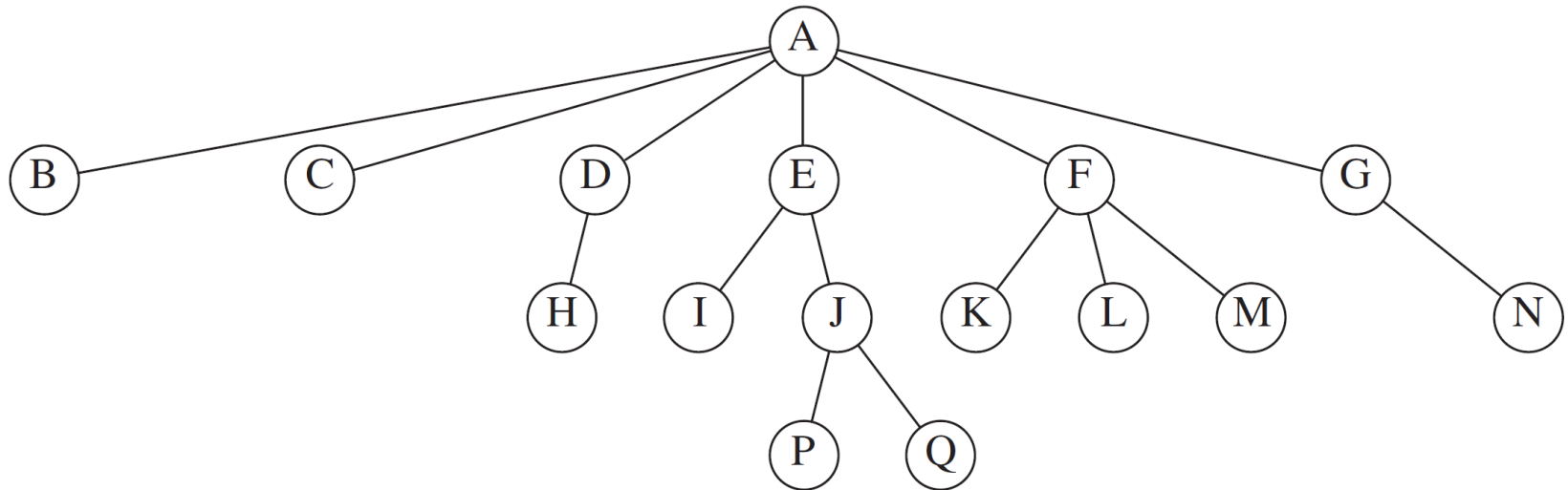
- Có một nút gốc;
- Có $n - 1$ cạnh vì mỗi nút (trừ nút gốc) có một cạnh liên kết nó với nút cha.

Các khái niệm về cây (2)



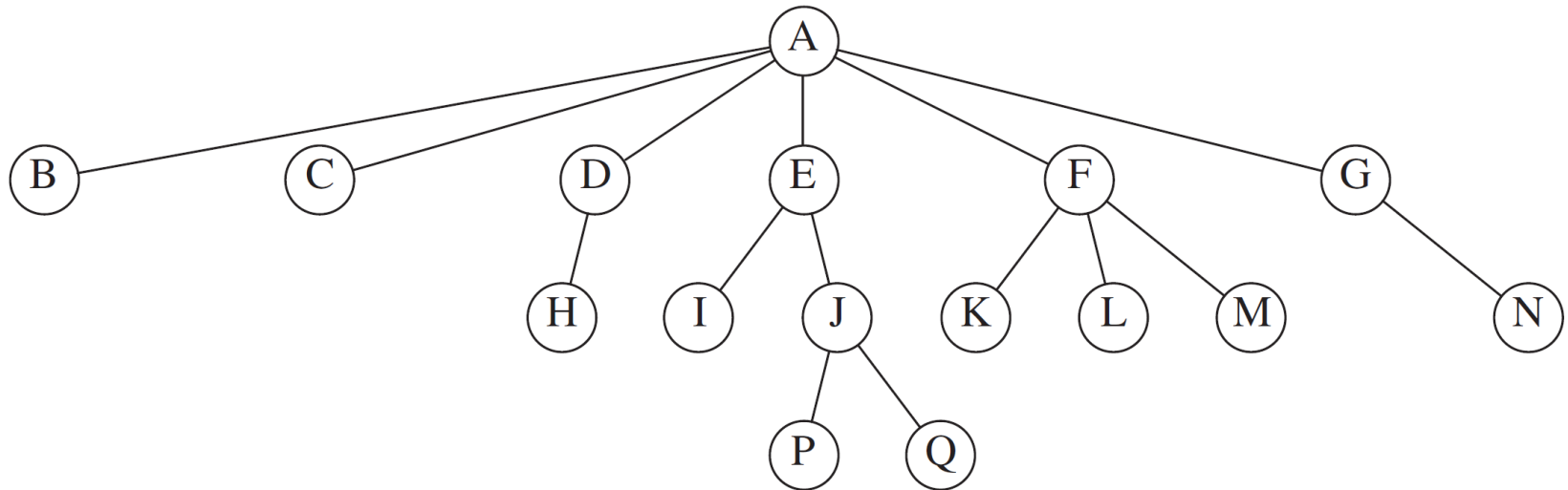
- **Nút lá:** Nút không có con (B, C, H...).
- **Nút anh em:** Các nút cùng cha (K, L, M cùng cha là F).
- **Nút ông (E) và nút cháu (P, Q).**

Các khái niệm về cây (3)



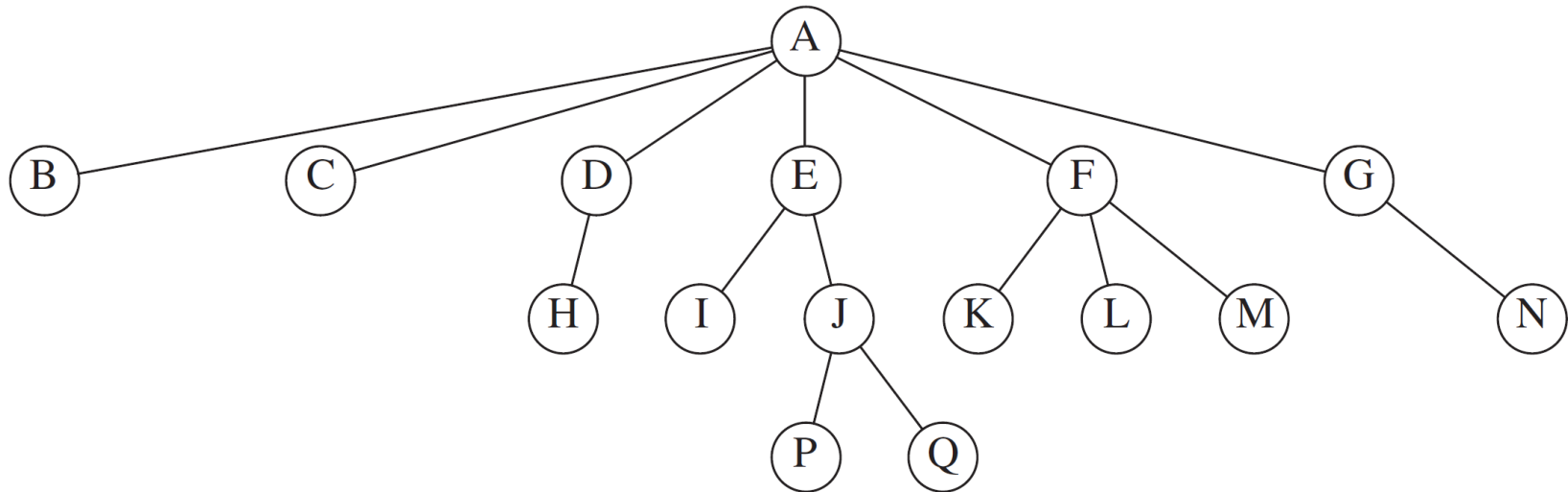
- **Đường đi** từ nút n_1 đến nút n_k là dãy nút n_1, n_2, \dots, n_k trong đó n_i là cha của n_{i+1} ($1 \leq i < k$).
- **Chiều dài đường đi** là số cạnh trên đường đi đó.
 - Đường đi từ một nút tới chính nó có chiều dài bằng 0.

Các khái niệm về cây (4)



- **Chiều sâu** của nút n_i là chiều dài đường đi từ nút gốc đến nút n_i .
 - Nút gốc có chiều sâu 0.
 - Chiều sâu của cây bằng chiều sâu của nút lá sâu nhất.
- **Chiều cao** của nút n_i là chiều dài của đường đi dài nhất từ nút n_i đến một nút lá.
 - Nút lá có chiều cao 0.
 - Chiều cao của cây bằng chiều cao của nút gốc.
- Chiều cao của cây = chiều sâu của cây.

Các khái niệm về cây (5)



- Nếu có đường đi từ nút n_1 đến nút n_2 :
 - Nút n_1 được gọi là **tổ tiên** của nút n_2 , và nút n_2 được gọi là **hậu duệ** của nút n_1 .
 - Nếu $n_1 \neq n_2$ thì ta có các khái niệm tổ tiên thực sự và hậu duệ thực sự.

