Cấu trúc dữ liệu và Giải thuật

Chương IV: Cấu trúc Cây

Cấu trúc Cây

- Nội dung
  1. Các khái niệm
  2. Cây tổng quát
     1. ADT Cây
     2. Biểu diễn cây tổng quát
     3. Duyệt cây tổng quát
  3. Cây nhị phân
     1. Định nghĩa và tính chất
     2. Duyệt cây nhị phân
     3. Biểu diễn cây nhị phân
  4. Ứng dụng của cấu trúc cây cho cây biểu thức

Đỗ Bích Diệp - Khoa CNTT

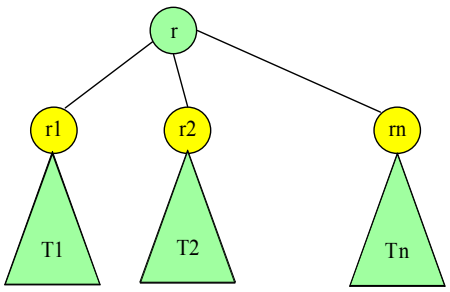
Định nghĩa Cây

- Cây là một cấu trúc phi tuyến, thiết lập trên một tập hữu hạn các "nút"
  - Tồn tại một nút đặc biệt gọi là "gốc" (root)
  - Giữa các nút tồn tại một quan hệ phân cấp hay gọi là quan hệ cha con
  - Một nút trừ nút gốc chỉ có một cha
  - Một nút có thể có từ 0 đến n con

Đỗ Bích Diệp - Khoa CNTT

Định nghĩa Cây

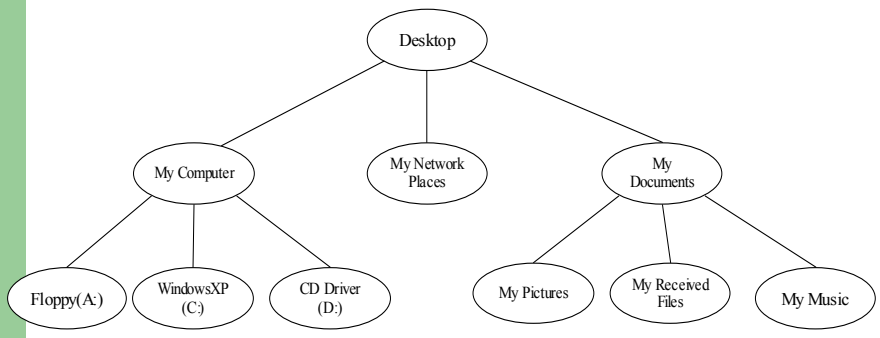
- Định nghĩa đệ quy về Cây
  - Một nút tạo thành một cây.
  - Nếu có n cây $T_1, T_2, \ldots, T_n$ tách biệt có các nút gốc lần lượt là $r_1, r_2, \ldots, r_n$; r là một nút có quan hệ cha-con với $r_1, r_2, \ldots, r_n$ thì tồn tại một cây mới T nhận r làm gốc.



Đỗ Bích Diệp - Khoa CNTT

Ví dụ Cây

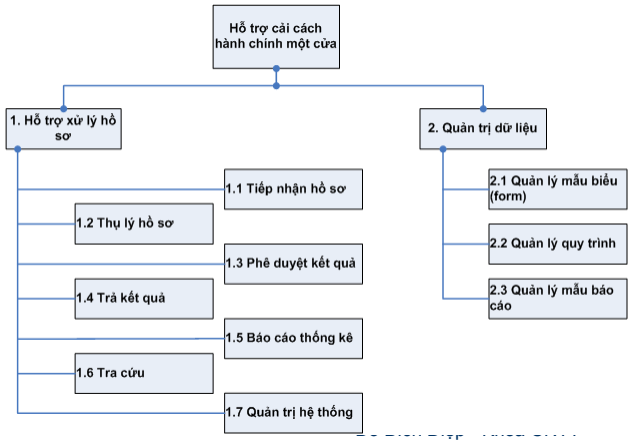
Cây thư mục trong máy tính



Đỗ Bích Diệp - Khoa CNTT

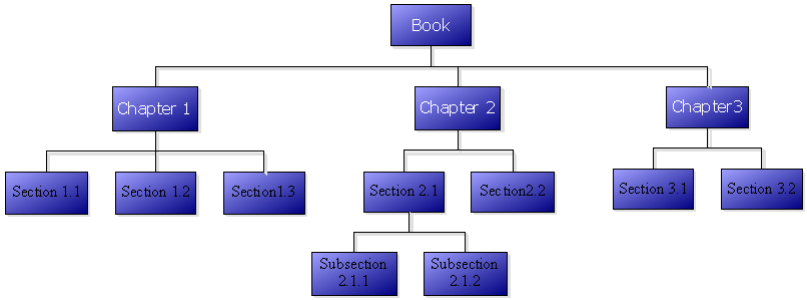
Ví dụ Cây

Cây phân cấp chức năng hệ thống thông tin



Ví dụ Cây

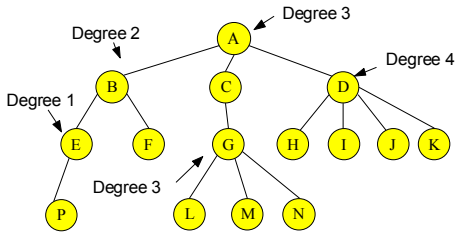
Cây mục lục Sách



Đỗ Bích Diệp - Khoa CNTT

Các thuật ngữ liên quan đến cây

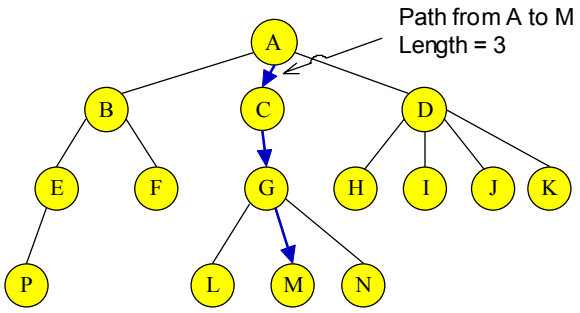
- Cấp (Degree) của một nút và của cây
  - Cấp của một nút là số các con của nút đó
  - Cấp của một cây là cấp cao nhất của một nút trên cây



