



FPT POLYTECHNIC



Bài 6:
CÁC C U TRÚC D LI U C BI T:
NG N X P, HÀNG I, CÂY

www.poly.edu.vn

hoclaptrinhweb.com

- Tìm hi u các gi i thu t s p x p c b n trên c u trúc d li u m ng
- Tìm hi u các gi i thu t tìm ki m c b n trên c u trúc d li u m ng
- ánh giá và so sánh hi u qu các gi i thu t

Mục tiêu bài học hôm nay

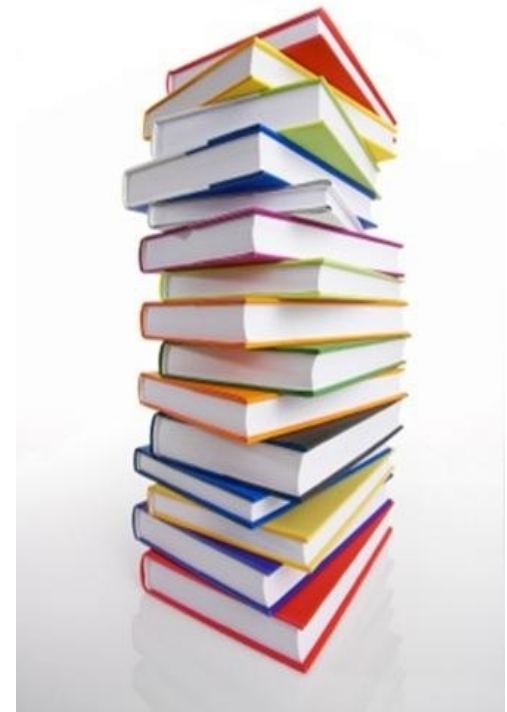
- Tìm hiểu 3 cấu trúc dữ liệu cơ bản: Ngăn xếp (Stack), Hàng đợi (Queue) và Cây (Tree):
 - Khái niệm
 - Cách cài đặt trong VB.Net
 - Các thao tác cơ bản trên các cấu trúc dữ liệu

Khái niệm ngăn xếp

■ Ngăn xếp (Stack):

- Các phần tử được lưu trữ thành một danh sách liên tiếp nhau.
- Vì có **thêm** hay **loại** lý một phần tử ra khỏi danh sách được thực hiện một cách duy nhất là **nhập** và **xuất**.

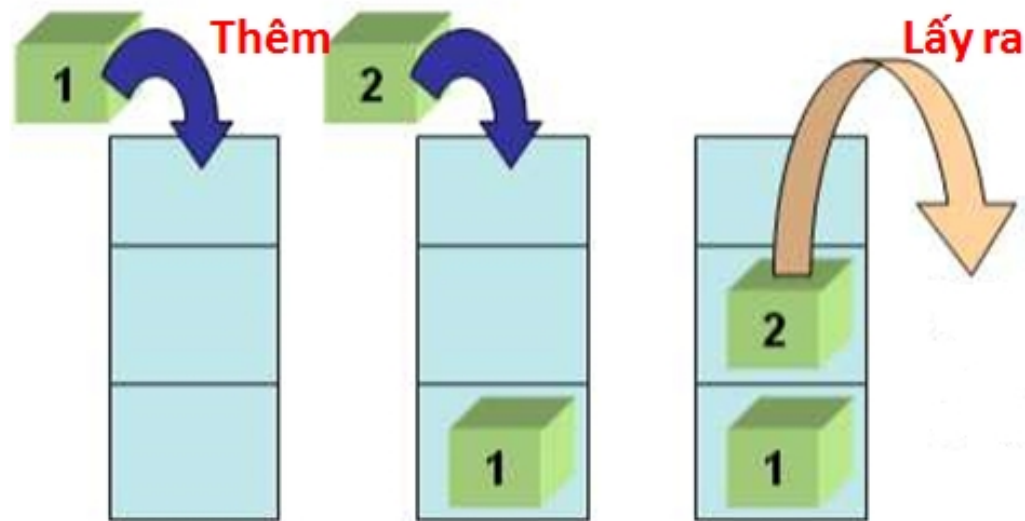
■ Ví dụ: Chồng sách trên bàn



Khái niệm ngăn xếp

■ Stack tuân theo cấu trúc: LIFO (Last In – First Out):