Cài đặt cây

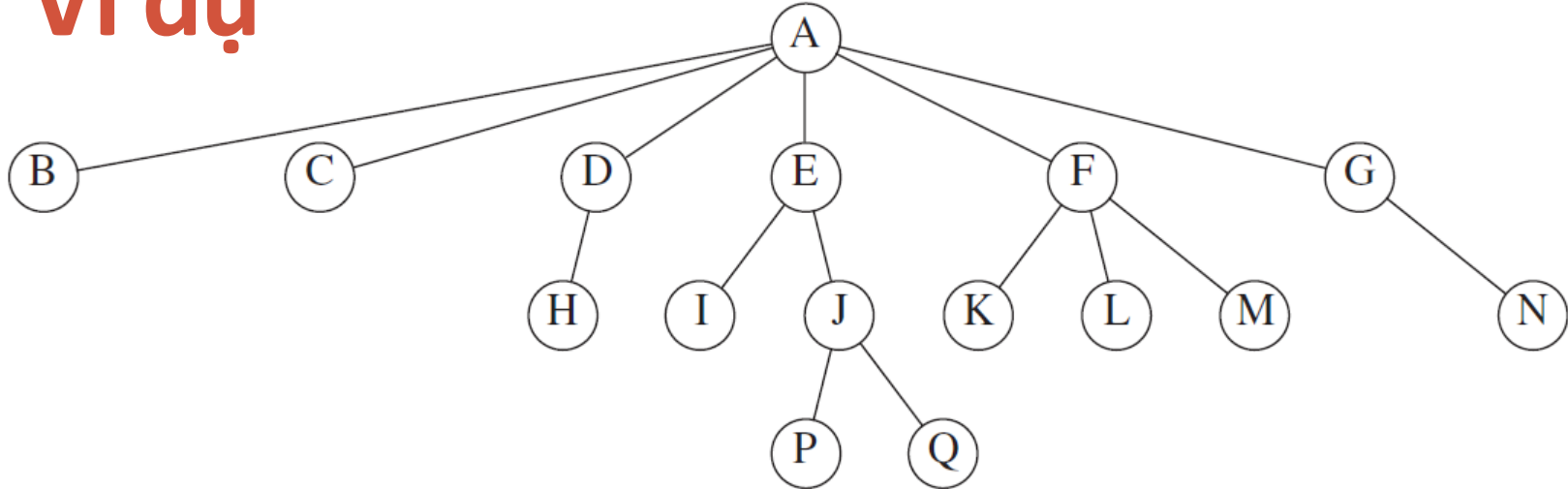
Mỗi nút trong cây chứa:

- Phần tử;
- Con trỏ tới nút con đầu tiên;
- Con trỏ tới nút anh em kế tiếp.

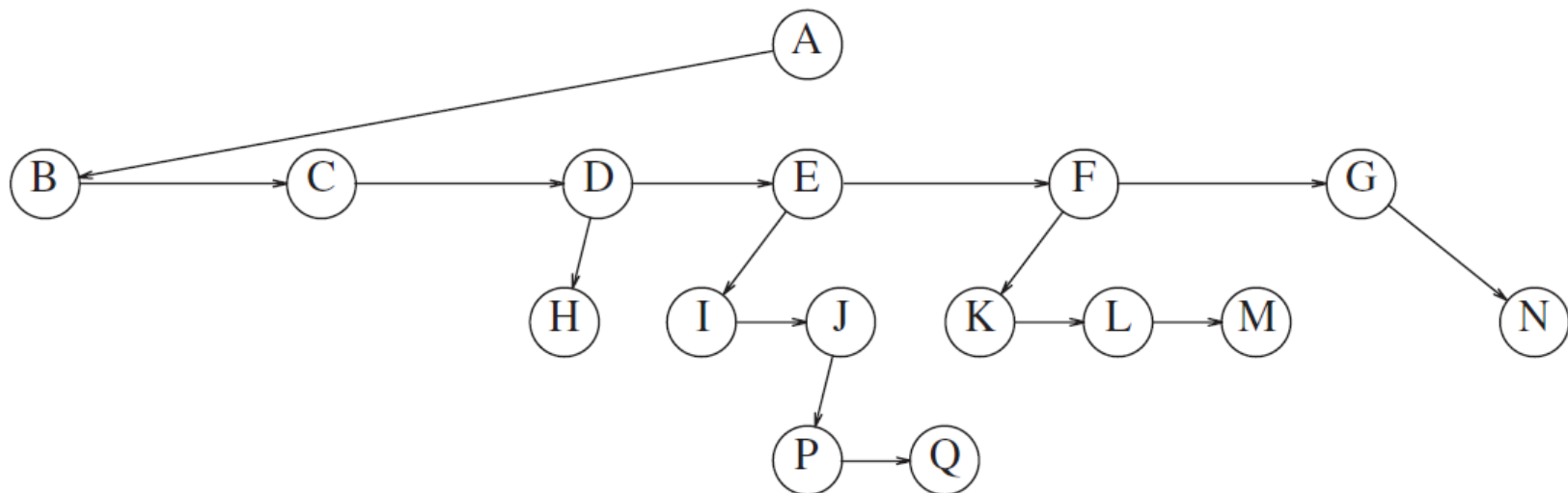
```
struct TreeNode {  
    T elem;  
    TreeNode * firstChild;  
    TreeNode * nextSibling;  
}
```

*Vì sao mỗi nút không
giữ con trỏ tới tất cả
các nút con của nó?*

Ví dụ



Biểu diễn con đầu
tiên/anh em kế tiếp



Duyệt cây

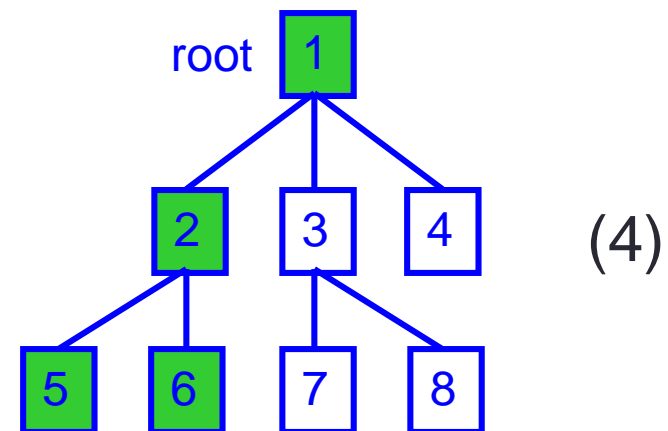
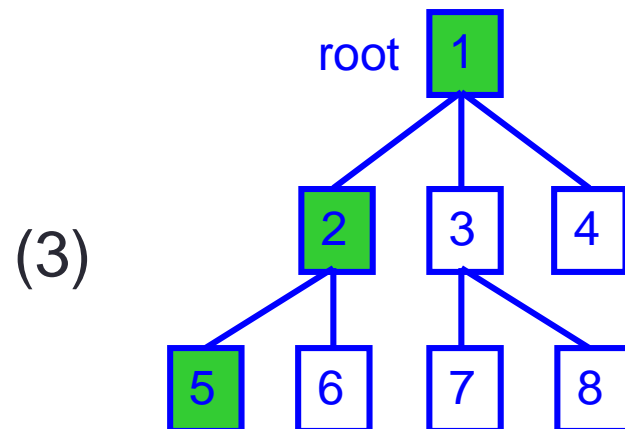
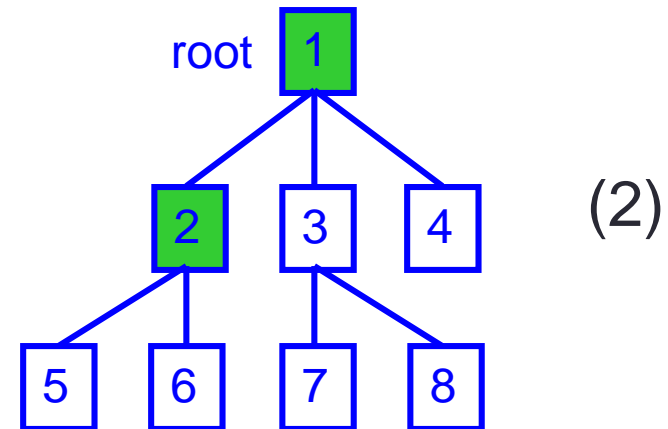
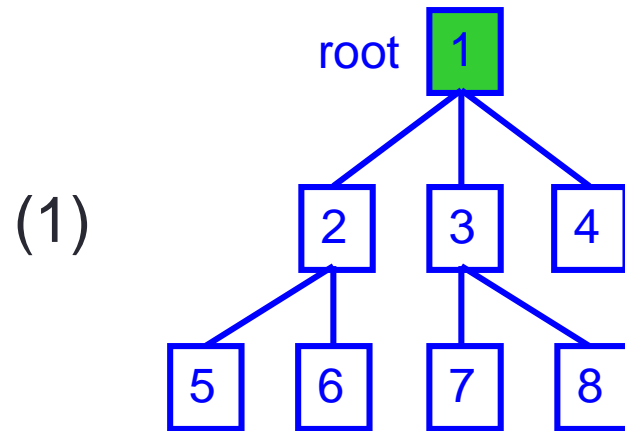
- Là cách đi qua tất cả các nút của cây sao cho mỗi nút chỉ được **thăm** (xử lý) đúng một lần.
 - Thăm có thể là in giá trị trong nút đang xét lên màn hình, hoặc cập nhật giá trị trong nút đó...
- Có 2 cách duyệt chính:
 - Duyệt theo thứ tự trước
 - Duyệt theo thứ tự sau

Duyệt cây theo thứ tự trước

Xuất phát từ nút gốc:

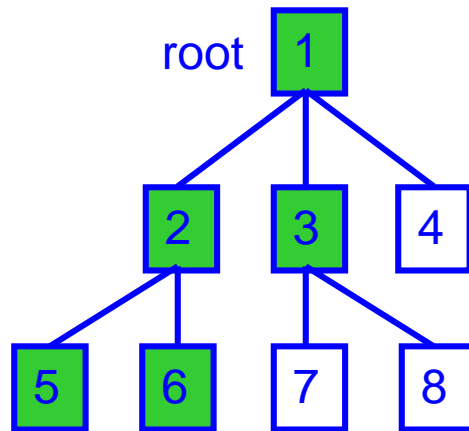
1. Thăm nút đang xét.
2. Duyệt các nút con của nút đang xét từ trái sang phải theo thứ tự trước (gọi đệ quy).