Đỗ Bích Diệp - Khoa CNTT

Các thuật ngữ liên quan đến cây

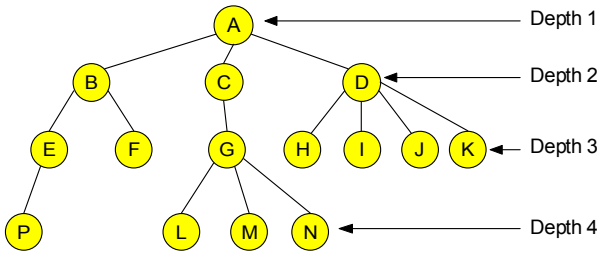
- Đường đi trên cây:
  - Dãy các nút $n_1, n_2, .., n_k$ trong đó n_i là nút cha của n_{i+1} (i = 1..k-1) là đường đi từ n_1 đến n_k



Đỗ Bích Diệp - Khoa CNTT

Các thuật ngữ liên quan đến cây

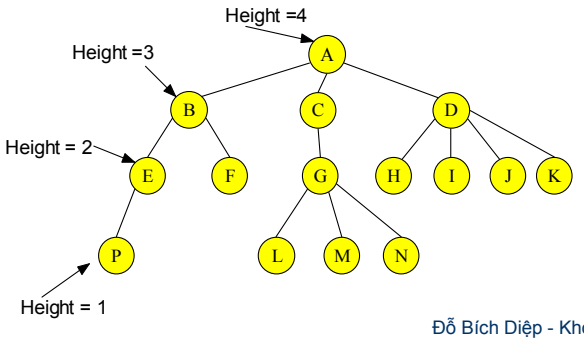
- Độ sâu hay mức (Depth – Level) của nút
  - Độ dài đường đi từ gốc đến nút đó + 1



Đỗ Bích Diệp - Khoa CNTT

Các thuật ngữ liên quan đến cây

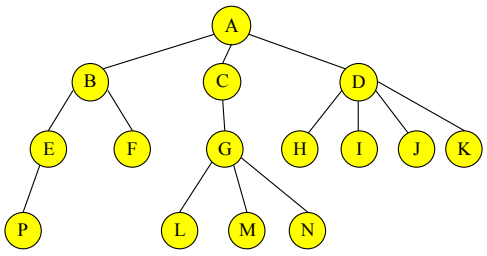
- Độ cao (Height) của nút
  - Độ dài đường đi dài nhất từ nút đó đến 1 nút lá trong cây + 1
  - Chiều cao của cây là chiều cao của nút gốc của cây đó



Đỗ Bích Diệp - Khoa CNTT

Các thuật ngữ liên quan đến cây

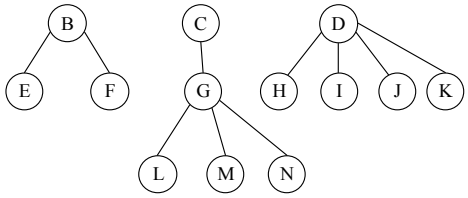
- Tổ tiên (Ancestor): A,C, G là tổ tiên của M
- Hậu duệ (descendants): E, F, G, H, L,M …đều là hậu duệ của A
- Anh em (siblings): E, F là một cặp anh em ; L, N là một cặp anh em



Đỗ Bích Diệp - Khoa CNTT

Các thuật ngữ liên quan đến cây

- Rừng là một tập hợp hữu hạn các cây phân biệt , không giao nhau



Đỗ Bích Diệp - Khoa CNTT

Các thao tác cơ bản trên Cây

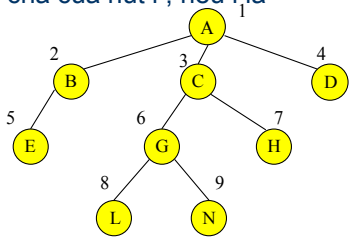
- Các thao tác truy nhập cây
  - root() : trả ra nút gốc của cây
  - parent(Tree T, Node p): trả ra nút cha của nút p trong cây T
  - children(Tree T, Node p): trả ra danh sách các nút con của nút p trong cây T
  - left_most_child(Tree T, Node p) : trả ra nút con cực trái của nút p
  - right_most_child(Tree T, Node p) : trả ra nút con cực phải của nút p
  - left_sibling (Tree T, Node p) : trả ra nút anh em kề cận bên trái của nút p
  - right_sibling(Tree T, Node p) : trả ra nút anh em kề cận bên phải của nút p
- Các thao tác khác
  - height (Tree T)
  - size(Tree T)
  - isRoot (Tree T, Node p); isLeaf (Tree T, Node p); isInternal (Tree T, Node p);

Đỗ Bích Diệp - Khoa CNTT

Biểu diễn cây tổng quát

- Dựa trên tham chiếu đến nút cha
  - Cây T có các nút được đánh số từ 1 đến n
  - Cây T được biểu diễn bằng một danh sách tuyến tính trong đó nút thứ i sẽ chứa một thành phần tham chiếu đến cha của nó
  - Nếu dùng mảng, A[i] = j nếu j là cha của nút i ; nếu i là gốc thì A[i] = 0;

0	1	1	1	2	3	3	6	6
A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]



Đỗ Bích Diệp - Khoa CNTT

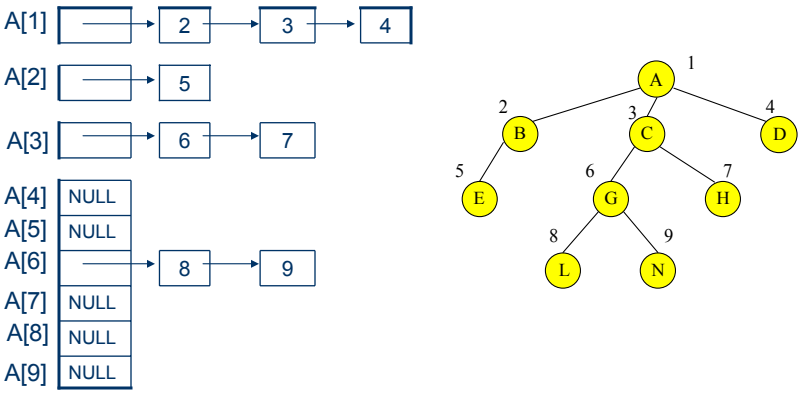
Biểu diễn cây tổng quát

- Dựa trên danh sách các nút con
  - 1 nút trong cây có một danh sách các nút con
  - Danh sách các nút con thường là danh sách móc nối
  - Trong trường hợp sử dụng danh sách móc nối, các nút đầu danh sách được lưu trong một mảng