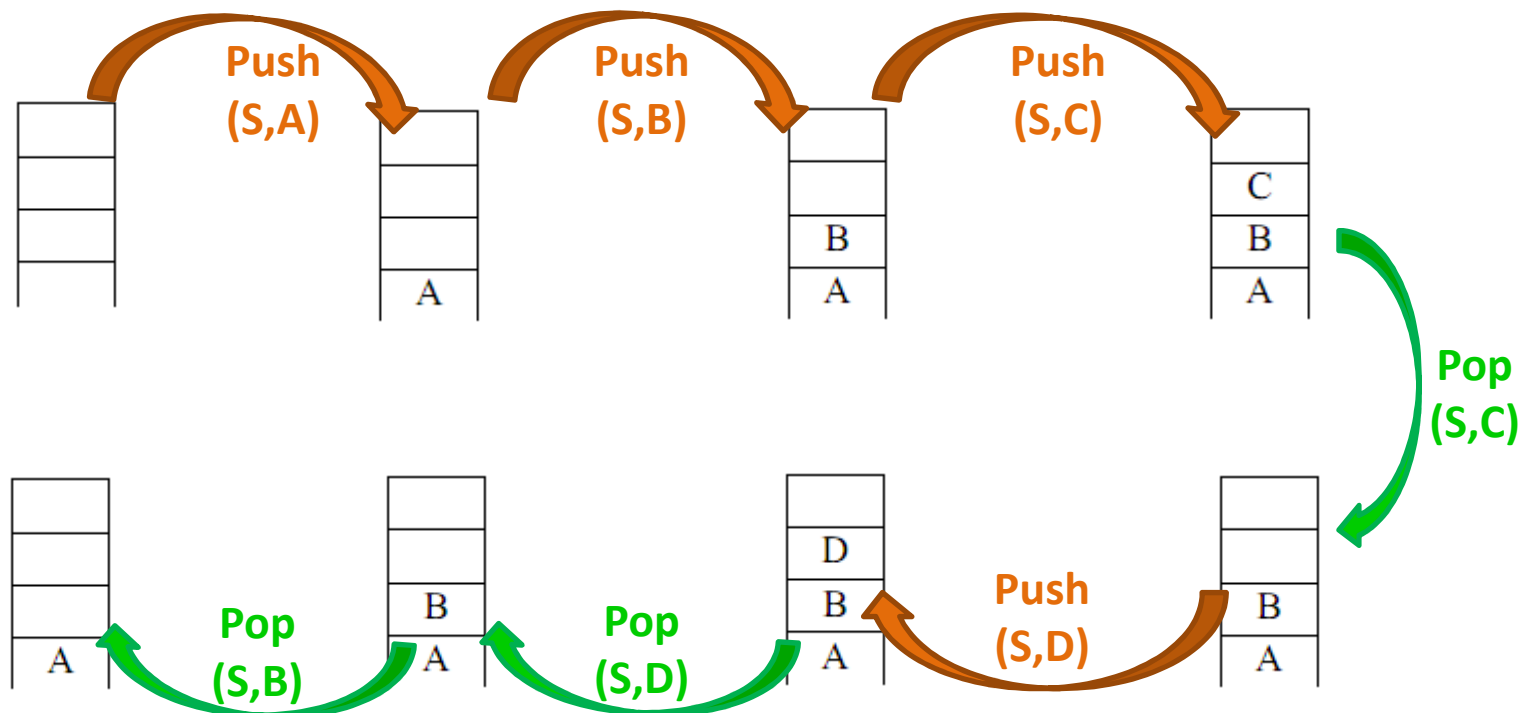
- Phép toán đưa vào trong ngăn xếp sau cùng sẽ lấy ra trước tiên.
- Phép toán đưa vào trong ngăn xếp trước tiên sẽ lấy ra sau cùng.