Duyệt cây theo thứ tự trước

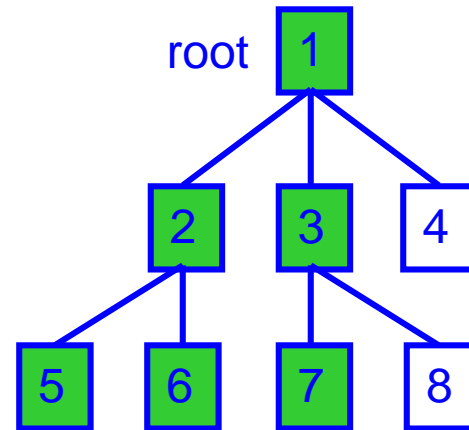


Duyệt cây theo thứ tự trước

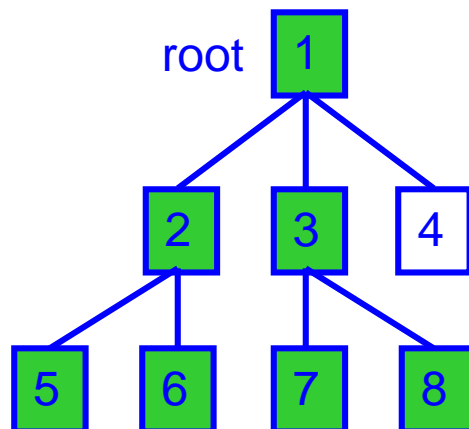
(5)



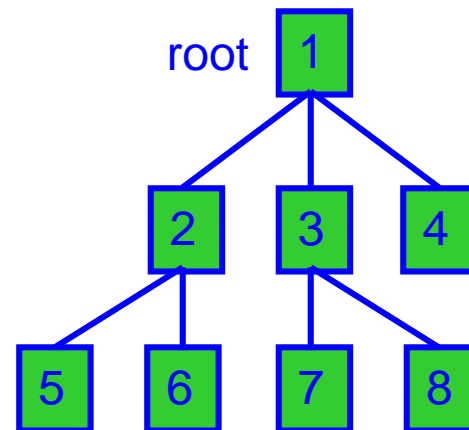
(6)



(7)



(8)



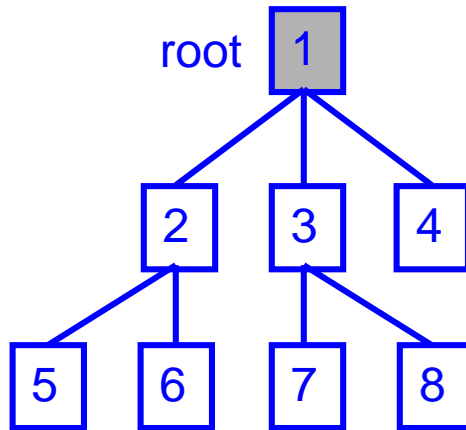
Duyệt cây theo thứ tự sau

Xuất phát từ nút gốc:

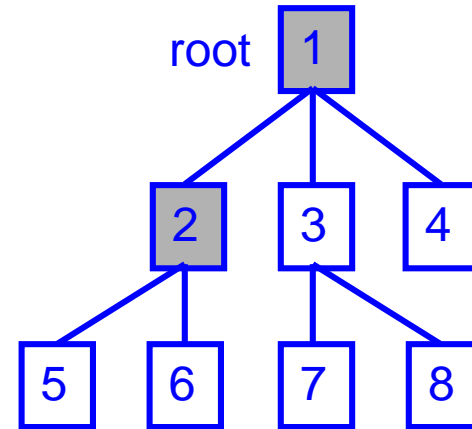
1. Duyệt các nút con của nút đang xét từ trái sang phải theo thứ tự sau (gọi đệ quy).
2. Thăm nút đang xét.

Duyệt cây theo thứ tự sau

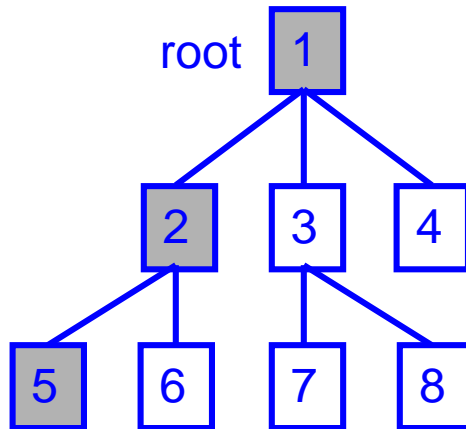
(1)



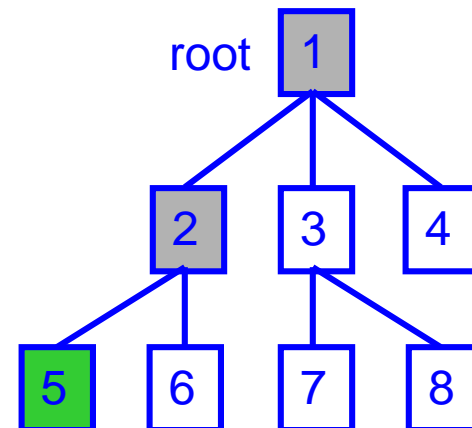
(2)



(3)

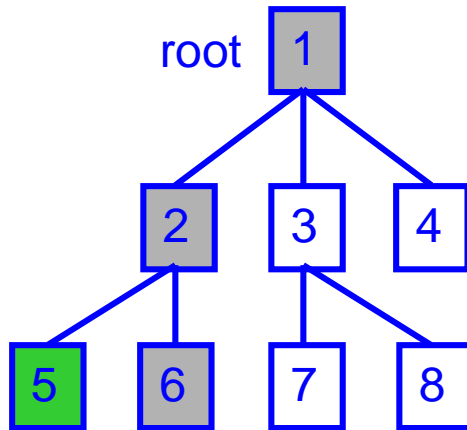


(4)

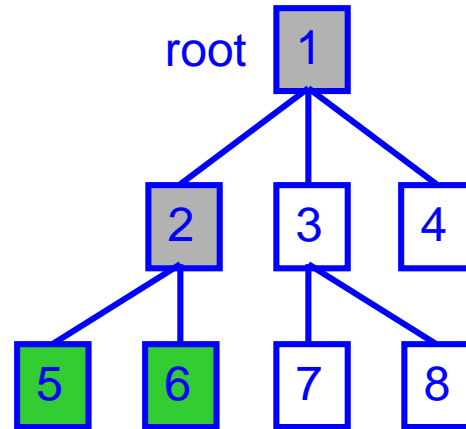


Duyệt cây theo thứ tự sau

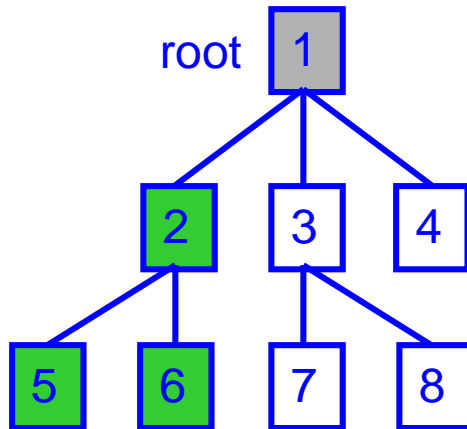
(5)



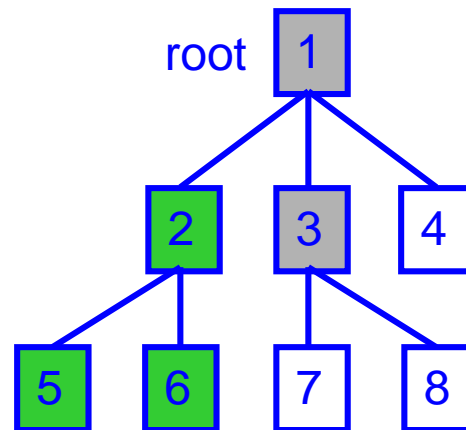
(6)



(7)

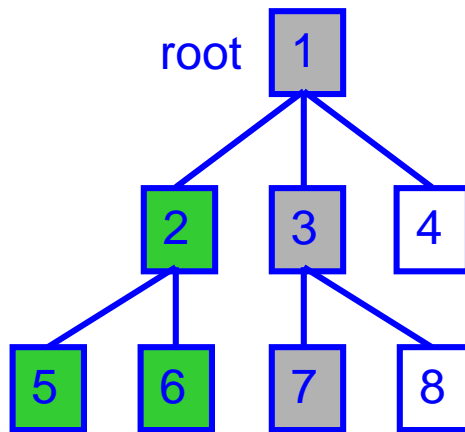


(8)

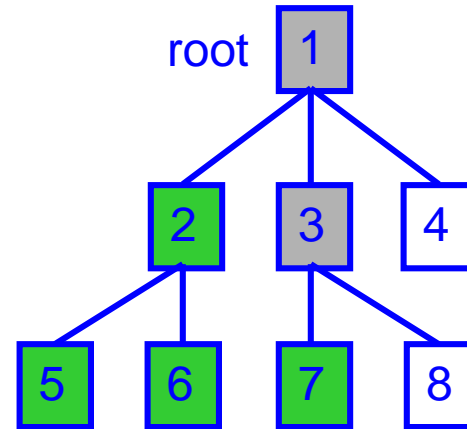


Duyệt cây theo thứ tự sau

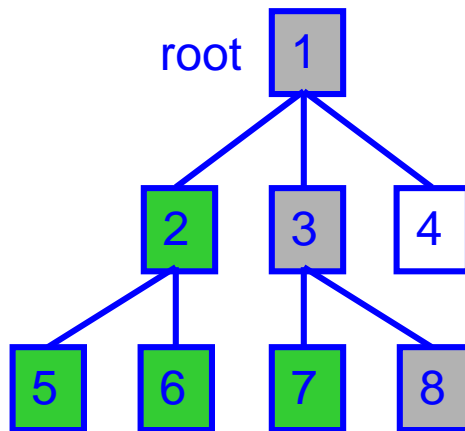
(9)