Đỗ Bích Diệp - Khoa CNTT

Biểu diễn cây tổng quát

– Dựa trên danh sách các nút con



Đỗ Bích Diệp - Khoa CNTT

Biểu diễn cây tổng quát

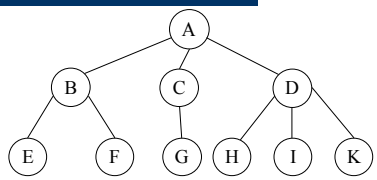
– Thông qua một cây cấp 2
- Với một nút trong cây , chỉ quan tâm tới 2 quan hệ
  – Quan hệ 1-1 giữa nút đó và nút con cực trái của nó (con cả)
  – Quan hệ 1-1 giữa nút đó và nút em kế cận bên phải của nó
- Dựa vào nhận định này, người ta biểu diễn được một cây tổng quát dưới dạng một cây nhị phân gọi là cây nhị phân tương đương (equivalent binary tree)
- Quy cách của 1 nút trên cây nhị phân tương đương sẽ như sau

LCHILD	INFO	RSIBLING

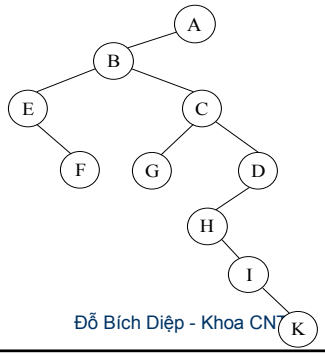
Đỗ Bích Diệp - Khoa CNTT

Biểu diễn cây tổng quát

- Ví dụ:
  - Cây tổng quát



  - Cây nhị phân tương đương

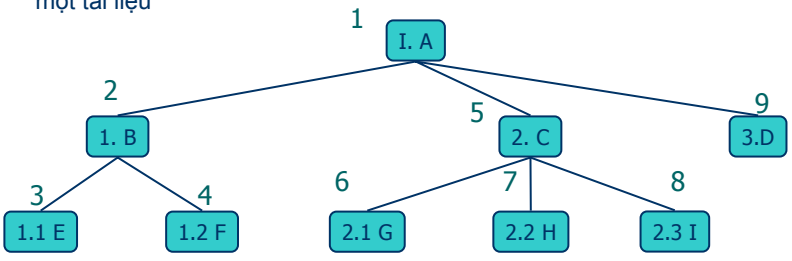


Đỗ Bích Diệp - Khoa CNTT

Duyệt cây theo thứ tự trước

- Duyệt cây là thăm các nút trên cây theo một thứ tự nhất định, mỗi nút thăm 1 lần
- Khi duyệt theo thứ tự trước, một nút sẽ được thăm trước các hậu duệ của nó
- Ứng dụng: In ra các mục lục của một tài liệu

Algorithm *preOrder*(*v*)
    visit(*v*)
    for each child *w* of *v*
        preOrder(*w*)



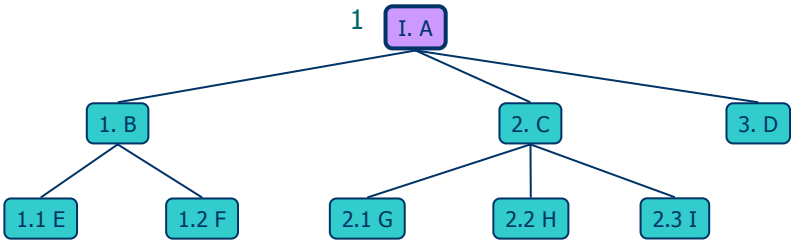
Đỗ Bích Diệp - Khoa CNTT

Duyệt cây theo thứ tự trước

- Duyệt cây là thăm các nút trên cây theo một thứ tự nhất định, mỗi nút thăm 1 lần
- Khi duyệt theo thứ tự trước, một nút sẽ được thăm trước các hậu duệ của nó
- Ứng dụng: In ra các mục lục của một tài liệu

⇒

```
Algorithm preOrder(v)
    visit(v)
    for each child w of v
        preOrder(w)
```



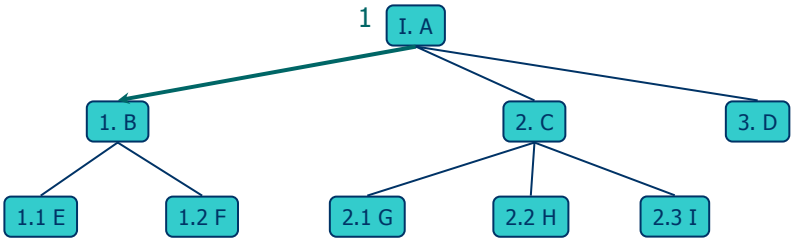
Đỗ Bích Diệp - Khoa CNTT

Duyệt cây theo thứ tự trước

- Duyệt cây là thăm các nút trên cây theo một thứ tự nhất định, mỗi nút thăm 1 lần
- Khi duyệt theo thứ tự trước, một nút sẽ được thăm trước các hậu duệ của nó
- Ứng dụng: In ra các mục lục của một tài liệu

⇒

```
Algorithm preOrder(v)
    visit(v)
    for each child w of v
        preOrder(w)
```



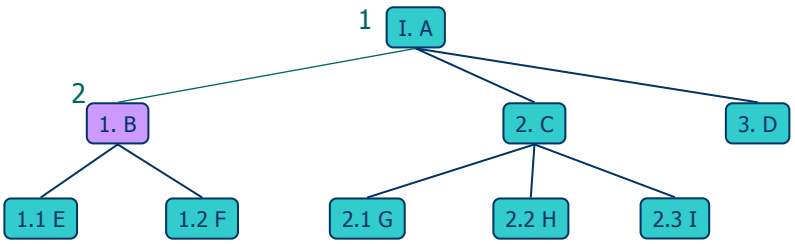
Đỗ Bích Diệp - Khoa CNTT

Duyệt cây theo thứ tự trước

- Duyệt cây là thăm các nút trên cây theo một thứ tự nhất định, mỗi nút thăm 1 lần
- Khi duyệt theo thứ tự trước, một nút sẽ được thăm trước các hậu duệ của nó
- Ứng dụng: In ra các mục lục của một tài liệu

⇒

```
Algorithm preOrder(v)
    visit(v)
    for each child w of v
        preOrder(w)
```