Các thao tác trên ngăn xếp

- Có một số thao tác với ngăn xếp hay cách thức hiển thị
 - **Thêm** (push) một phần tử vào ngăn xếp
 - **Lấy** (pop) một phần tử từ ngăn xếp
 - **Xem** (peek) nội dung của phần tử ngăn xếp

Stack S lưu trữ các kí tự



Lớp Stack trong VB.Net

- Ngăn xếp cài đặt trong VB.Net bằng lớp Stack
- Lớp Stack là cài đặt giao diện ICollection
- Lớp Stack cung cấp các phương thức cho phép thực hiện các thao tác trên ngăn xếp như : **Push()**, **Pop()**, **Peek()**, **Contains()**, ...

3 cách kh i t o:

- **Cách 1:** T o 1 ng n x p r ng m c nh ch a c 10 giá tr

- Ví d : `Dim myStack As New Stack()`

- **Cách 2:** T o 1 ng n x p t 1 i t ng collection khác

- Ví d : `Dim names() As String = {"Raymond",
"David","Mike"}`

`Dim nameStack As New Stack(names)`

- **Cách 3:** T o 1 ng n x p và ch nh luôn dung l ng ng n x p

- Ví d : `Dim myStack As New Stack(25)`

Các phương thức của Stack

- **Push():** Thêm một phần tử (Item) vào danh sách myStack
 - Cú pháp: **myStack.Push(Item)**
- **Pop():** Lấy phần tử từ danh sách myStack
 - Cú pháp: **myStack.Pop()** -> trả về phần tử đang nằm ở đỉnh
- **Peek():** Xem nội dung phần tử tại vị trí đang nằm ở đỉnh
 - Cú pháp: **myStack.Peek()**

Các phương thức của Stack

- **Count():** Trả về số phần tử có trong ngăn xếp myStack
 - Cú pháp: **myStack.Count()**
- **Clear():** Xóa tất cả các phần tử có trong ngăn xếp myStack
 - Cú pháp: **myStack.Clear()**
- **Contains():** Kiểm tra xem một phần tử Item nào đó có tồn tại trong ngăn xếp myStack không
 - Cú pháp: **myStack.Contains(Item)**

Các phương thức của Stack

- **CopyTo()**: copy nội dung của mảng myStack vào mảng myArray bắt đầu từ vị trí **index**
 - Cú pháp: **myStack.CopyTo(myArray, index)**
- **ToArray()**: copy nội dung của mảng myStack vào mảng myArray
 - Cú pháp: **myArray = myStack.ToArray()**

Ví dụ sử dụng mảng s

phương thức

```
Imports System.Collections

Module Module1
    Sub Main()
        Dim Nums As New Stack()
        Dim num As Integer
        Dim x As Integer
        Dim arrayCopy() As Object
        Dim myArray() As Object

        'Phuong thuc Push()
        For x = 5 To 20 Step +5
            Nums.Push(x)
        Next

        ' Phuong thuc Peek() va Pop()
        If (IsNumeric(Nums.Peek())) Then
            num = Nums.Pop()
            Console.WriteLine("Phan tu vua duoc lay ra
khoi ngan xep la: " + num)
        End If
    End Sub
End Module
```

```
'Phuong thuc Contains()  
If (Nums.Contains(10)) Then  
    Console.WriteLine("Gia tri 10 ton  
tai trong Stack")  
End If  
  
'Phuong thuc Count() va Clear()  
If (Nums.Count() = 0) Then  
    Nums.Clear()  
End If  
  
'Phuong thuc CopyTo() va ToArray()  
Nums.CopyTo(arrayCopy, 0)  
myArray = Nums.ToArray()  
  
End Sub  
End Module
```

Ví dụ ứng dụng Stack

- Ứng dụng stack trong các danh sách

- Giải thuật:

1. Lắp rập

- 1.1. Nhập vào một giá trị

- 1.2. Push nó vào stack

2. Lắp rập khi stack chưa rỗng

- 2.1. Lấy một giá trị từ stack

- 2.2. In ra

Ví dụ ứng dụng Stack

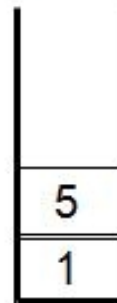
Ban đầu



Nhập 1



Nhập 5



Nhập 7



Nhập 3



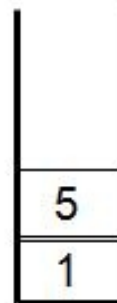
Lấy ra => 3



Lấy ra => 7



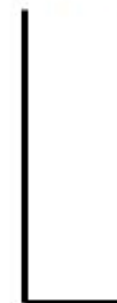
Lấy ra => 5



Lấy ra => 1



Stack đã rỗng
Ngừng



Khái niệm Hàng đợi (Queue)

■ Hàng đợi (Queue):

- Các phần tử được lưu trữ thành một danh sách liên tiếp nhau.
- Vì có **thêm** 1 phần tử vào danh sách có thể hiển thị một đầu (**cuối hàng**)
- Vì có **lấy** ra 1 phần tử của danh sách có thể hiển thị đầu khác (**đầu hàng**)

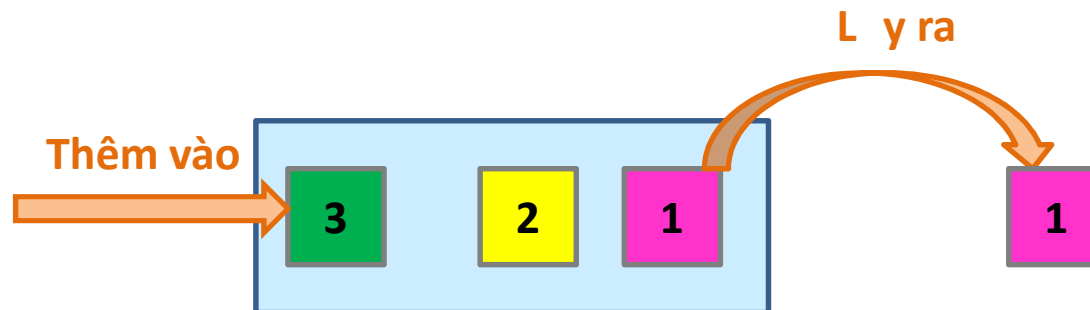
■ Ví dụ :

- Dòng người xếp hàng chờ trong siêu thị



Khái niệm Hàng đợi (Queue)