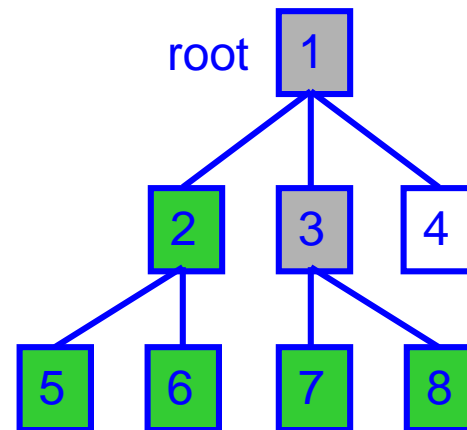
(10)



(11)

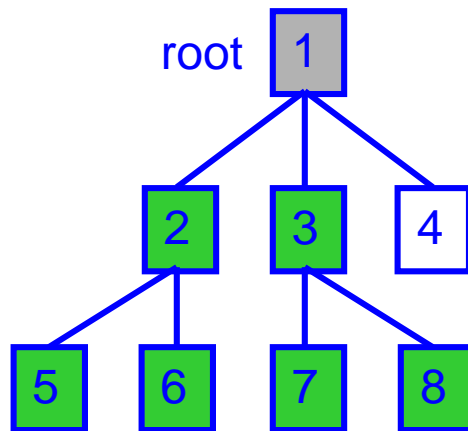


(12)

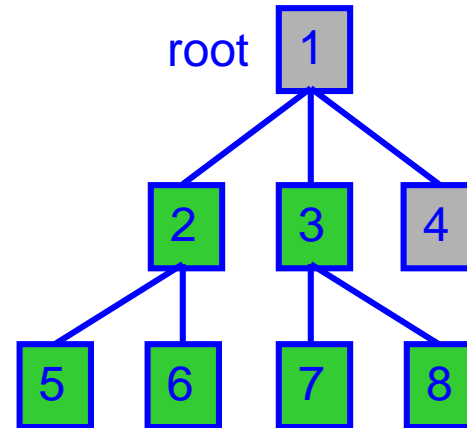


Duyệt cây theo thứ tự sau

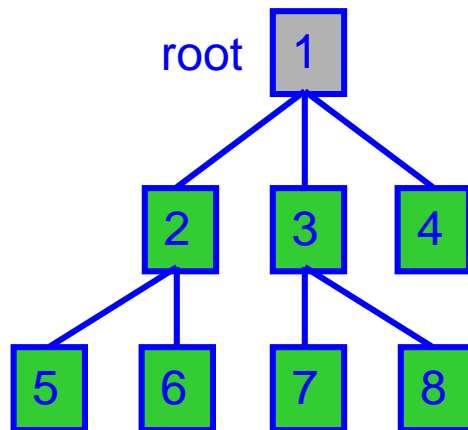
(13)



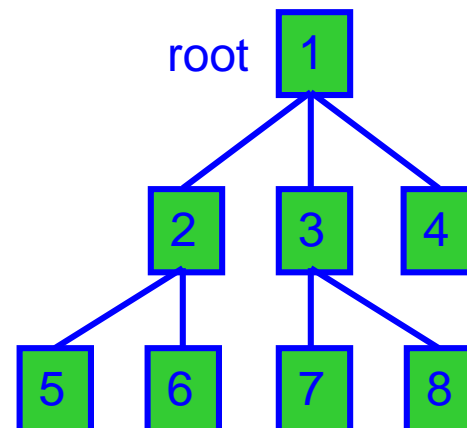
(14)



(15)



(16)

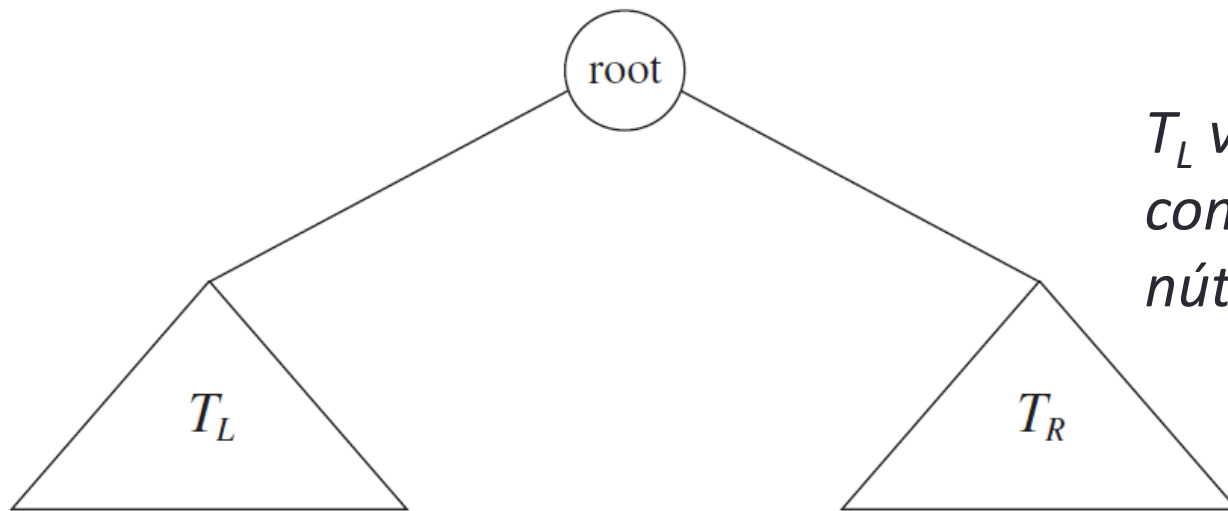


2. Cây nhị phân

Định nghĩa cây nhị phân

Cây nhị phân là cây, trong đó mỗi nút có không quá 2 con, phân thành **con trái** và **con phải**:

- Con trái chính là gốc của **cây con trái** của nút đang xét.
- Con phải chính là gốc của **cây con phải** của nút đang xét.



T_L và T_R là hai cây con trái và phải của nút gốc root.

Cài đặt cây nhị phân

Mỗi nút trong cây nhị phân chứa:

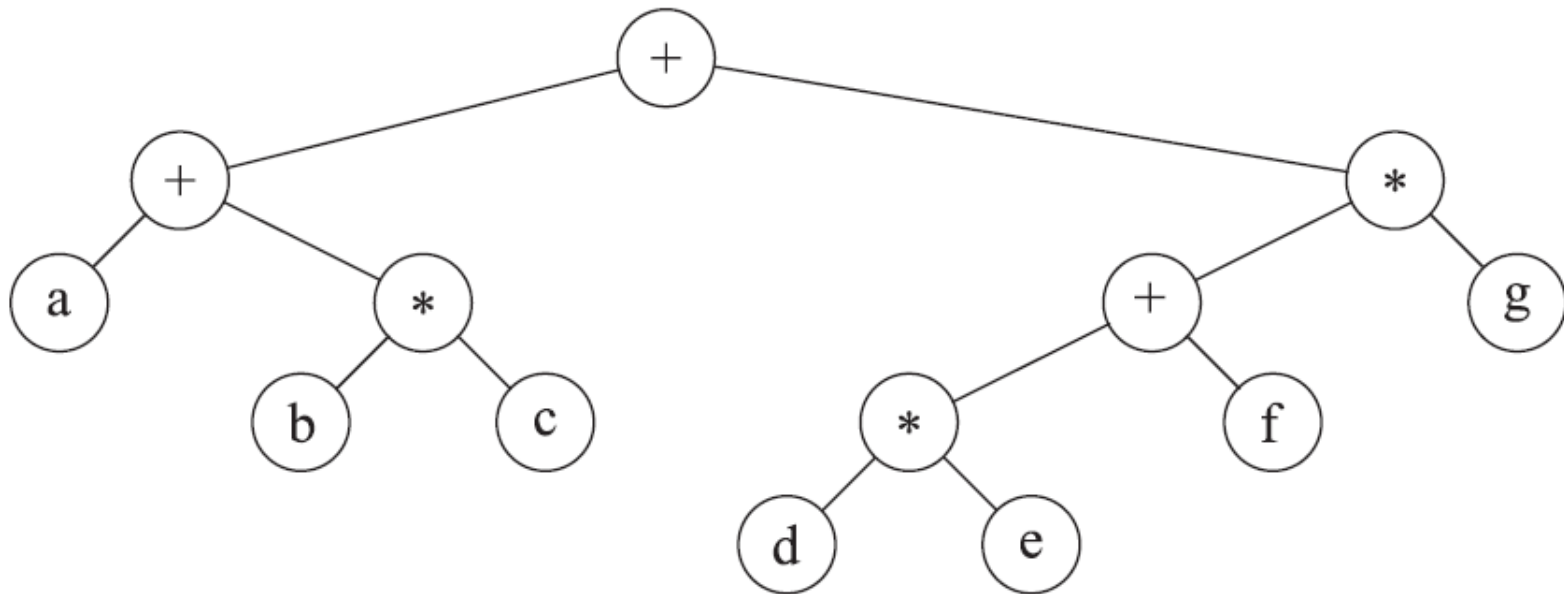
- Phần tử;
- Con trỏ tới nút con trái (có thể rỗng);
- Con trỏ tới nút con phải (có thể rỗng).

```
struct BinaryNode {  
    T elem;  
    BinaryNode * left;  
    BinaryNode * right;  
}
```