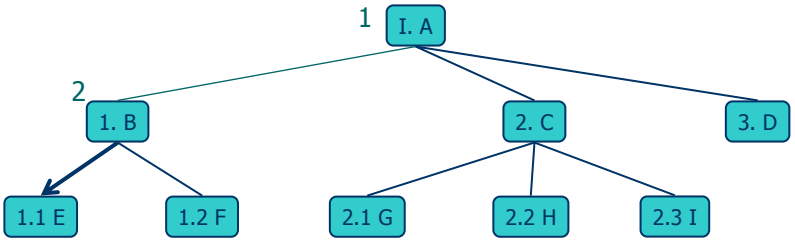
Đỗ Bích Diệp - Khoa CNTT

Duyệt cây theo thứ tự trước

- Duyệt cây là thăm các nút trên cây theo một thứ tự nhất định, mỗi nút thăm 1 lần
- Khi duyệt theo thứ tự trước, một nút sẽ được thăm trước các hậu duệ của nó
- Ứng dụng: In ra các mục lục của một tài liệu

⇒

```
Algorithm preOrder(v)
    visit(v)
    for each child w of v
        preOrder(w)
```



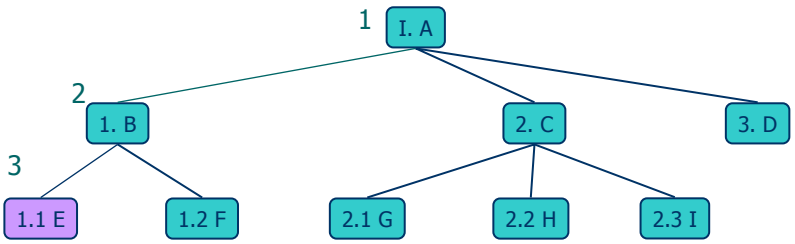
Đỗ Bích Diệp - Khoa CNTT

Duyệt cây theo thứ tự trước

- Duyệt cây là thăm các nút trên cây theo một thứ tự nhất định, mỗi nút thăm 1 lần
- Khi duyệt theo thứ tự trước, một nút sẽ được thăm trước các hậu duệ của nó
- Ứng dụng: In ra các mục lục của một tài liệu

⇒

```
Algorithm preOrder(v)
    visit(v)
    for each child w of v
        preOrder(w)
```

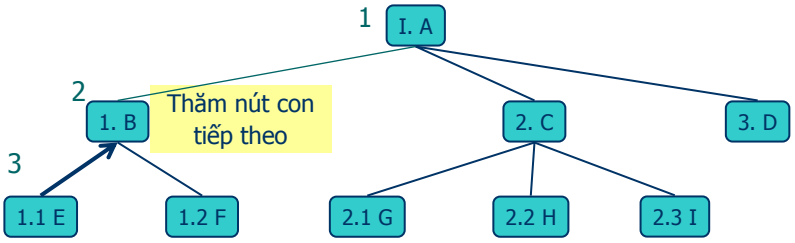


Đỗ Bích Diệp - Khoa CNTT

Duyệt cây theo thứ tự trước

- Duyệt cây là thăm các nút trên cây theo một thứ tự nhất định, mỗi nút thăm 1 lần
- Khi duyệt theo thứ tự trước, một nút sẽ được thăm trước các hậu duệ của nó
- Ứng dụng: In ra các mục lục của một tài liệu

```
Algorithm preOrder(v)
    visit(v)
    for each child w of v
        preOrder(w)
```

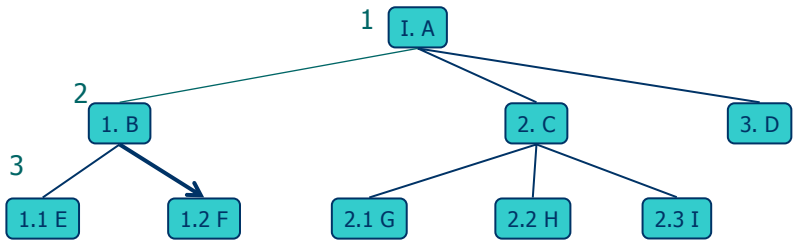


Đỗ Bích Diệp - Khoa CNTT

Duyệt cây theo thứ tự trước

- Duyệt cây là thăm các nút trên cây theo một thứ tự nhất định, mỗi nút thăm 1 lần
- Khi duyệt theo thứ tự trước, một nút sẽ được thăm trước các hậu duệ của nó
- Ứng dụng: In ra các mục lục của một tài liệu

Algorithm *preOrder*(*v*)
  visit(*v*)
  for each child *w* of *v*
⇒ *preOrder*(*w*)



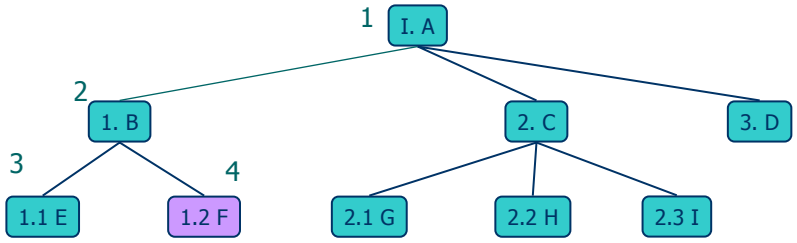
Đỗ Bích Diệp - Khoa CNTT

Duyệt cây theo thứ tự trước

- Duyệt cây là thăm các nút trên cây theo một thứ tự nhất định, mỗi nút thăm 1 lần
- Khi duyệt theo thứ tự trước, một nút sẽ được thăm trước các hậu duệ của nó
- Ứng dụng: In ra các mục lục của một tài liệu

⇒

Algorithm *preOrder*(*v*)
  visit(*v*)
  for each child *w* of *v*
    preOrder(*w*)

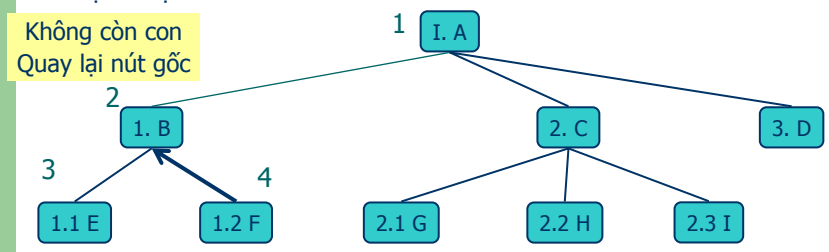


Đỗ Bích Diệp - Khoa CNTT

Duyệt cây theo thứ tự trước

- Duyệt cây là thăm các nút trên cây theo một thứ tự nhất định, mỗi nút thăm 1 lần
- Khi duyệt theo thứ tự trước, một nút sẽ được thăm trước các hậu duệ của nó
- Ứng dụng: In ra các mục lục của một tài liệu

```
Algorithm preOrder(v)
    visit(v)
    for each child w of v
        preOrder(w)
```