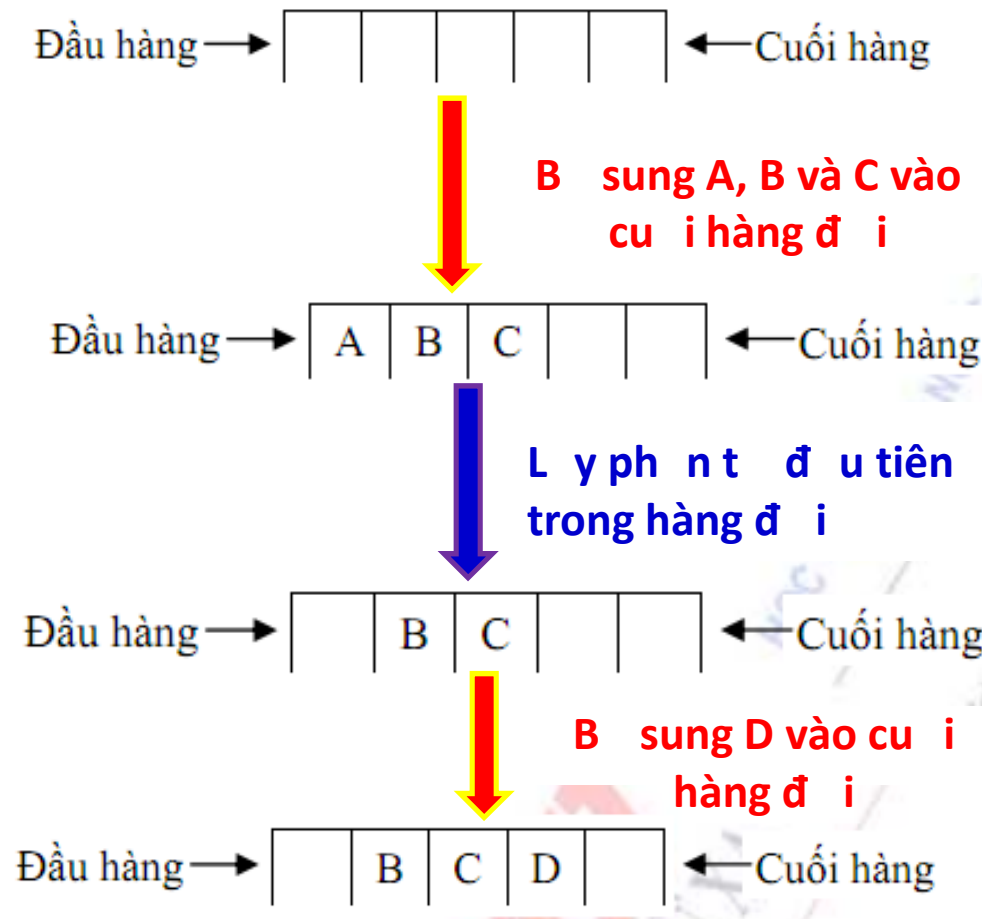
- Hàng đợi tuân theo cấu trúc FIFO (First In – First Out):
 - Các phần tử vào trong hàng đợi trước sẽ lấy ra trước.



Các thao tác trên hàng đợi

- Một số thao tác cơ bản trên queue
 - Thêm (enqueue) thêm phần tử vào cuối hàng đợi
 - Lấy (dequeue) phần tử đầu hàng đợi
 - Xem (peek) nội dung phần tử đầu hàng đợi

- Ví dụ hàng đợi Q lưu trữ các kí tự



Lớp Hàng đợi trong VB.Net

- Hàng đợi cài đặt trong VB.Net bằng lớp Queue
- Lớp Queue bao gồm các phương thức như :
 - **Enqueue()**: Bổ sung thêm phần tử vào cuối của hàng đợi
 - **Dequeue()**: Lấy ra phần tử từ đầu của hàng đợi
 - **Peek()**: Xem nội dung của phần tử từ đầu của hàng đợi
 - **Count()**: Trả về số phần tử có trong hàng đợi
 - **Clear()**: Xóa hàng đợi và tất cả các phần tử có trong hàng đợi

Cách khai báo Hàng đợi

Có 3 cách:

- **Cách 1:** Tạo một hàng đợi rỗng mặc định chứa các 32 phần tử
 - Ví dụ : `Dim myQueue As New Queue()`
- **Cách 2:** Tạo một hàng đợi và chỉ định số dung lượng hàng đợi
 - Ví dụ : `Dim myQueue As New Queue(100)`
- **Cách 3:** Tạo một mảng và chỉ định kích thước gia tăng kích thước
 - Ví dụ : `Dim myQueue As New Queue(32, 3R)`

- **Enqueue():** Bổ sung phần tử Item vào cuối của hàng đợi myQueue
 - Cấu trúc: **myQueue.Enqueue(Item)**
- **Dequeue():** Lấy phần tử đầu của hàng đợi myQueue
 - Cấu trúc: **myQueue.Dequeue**
- **Peek():** Tham phần tử đầu của hàng đợi myQueue
 - Cấu trúc: **myQueue.Peek**

Các phương thức trong Hàng đợi

- **Count:** Trả về số phần tử có trong hàng đợi
 - Cấu trúc: `myQueue.Count`
- **Clear:** Xóa hàng đợi và tất cả các phần tử có trong hàng đợi
 - Cấu trúc: `myQueue.Clear`