Cây biểu thức

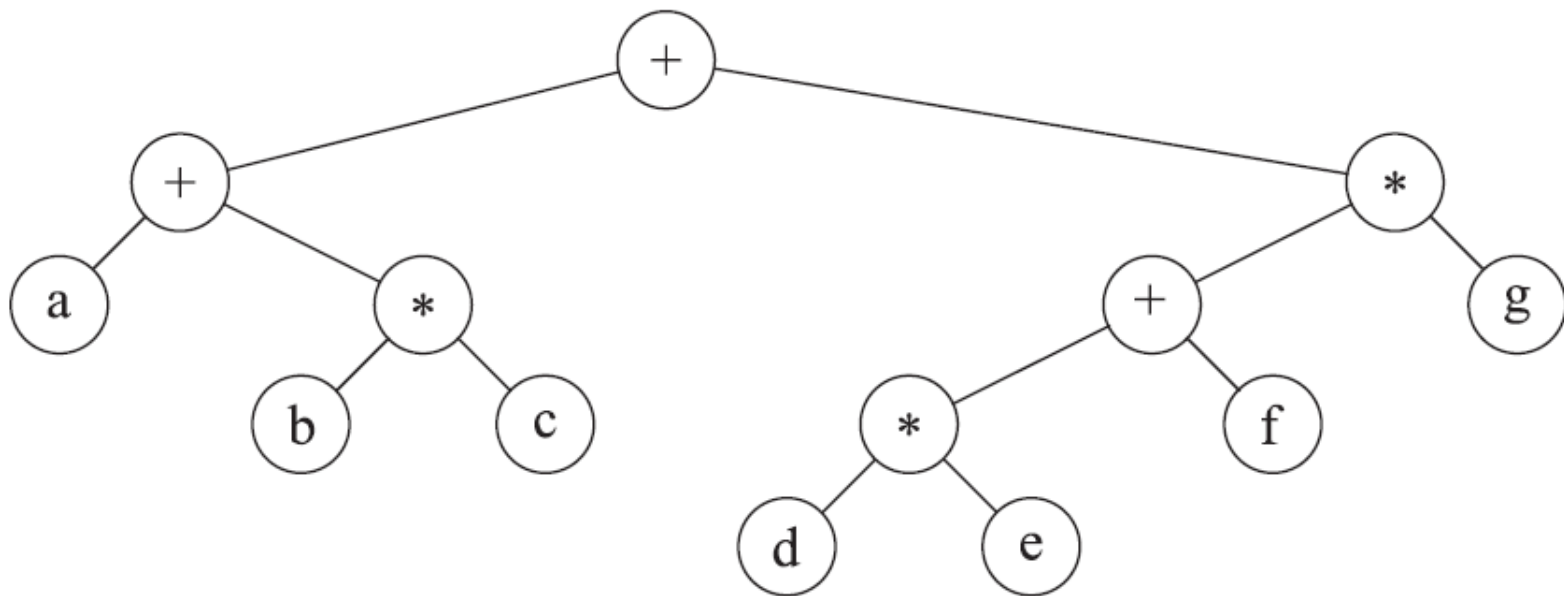
Cây biểu thức là một cây nhị phân, trong đó:

- Nút trong lưu trữ toán tử;
- Nút lá lưu trữ toán hạng.



Duyệt cây biểu thức theo thứ tự giữa

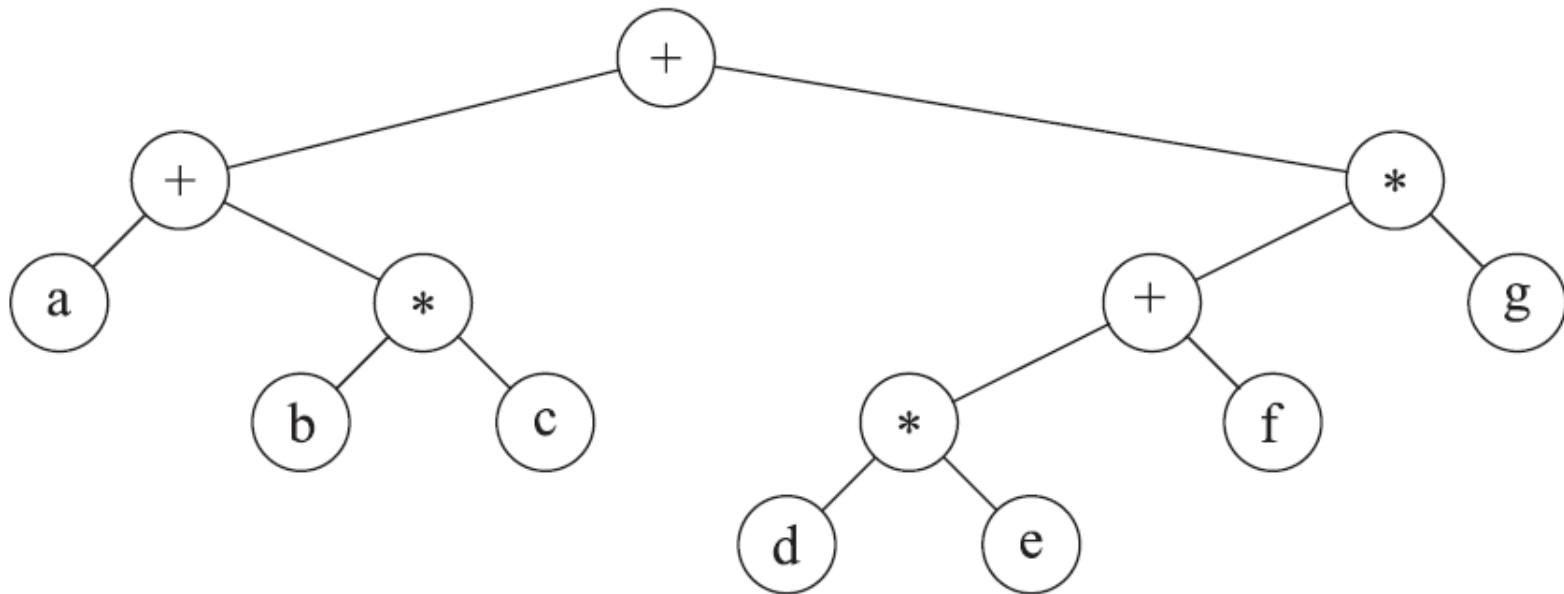
- Xuất phát từ nút gốc.
- Trình tự duyệt: duyệt con trái theo thứ tự giữa \rightarrow thăm nút đang xét \rightarrow duyệt con phải theo thứ tự giữa.
- Đầu ra: biểu thức trung tố.



$(a + b * c) + ((d * e + f) * g)$

Duyệt cây biểu thức theo thứ tự sau

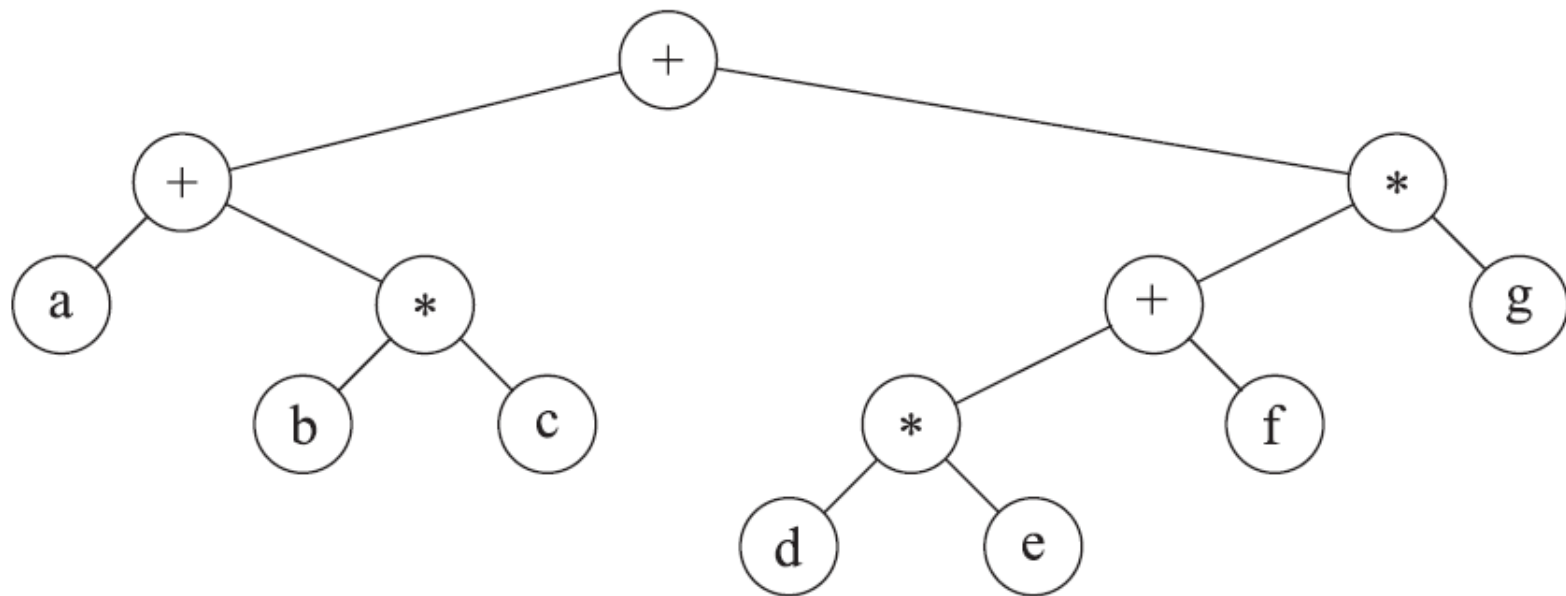
- Xuất phát từ nút gốc.
- Trình tự duyệt: duyệt con trái theo thứ tự sau \rightarrow duyệt con phải theo thứ tự sau \rightarrow thăm nút đang xét.
- Đầu ra: biểu thức hậu tố.



a b c * + d e * f + g * +

Duyệt cây biểu thức theo thứ tự trước

- Xuất phát từ nút gốc.
- Trình tự: thăm nút đang xét \rightarrow duyệt con trái theo thứ tự trước \rightarrow duyệt con phải theo thứ tự trước.
- Đầu ra: biểu thức tiền tố.