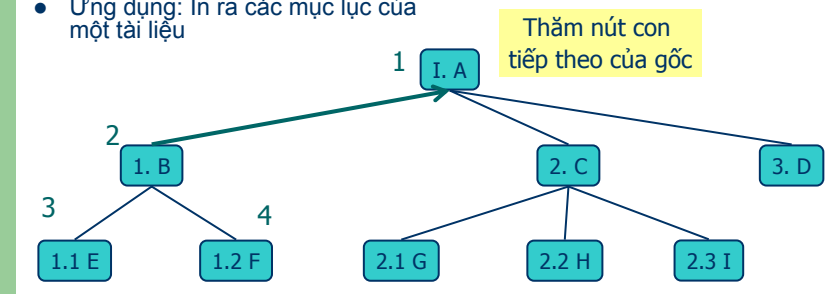


Đỗ Bích Diệp - Khoa CNTT

Duyệt cây theo thứ tự trước

- Duyệt cây là thăm các nút trên cây theo một thứ tự nhất định, mỗi nút thăm 1 lần
- Khi duyệt theo thứ tự trước, một nút sẽ được thăm trước các hậu duệ của nó
- Ứng dụng: In ra các mục lục của một tài liệu

```
Algorithm preOrder(v)
    visit(v)
    for each child w of v
        preOrder(w)
```

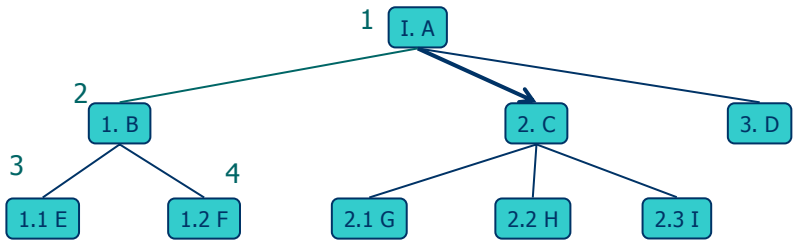


Đỗ Bích Diệp - Khoa CNTT

Duyệt cây theo thứ tự trước

- Duyệt cây là thăm các nút trên cây theo một thứ tự nhất định, mỗi nút thăm 1 lần
- Khi duyệt theo thứ tự trước, một nút sẽ được thăm trước các hậu duệ của nó
- Ứng dụng: In ra các mục lục của một tài liệu

Algorithm *preOrder*(*v*)
  visit(*v*)
  for each child *w* of *v*
⇒ *preOrder*(*w*)



Đỗ Bích Diệp - Khoa CNTT

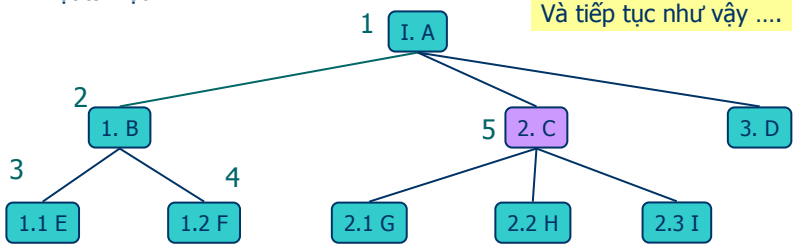
Duyệt cây theo thứ tự trước

- Duyệt cây là thăm các nút trên cây theo một thứ tự nhất định, mỗi nút thăm 1 lần
- Khi duyệt theo thứ tự trước, một nút sẽ được thăm trước các hậu duệ của nó
- Ứng dụng: In ra các mục lục của một tài liệu

⇒

Algorithm *preOrder*(*v*)
  visit(*v*)
  for each child *w* of *v*
    preOrder(*w*)

Và tiếp tục như vậy ....

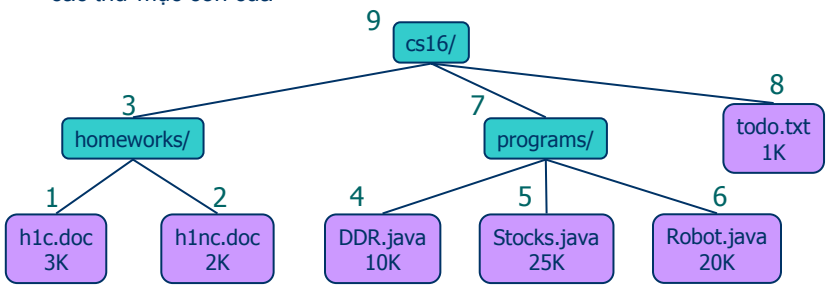


Đỗ Bích Diệp - Khoa CNTT

Duyệt cây theo thứ tự sau

- Duyệt theo thứ tự sau thì một nút sẽ được thăm sau các hậu duệ của nó
- Ứng dụng: Xác định kích thước của các tệp trong một thư mục và các thư mục con của

```
Algorithm postOrder(v)
    for each child w of v
        postOrder(w)
    visit(v)
```

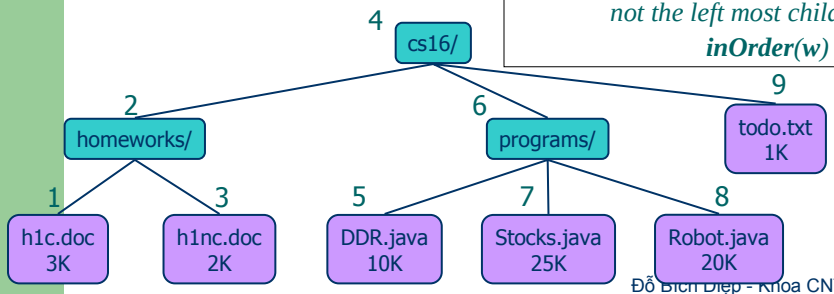


Đỗ Bích Diệp - Khoa CNTT

Duyệt cây theo thứ tự giữa

- Duyệt theo thứ tự giữa thì một nút sẽ được thăm sau các hậu duệ của nó trong cây con cực trái và trước các hậu duệ trong các cây con tiếp theo