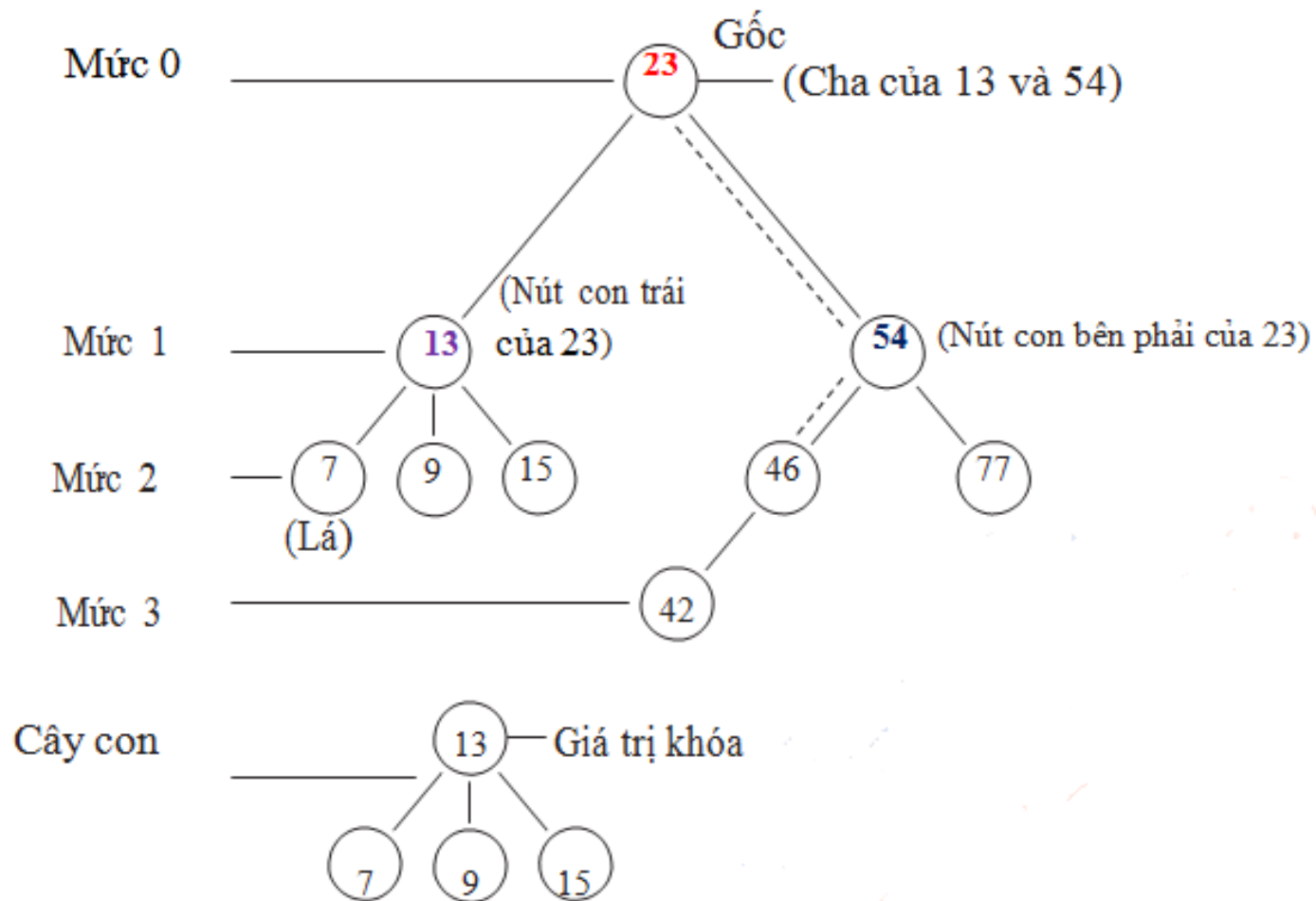
■ Hàng i có th c s d ng trong m t s bài toán:

- Bài toán 's n xu t và tiêu th ' (ng d ng trong các h i u hành song song).
- B m l nh (ví d : Nh n phím \Rightarrow B m \Rightarrow CPU x lý).
- X lý các l nh trong máy tính (ng d ng trong H H, trình biên d ch), hàng i các ti n trình ch c x lý,