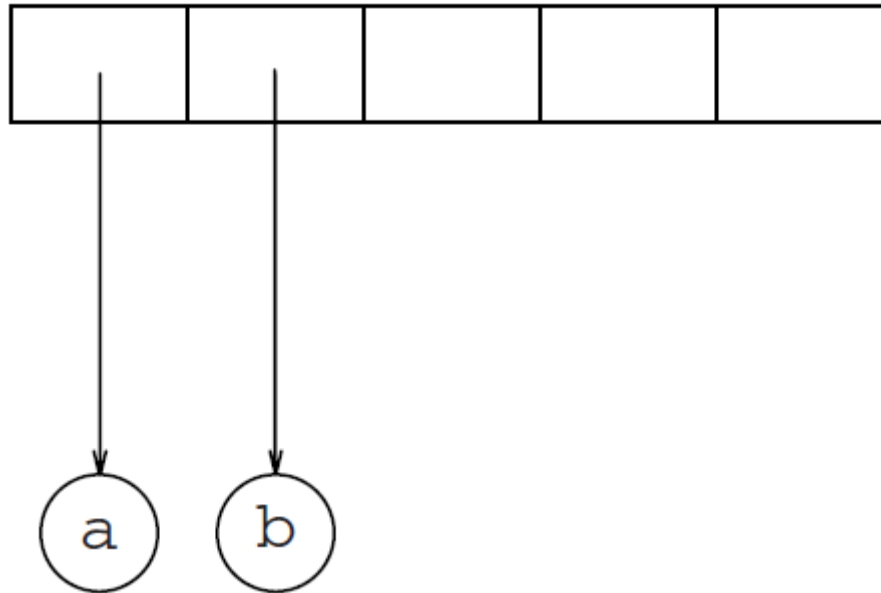
+ + a * b c * + * d e f g

Xây dựng cây biểu thức

- Xét trường hợp biểu thức hậu tố.
(Ví dụ: $a \ b \ + \ c \ d \ e \ + \ * \ *$)
- Cách xây dựng (dùng ngăn xếp):
 - Đọc từng toán hạng/toán tử từ trái sang phải.
 - Gặp toán hạng: (1) Tạo nút chứa toán hạng; (2) Đặt con trỏ tới nút đó vào ngăn xếp.
 - Gặp toán tử: (1) Lấy hai con trỏ, đang trỏ tới hai cây T_1 và T_2 , ra khỏi ngăn xếp; (2) Tạo nút chứa toán tử và trỏ sang trái và phải tới T_1 và T_2 ; (3) Đặt con trỏ tới nút toán tử vào ngăn xếp.