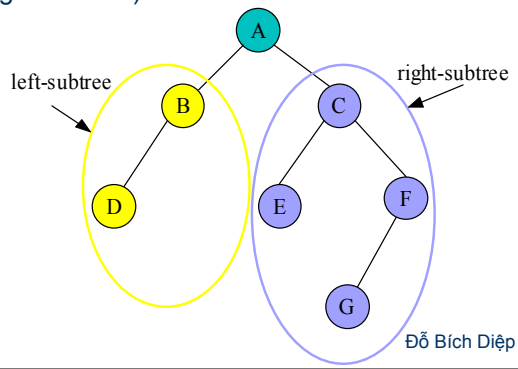
```
Algorithm inOrder(v)
    if (isLeaf(v)) then visit(v)
    else
        inOrder(left_most_child(v))
        visit(v)
        for each child w of v (w is
        not the left most child)
            inOrder(w)
```



Đỗ Bích Diệp - Khoa CNTT

Cây nhị phân (Binary Tree)

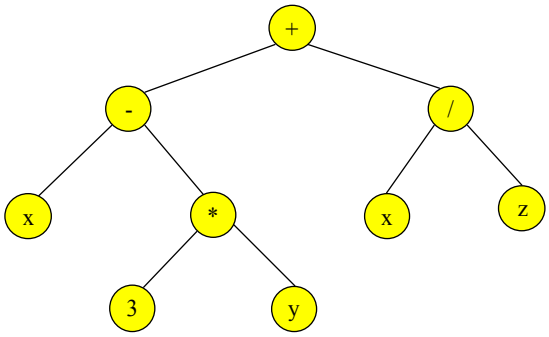
- Là cây mà mọi nút trên cây chỉ có tối đa là 2 con.
  - Cây con của một nút cũng cần phải được phân biệt rõ ràng thành cây con trái (left subtree) và cây con phải (right subtree)



Đỗ Bích Diệp - Khoa CNTT

Ví dụ cây nhị phân

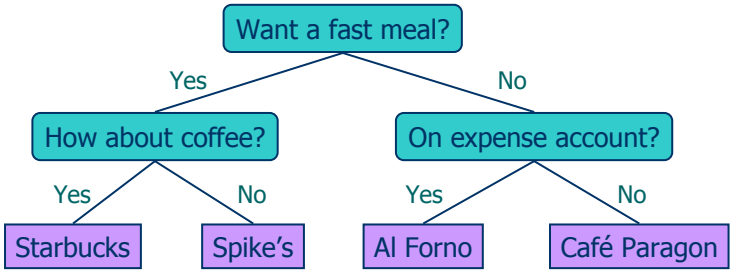
- Cây biểu thức số học với các phép toán 2 ngôi



Đỗ Bích Diệp - Khoa CNTT

Ví dụ cây nhị phân

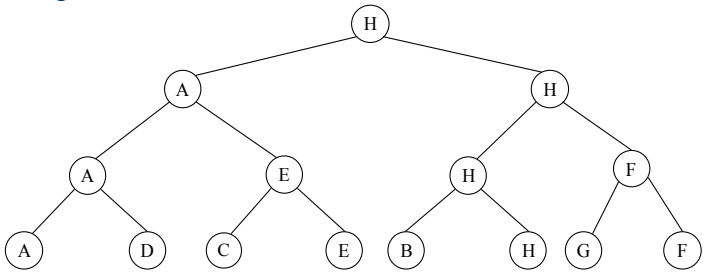
- Cây quyết định



Đỗ Bích Diệp - Khoa CNTT

Ví dụ cây nhị phân

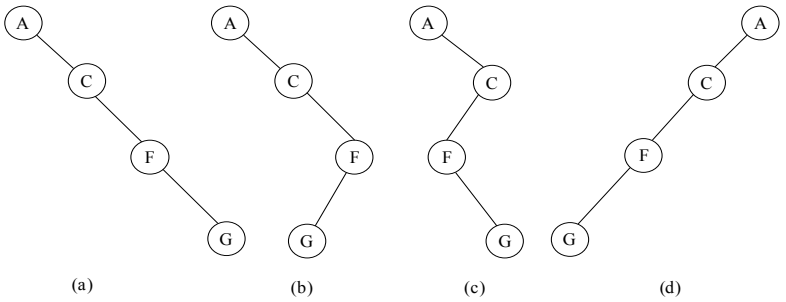
- Kết quả thi đấu một môn thể thao theo cặp tại nhiều vòng



Đỗ Bích Diệp - Khoa CNTT

Các dạng đặc biệt của cây nhị phân

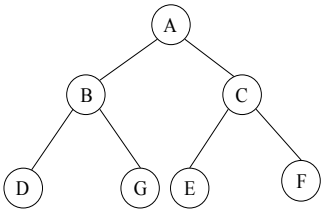
- Cây suy biến (degenerate binary tree)
  - Mỗi nút trong của cây có đúng 1 nút con



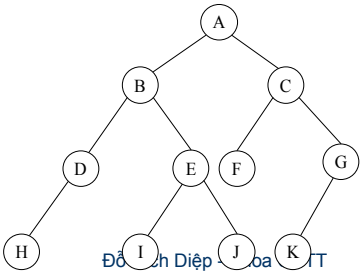
Đỗ Bích Diệp - Khoa CNTT

Các dạng đặc biệt của cây nhị phân

- Cây nhị phân đầy đủ (full binary tree)
  - Mỗi nút trong của cây đều có đầy đủ 2 con



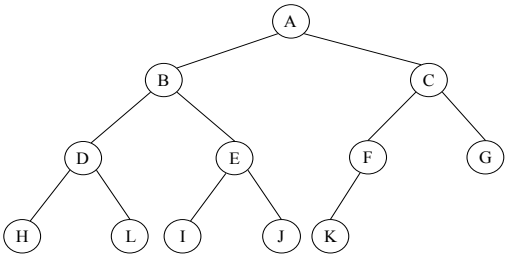
- Cây nhị phân gần đầy
  - Ở mức cuối cùng không có đầy đủ các nút



Đỗ Bích Diệp - Khoa CNTT

Các dạng đặc biệt của cây nhị phân

- Cây nhị phân hoàn chỉnh
  - Là cây nhị phân gần đầy
  - Tất cả các nút ở mức cuối cùng đều lệch về bên trái nhất có thể



- Cây nhị phân cân đối
  - Cây con trái và cây con phải lệch nhau không quá 1 đơn vị