Khái niệm Cây (Tree)

- Cây là tập hợp các phần tử (Nút) và các cạnh kết nối chúng theo các quy tắc sau:
 - Mọi nút của cây mang một thông tin nào đó
 - Cách các cạnh dùng để liên kết các nút với nhau

Khái niệm Cây (Tree)



Khái niệm Cây (Tree)

■ Các khái niệm liên quan:

- **Nút cha – Nút con:** Mọi nút trong cây có đúng một nút nằm trên nó (nút cha), các nút nằm ngay dưới nó gọi là nút con
- **Nút anh em:** các nút nằm cùng cấp
- **Nút lá:** Nút không có nút con nào.

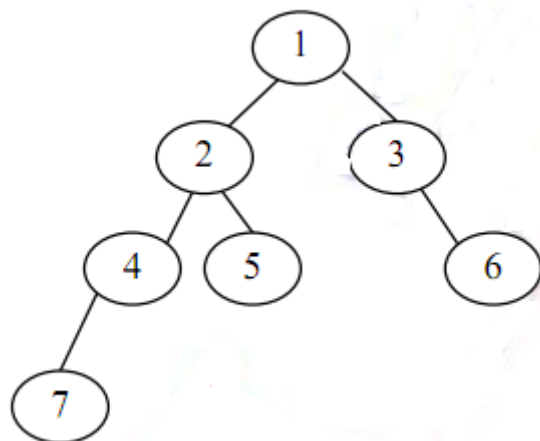
Khái niệm Cây (Tree)

■ Các khái niệm liên quan:

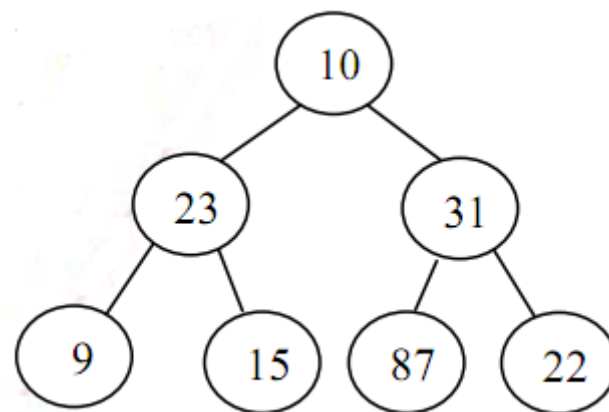
- **Bậc của nút**: là số cây con của nút đó
- **Bậc của cây**: là bậc lớn nhất của các nút trong cây
- **Mức của nút**: mức của nút gốc cây con của nó bằng thêm 1, trong đó mức của nút gốc = 0.
- **Chiều cao của cây**: là mức cao nhất của các nút trong cây
- **Chiều dài đường đi của một nút**: là tổng số nút tính từ nút gốc đến nút đó.

Cây nh phân

- Cây nh phân là cây c bi t mà m i nút c a nó ch có nhi u nh t 2 nút con.
- Khi ó, 2 cây con c a m i nút c g i là cây con trái và cây con ph i



Cây nh phân



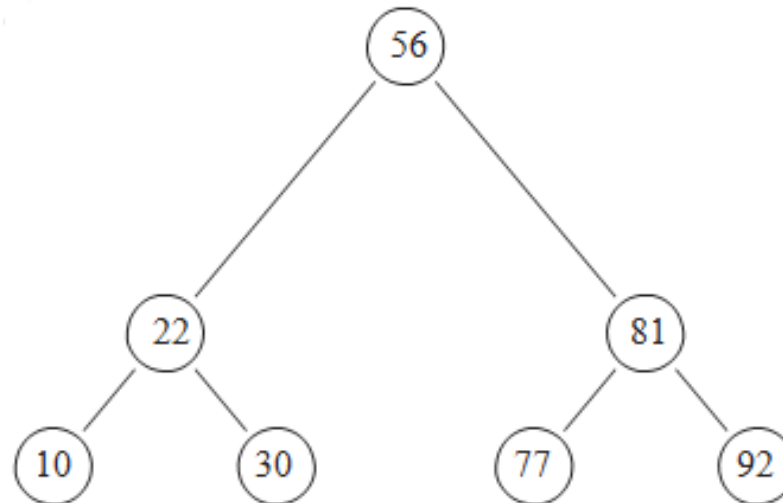
Cây nh phân đ y đ

Cây nh phân tìm kiếm

■ Là cây nh phân có tính chất:

- Khóa của nút con bên trái bao giờ cũng nhỏ hơn khóa của nút cha
- Khóa của nút con bên phải bao giờ cũng lớn hơn khóa của nút cha.

■ Ví dụ



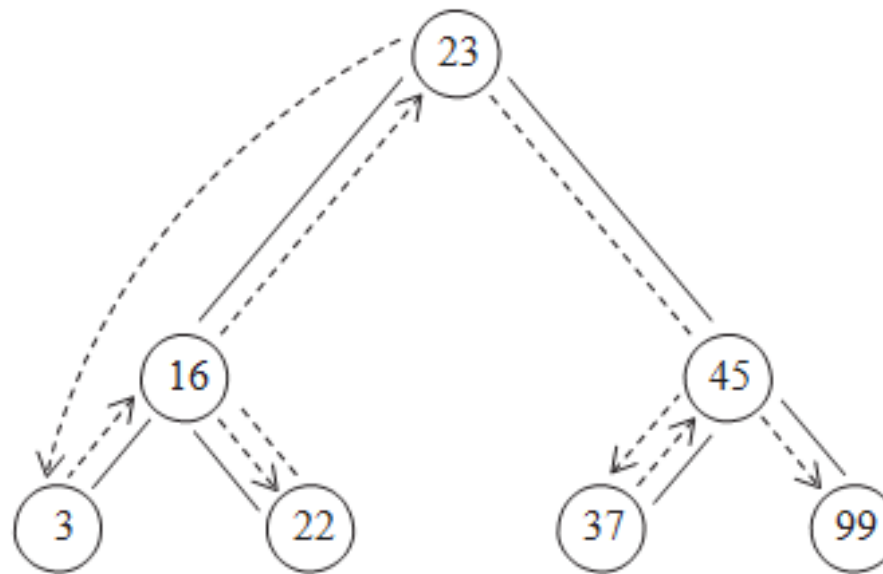
Duy_t cây nh_phân

- Là thao tác th_m t_t c_các nút c_a cây, m_i nút úng m_t l_n.
- 3 ph_nhng pháp duy_t:
 - Duy_t theo th_t tr_c
 - Duy_t theo th_t gi_a
 - Duy_t theo th_t cu_i
- Ph_nhng pháp qui_c_nhng d_nhng cài_t các ph_nhng pháp duy_t trên

Duyệt cây nh phân

■ Ví dụ duyệt theo thứ tự giữa:

- Duyệt cây gốc 16, Thăm 23, Duyệt cây gốc 45
- Thứ tự duyệt: 3, 16, 22, 23, 27, 45, 99



ng d _ng cây nh _phân

- ng d _ng gi i bài toán tìm ki m ki u nh _phân và duy t cây theo th t gi a.

- **Ng** **n** **x** **p** là danh sách các ph **n** **t** mà vi **c** **thêm** vào hay **l** **y** **ra** các ph **n** **t** ch **th** **c** hi **n** nh **ng** **n** **x** **p**.
- **Hàng** **i** là danh sách các ph **n** **t** mà vi **c** **thêm** ph **n** **t** c **th** **c** hi **n** **cu** **i** **hàng**, vi **c** **l** **y** **ra** ph **n** **t** **th** **c** hi **n** **u** **hàng**.
- **Cây** là c **u** **trúc** mà các ph **n** **t** óng vai trò là các **Nút** c **a** **cây**. Có **3 cách** khác nhau duy **t** các ph **n** **t** c **a** **cây**.