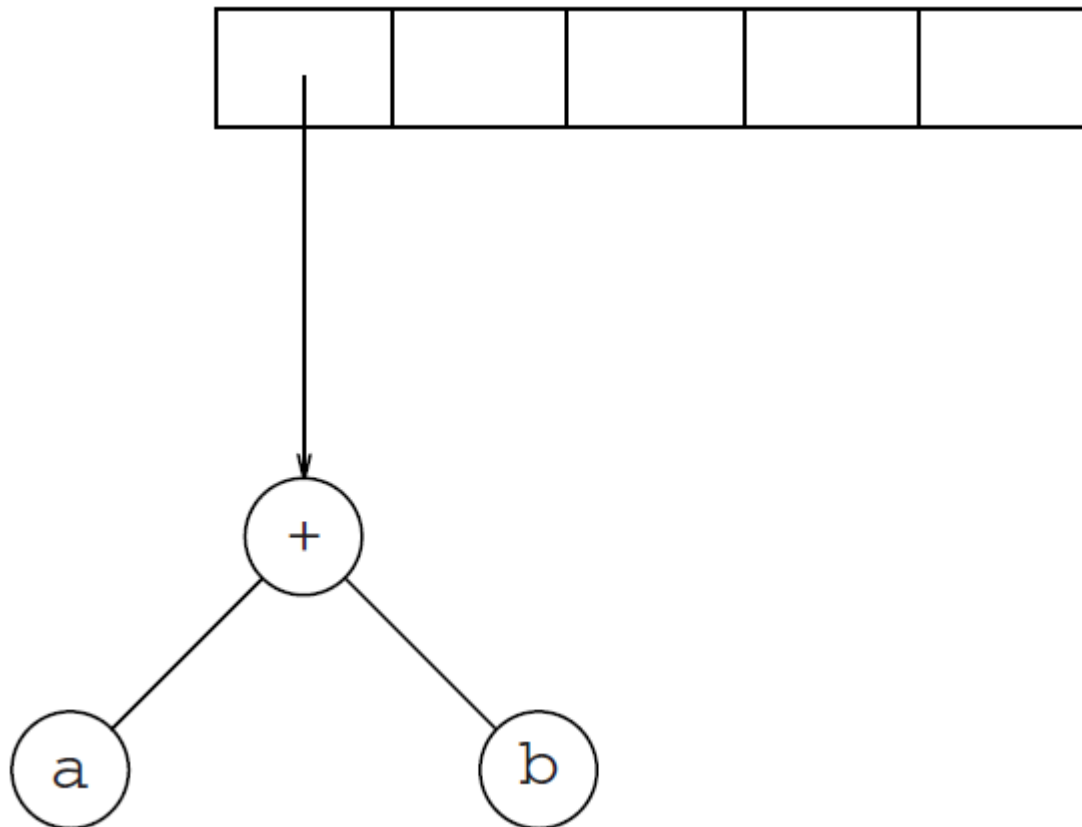
Xây dựng cây: $a b + c d e + * *$

- Đọc a, b



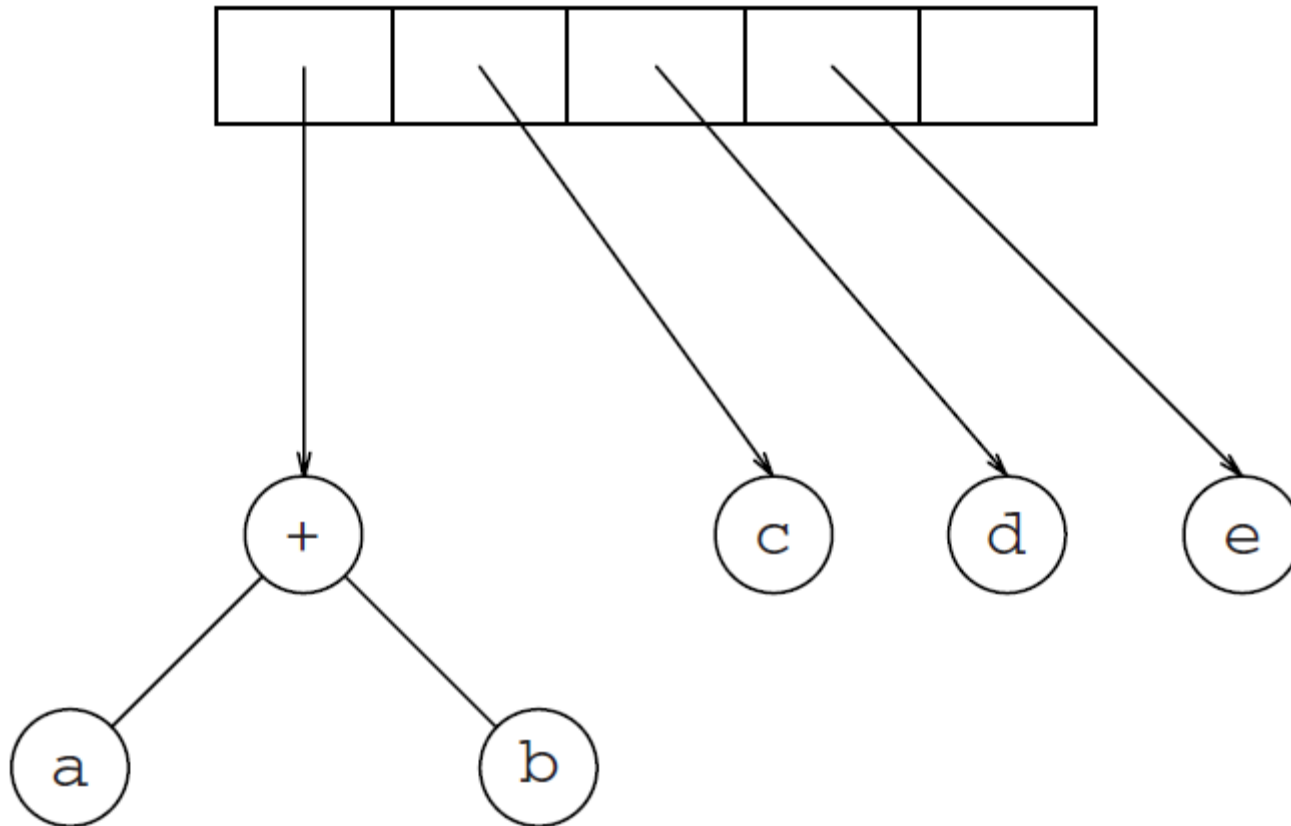
Xây dựng cây: $a b + c d e + * *$

- Đọc +



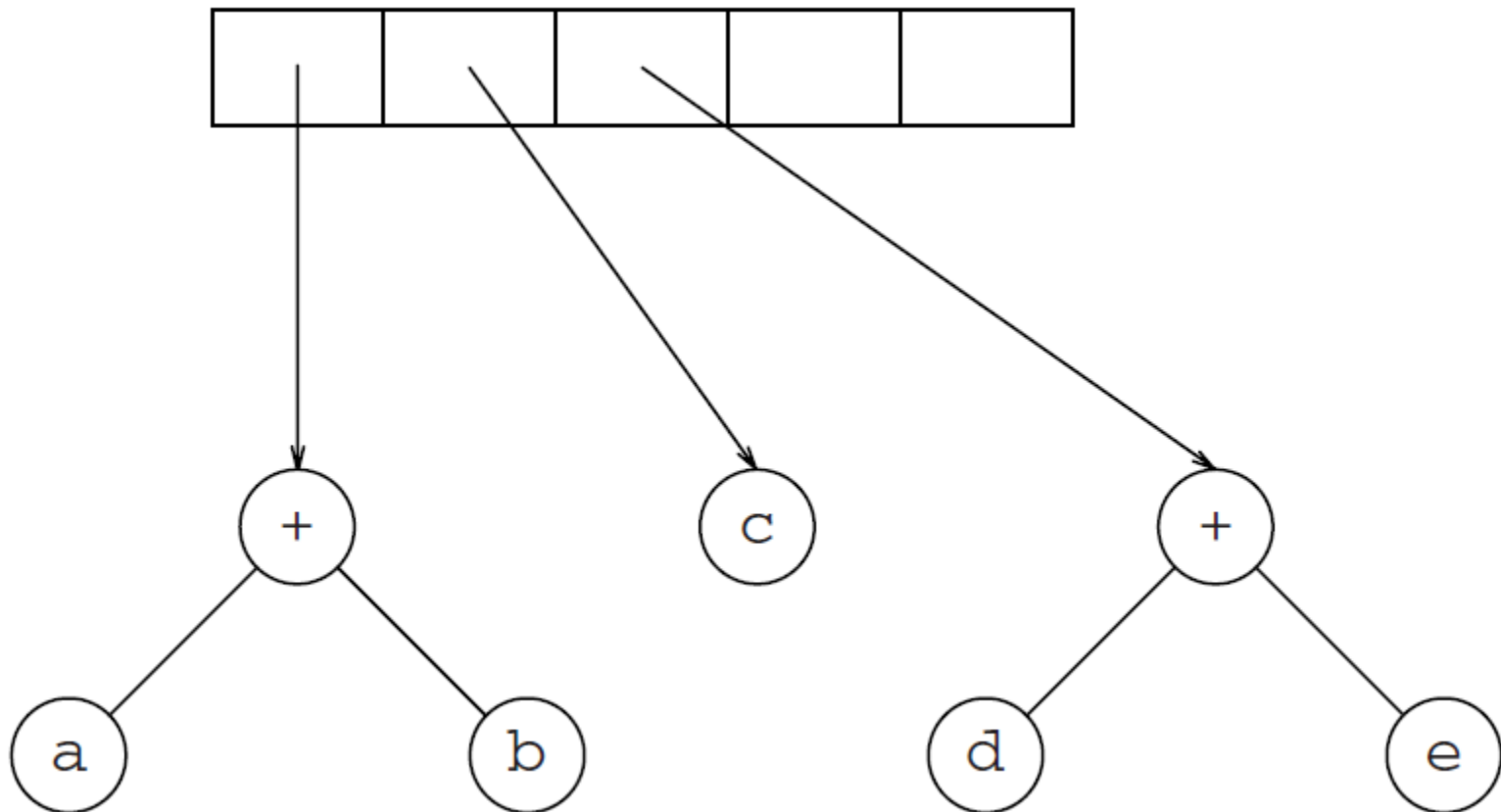
Xây dựng cây: $a b + c d e + * *$

- Đọc c, d, e



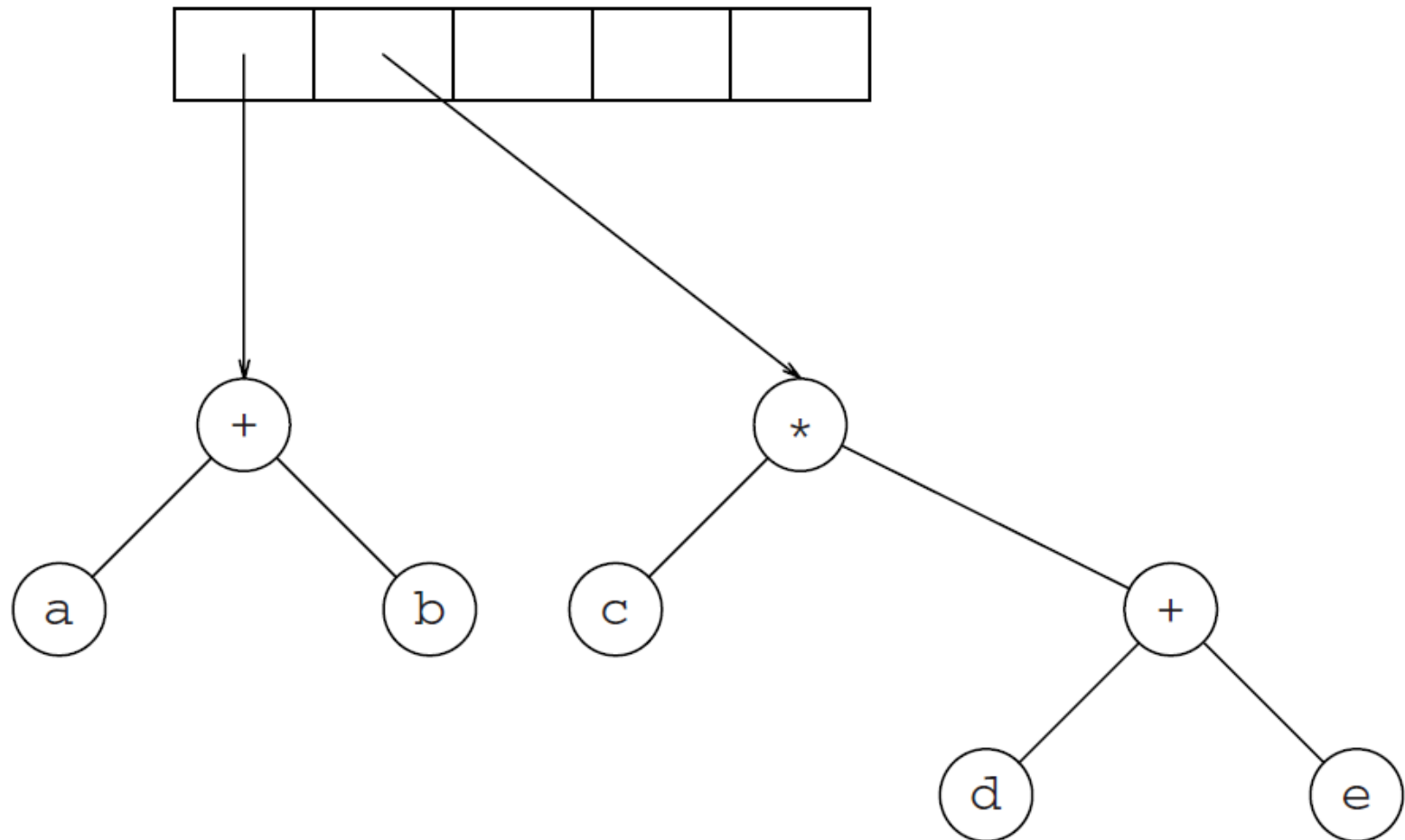
Xây dựng cây: $a b + c d e + * *$

- Đọc +



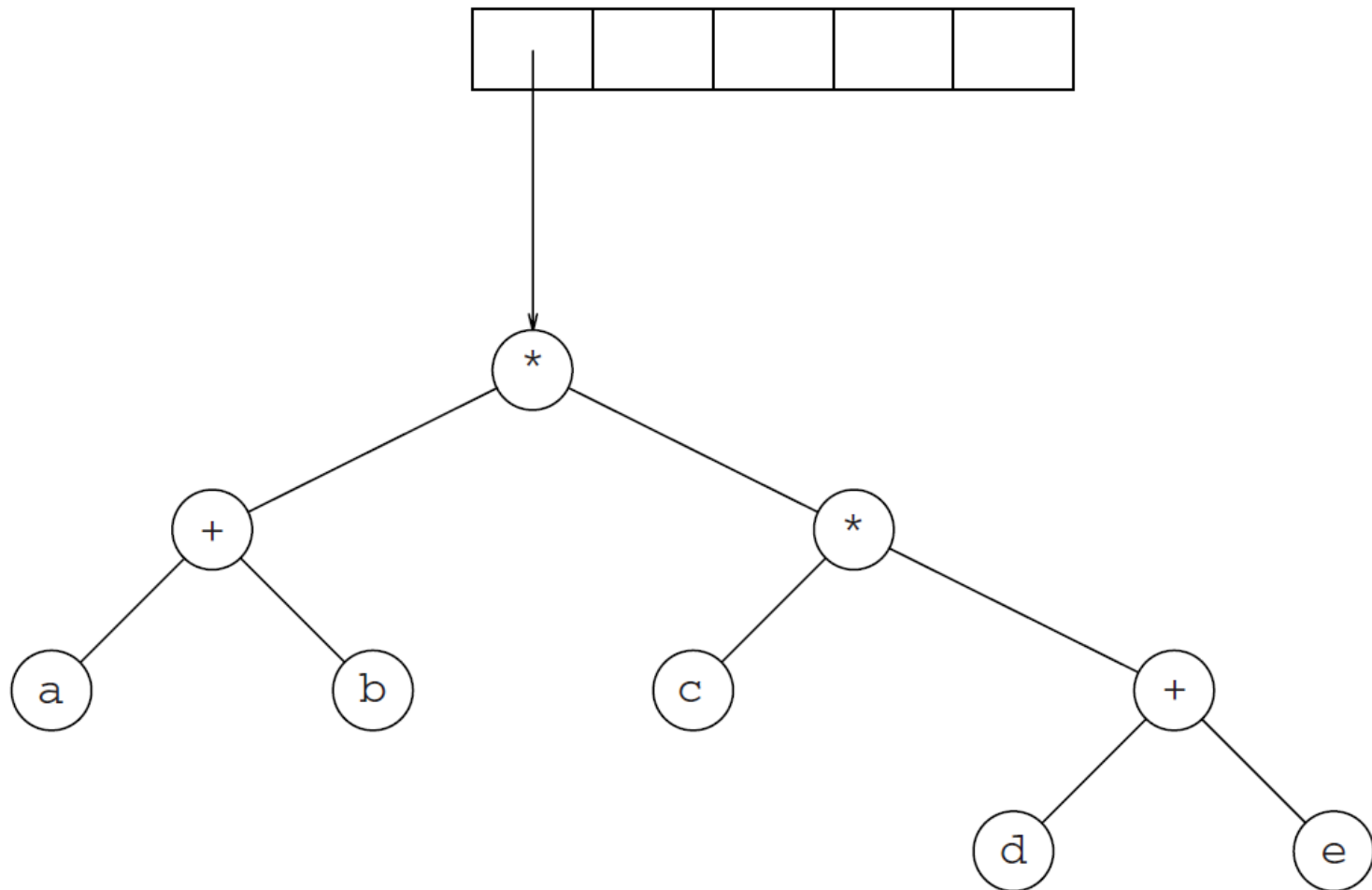
Xây dựng cây: $a b + c d e + * *$

- Đọc *



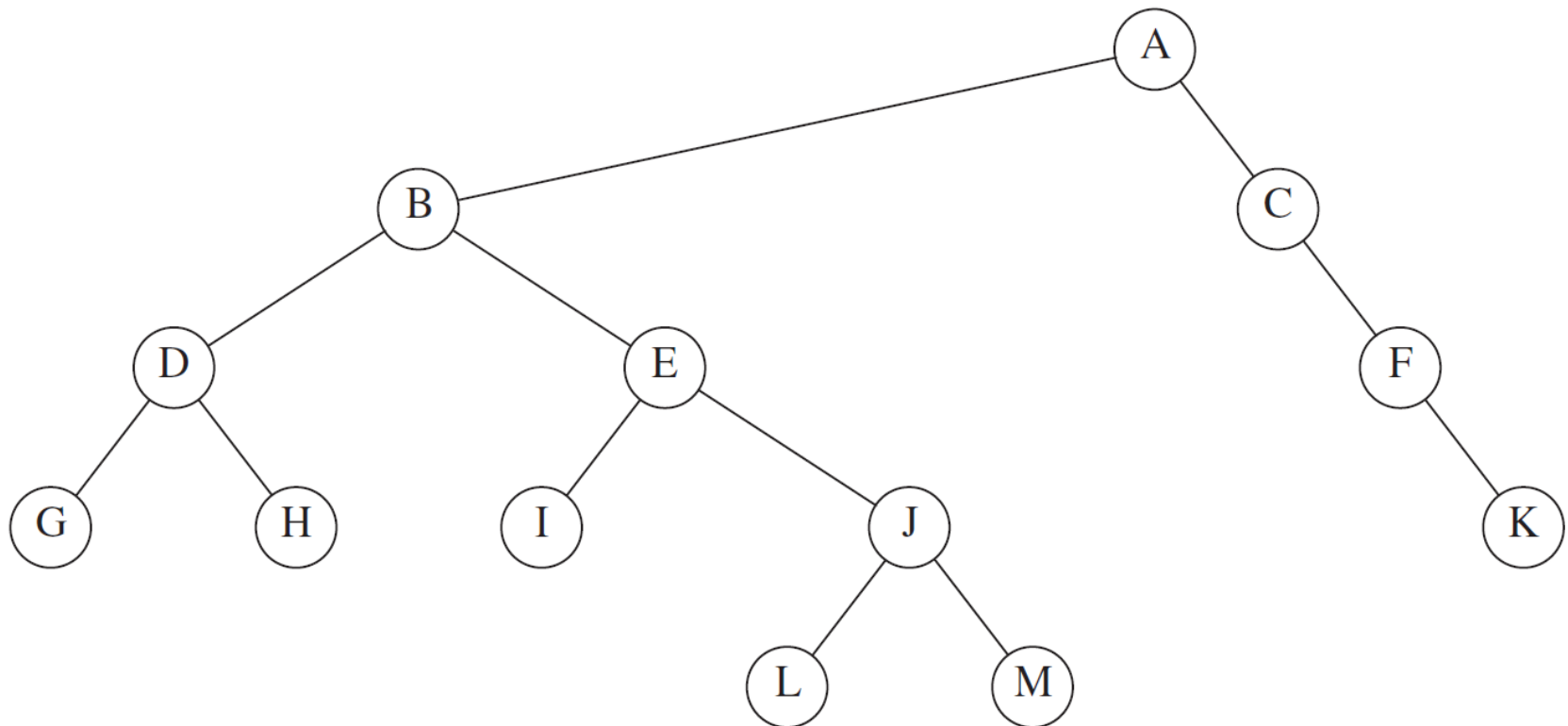
Xây dựng cây: $a b + c d e + * *$

- Đọc *



Bài tập

1. Chỉ ra nút gốc và các nút lá trên cây. Tính chiều sâu và chiều cao của các nút C, E và H. Tính chiều sâu và chiều cao của cây.

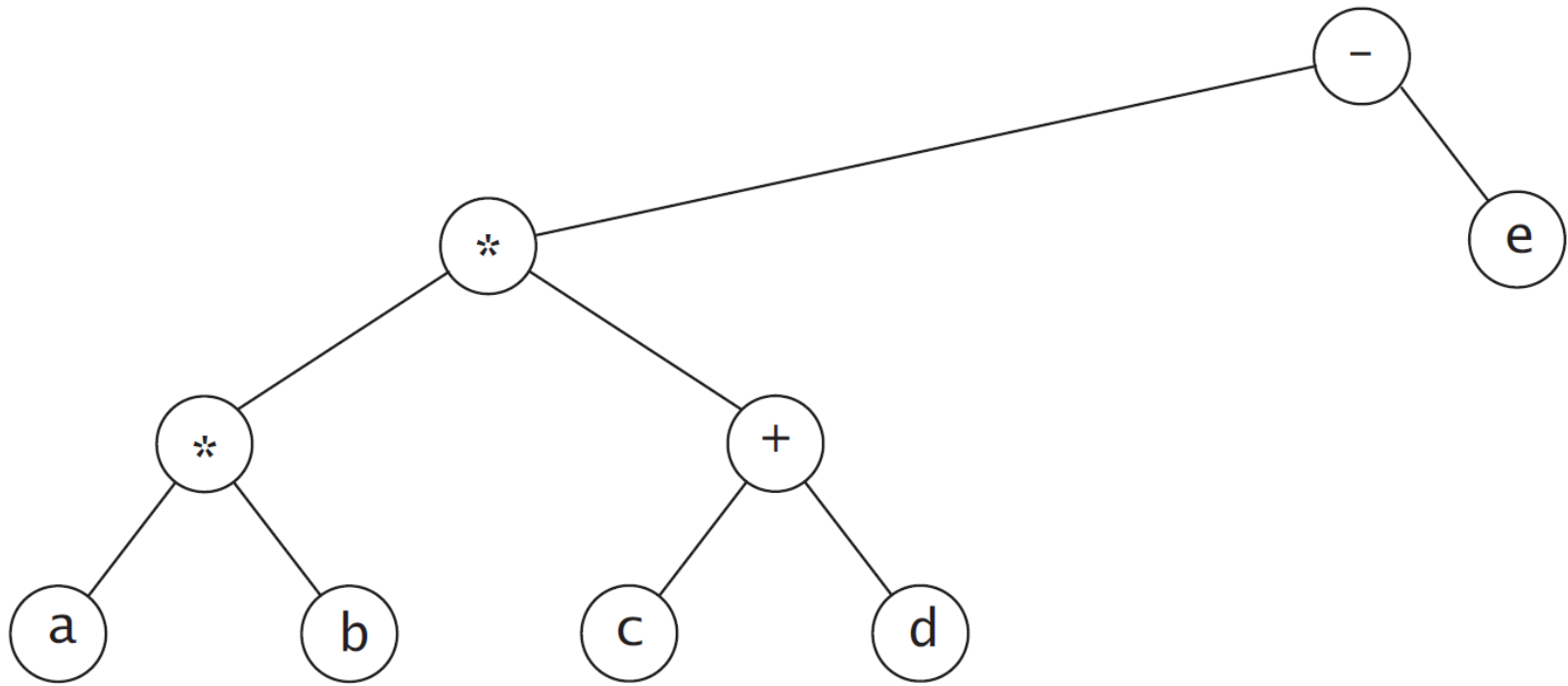


Bài tập

2. Xét cây nhị phân có n nút. Chứng minh rằng có $n + 1$ liên kết rỗng (tức là con trỏ NULL) trên cây.
3. Xét cây nhị phân có chiều cao h . Chứng minh rằng số nút trên cây không vượt quá $2^{h+1} - 1$. *Gợi ý: Dùng quy nạp toán học.*

Bài tập

4. Viết các biểu thức trung tố, tiền tố và hậu tố tương ứng với cây biểu thức bên dưới.



Bài tập

5. Viết hàm C++ có một tham số duy nhất là con trỏ tới gốc của một cây nhị phân T để:

- (a) đếm số nút của T.
- (b) đếm số nút lá của T.
- (c) đếm số nút có đủ cả 2 con của T.

Gợi ý: Dùng đệ quy.

6. Viết các hàm C++ duyệt cây nhị phân theo thứ tự trước, giữa và sau. *Gợi ý: Dùng đệ quy.*

7. Nêu cách thực hiện phép duyệt cây nhị phân theo **thứ tự mức**: thăm nút gốc trước tiên, rồi đến thăm các nút ở độ sâu 1, rồi đến thăm các nút ở độ sâu 2, v.v... *Gợi ý: Dùng hàng đợi.*