Tính chất của Cây nhị phân

1. Số lượng tối đa của các nút ở mức i trên một cây nhị phân là 2^{i-1} (i >= 1)
2. Số lượng tối đa các nút trên một cây nhị phân có chiều cao là h là $2^h - 1$ (h >= 1)
3. Một cây nhị phân có n nút có chiều cao tối thiểu là

$$\lceil \log_2(n+1) \rceil$$

4. Một cây nhị phân đầy đủ có độ sâu n thì có $2^n - 1$ nút
5. Một cây nhị phân hoàn chỉnh có chiều cao h có số lượng nút nằm trong khoảng 2^{h-1} đến $2^h - 1$
6. Trong một cây nhị phân có n_0 nút lá và n_2 nút cấp 2 thì ta có $n_0 = n_2 + 1$

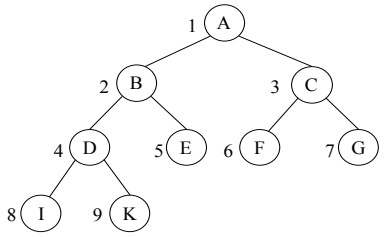
Biểu diễn cây nhị phân

- Biểu diễn kế tiếp sử dụng mảng
  - Đánh số các nút trên cây theo trình tự từ mức 1, hết mức này đến mức khác, từ trái sang phải
  - Lưu trữ trong vector lưu trữ V theo nguyên tắc phần tử V[i] sẽ lưu thông tin của nút được đánh số i

Đỗ Bích Diệp - Khoa CNTT

Biểu diễn cây nhị phân

- Ví dụ



- Cây cho ở trên được lưu trữ trên vector lưu trữ V như sau

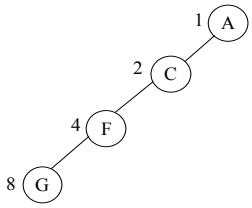
A	B	C	D	E	F	G	I	K
V[1]	V[2]	V[3]	V[4]	V[5]	V[6]	V[7]	V[8]	V[9]

Đỗ Bích Diệp - Khoa CNTT

Biểu diễn cây nhị phân

- Cách lưu trữ kế tiếp phù hợp để lưu trữ cây nhị phân gần đầy hoặc đầy đủ
- Với các dạng khác có thể dẫn đến lãng phí bộ nhớ

A	C		F				8
V[1]	V[2]	V[3]	V[4]	V[5]	V[6]	V[7]	V[8]



Đỗ Bích Diệp - Khoa CNTT

Biểu diễn cây nhị phân

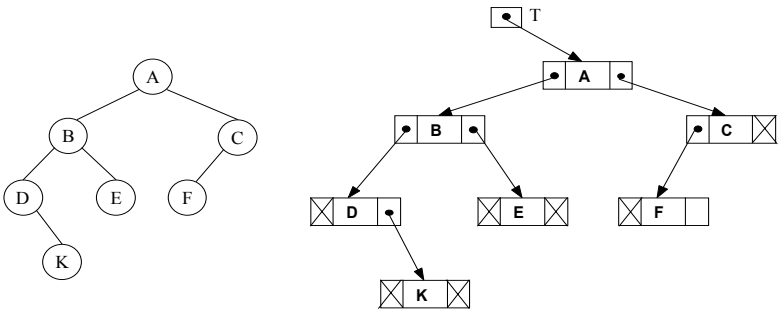
- Biểu diễn móc nối sử dụng con trỏ
  - Mỗi nút trên cây được lưu trữ bởi một phần tử có quy cách như sau

LPTR	INFO	RPTR

    - INFO: chứa dữ liệu của nút
    - LPTR: chứa địa chỉ của nút gốc của cây con trái
    - RPTR: chứa địa chỉ của nút gốc của cây con phải
  - Cần nắm một con trỏ T trỏ tới nút gốc của cây. Nếu cây rỗng thì T = NULL

Đỗ Bích Diệp - Khoa CNTT

Biểu diễn cây nhị phân



Đỗ Bích Diệp - Khoa CNTT

Biểu diễn cây nhị phân

```
struct Tnode{
        int  info;
        struct Tnode * lptr;
        struct Tnode * rptr;
} ;
typedef struct Tnode TREENODE;
typedef TREENODE *TREENODEPTR;
```

Đỗ Bích Diệp - Khoa CNTT

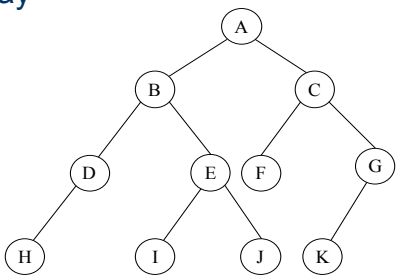
Duyệt cây nhị phân

- Phép duyệt cây nhị phân
  - Phép duyệt một cây là phép "thăm" lần lượt các nút trên cây đó sao cho mỗi nút chỉ được thăm một lần
  - Tồn tại 3 phép duyệt khác nhau đối với 1 cây nhị phân
    - Duyệt cây theo thứ tự trước
    - Duyệt cây theo thứ tự giữa
    - Duyệt cây theo thứ tự sau:

Đỗ Bích Diệp - Khoa CNTT

Duyệt cây nhị phân

- Ví dụ: Thực hiện duyệt cây
  - Duyệt theo thứ tự trước
    A B D H E I J C F G K
  - Duyệt theo thứ tự giữa
    H D B I E J A F C K G
  - Duyệt theo thứ tự sau
    H D I J E B F K G C A



Đỗ Bích Diệp - Khoa CNTT

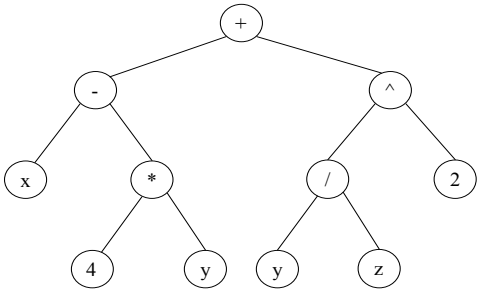
Duyệt cây nhị phân theo thứ tự trước

```
void PREORDER(TREENODEPTR tree) {
if (tree != NULL) {
     printf("%3d", tree->info;
      PREORDER(tree->lptr);
     PREORDER(tree->rptr);
     }
}
```

Đỗ Bích Diệp - Khoa CNTT

Duyệt cây nhị phân

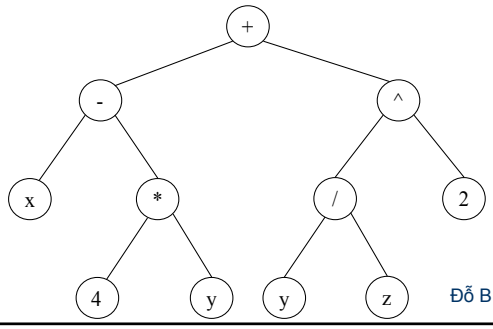
– Ví dụ 2: Cho cây nhị phân biểu diễn biểu thức số học sau, hãy đưa ra dãy các nút được thăm khi thực hiện các phép duyệt theo thứ tự trước, giữa và sau. Nhận xét về các dãy thu được



Đỗ Bích Diệp - Khoa CNTT

Cây biểu thức

- Bài toán 1: Dựng cây biểu diễn biểu thức số học:
  - Cho một biểu thức số học dưới dạng hậu tố, dựng cây biểu diễn biểu thức số học đó
  - Ví dụ: Cho biểu thức x 4 y * - y z / 2 ^ + . Dựng được cây biểu diễn biểu thức này như sau



Đỗ Bích Diệp - Khoa CNTT

Dựng cây biểu diễn biểu thức

- Giải thuật

```
Function BUILD_TREE_POSTFIX(TOKENS, n)
{TOKENS : mang cac token cua xâu ký tự biểu diễn biểu thức ban đầu là biểu
    thức dưới dạng hậu tố, dưới dạng một mảng. n là số ký tự trong xâu}
{ S : Stack de chua du lieu tam thoi}
Begin
for i = 1 to n do
Begin
    TK = TOKENS[i];
    if isNumber(TK) | isChar(TK) Then
   begin
        Node = CreateTreeNode(TK); {tạo cây có một nút gốc là hạng tử }
       PUSH(S, Node);
   end;
```

Đỗ Bích Diệp - Khoa CNTT

Dựng cây biểu diễn biểu thức

- Giải thuật (tiếp)

```
Else {TK la 1 toan tu}
   begin
       Right = POP(S);
       Left   = POP(S);
       Node = CreateTreeNode(TK, Right, Left); {tao 1 cay gom 1 nut goc la
toan tu TK va 2 nut con la 2 hang tu la right, left}
      PUSH(S,Node);
  end;
end;
T= POP(S);
Return T;
End;
```

Đỗ Bích Diệp - Khoa CNTT

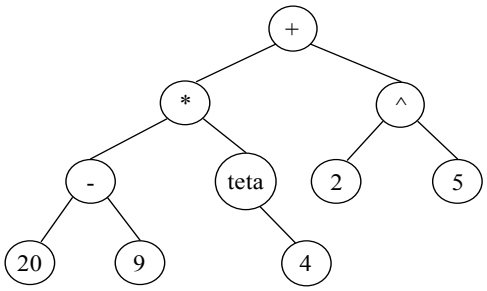
Cây biểu thức

- Bài toán 2: Tính giá trị của biểu thức số học biểu diễn bằng một cây nhị phân
  - Các nút lá biểu diễn các giá trị của các toán hạng
  - Các nút nhánh biểu diễn các dấu phép toán
    - Các dấu phép toán có thể sử dụng trong bài toán này là: +, - , *, /, ^ và teta (biểu diễn dấu âm)
    - Qui ước là với nút nhánh là teta thì toán hạng của nó là con phải của nó

Đỗ Bích Diệp - Khoa CNTT

Tính giá trị của biểu thức số học

- Ví dụ



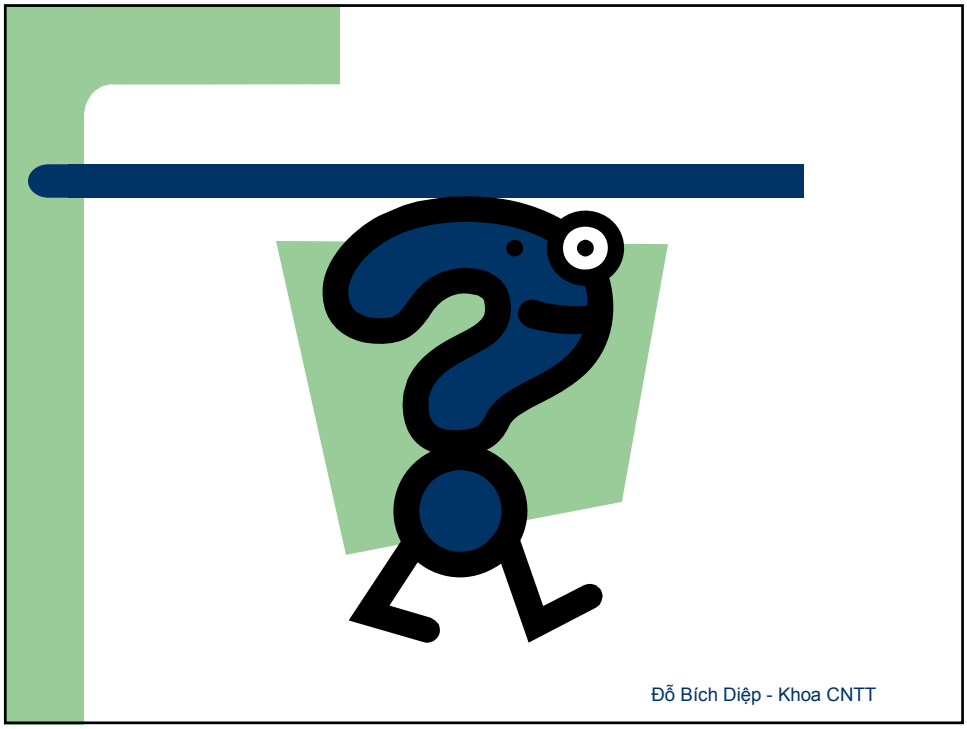
Đỗ Bích Diệp - Khoa CNTT

Tính giá trị của biểu thức

- Giải thuật tính giá trị biểu thức biểu diễn bằng cấu trúc cây

```
Function COMPUTE_EXPRESSION(T) Begin
  IF   IsLeaf(T) THEN Result := VAL(INFO(T));
  ELSE   BEGIN
     leftvalue  := COMPUTE_EXPRESSION(LPTR(T));
     rightvalue := COMPUTE_EXPRESSION(RPTR(T));
     case (INFO(T))
            '+' : Result  = leftvalue + rightvalue;
            '-' :  Result  = leftvalue - rightvalue;
            '*' : Result  = leftvalue * rightvalue;
            '/' :  Result  = leftvalue / rightvalue;
           '^' :  Result  = leftvalue ^ rightvalue;
           'teta' :  Result  = -( rightvalue);
     end case;
     END;
     return Result;
End
```

Đỗ Bích Diệp - Khoa CNTT


Đỗ Bích Diệp - Khoa CNTT