



**FPT POLYTECHNIC**



Bài 1:

T NG QUAN V C U TRÚC D LI U  
VÀ GI I THU T

[www.poly.edu.vn](http://www.poly.edu.vn)

[hoclaptrinhweb.com](http://hoclaptrinhweb.com)

# Mục tiêu bài học hôm nay

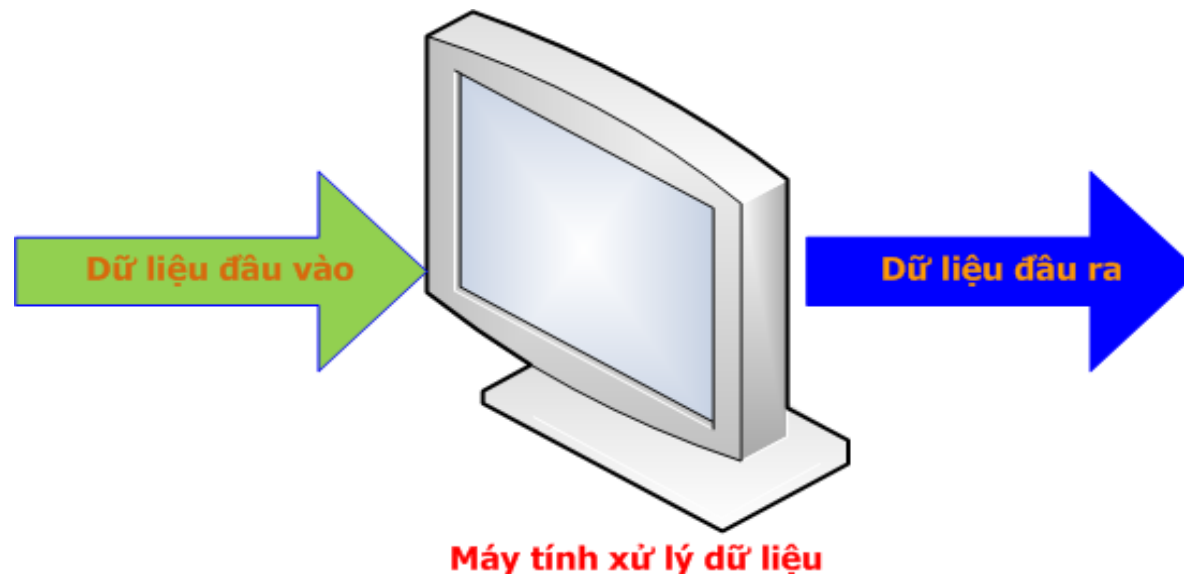
- Tìm hiểu khái niệm cấu trúc dữ liệu
  - Dữ liệu, Cấu trúc dữ liệu
  - Các kiểu cấu trúc dữ liệu
- Tìm hiểu khái niệm giải thuật (thuật toán, thuật giải)
  - Khái niệm về giải thuật
  - Biểu diễn giải thuật
  - Phân tích độ phức tạp giải thuật
- Mối liên hệ giữa cấu trúc dữ liệu và giải thuật

# Dữ liệu và Giải thuật

- Tại sao sử dụng máy tính xử lý dữ liệu
  - Nhanh hơn, chính xác hơn
  - Giải quyết nhiều bài toán đòi hỏi khả năng tính toán phức tạp, hoặc những bài toán phức tạp với khả năng dữ liệu lớn
- Phương pháp?
  - Nhập vào các thuật toán hiệu quả, thông minh -> chi phí thấp
  - Nhập vào số nâng cấp cấu hình máy -> chi phí cao

# Khái niệm Dữ liệu

- Trong tin học: Dữ liệu biểu u di n các thông tin c n thi t cho bài toán.
- Các d ì u máy tính g m: d ì u u vào, d ì u trung gian, d ì u u ra.



# Cấu trúc dữ liệu là gì?

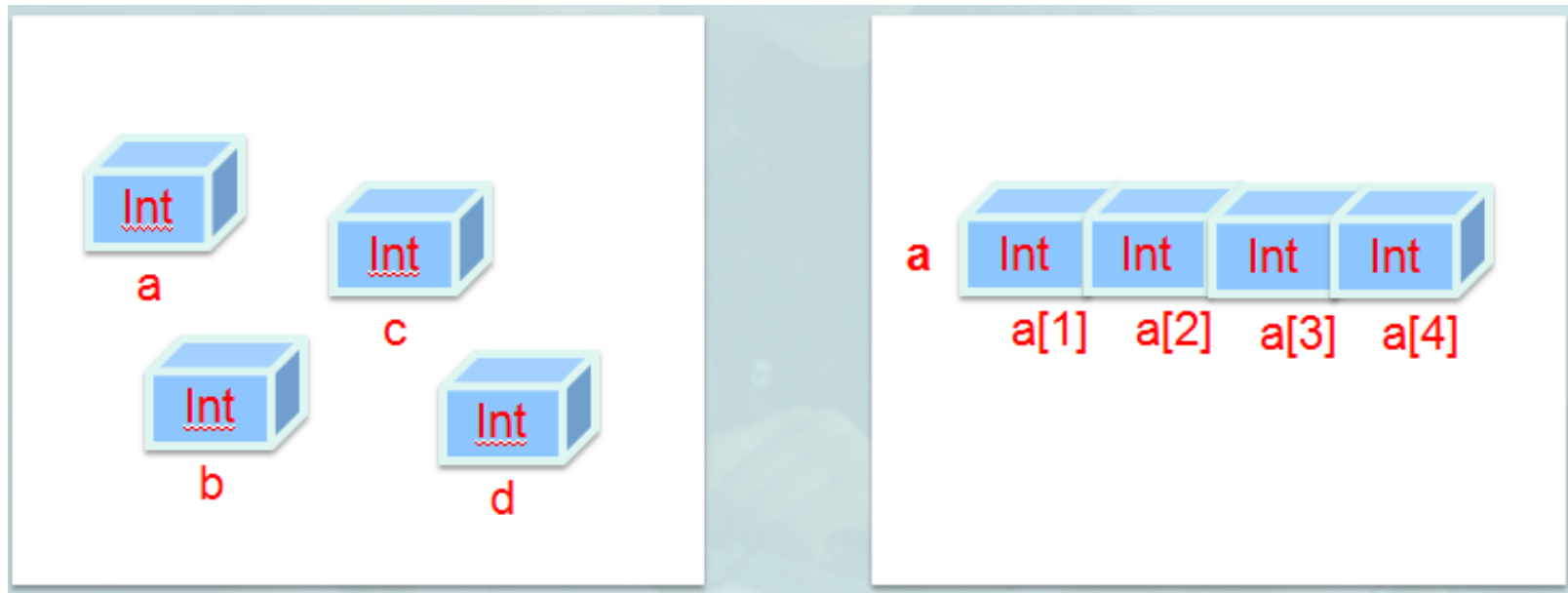


Hình các cấu trúc sách cha c t ch c,  
s p x p



Hình các cấu trúc sách ã c t ch c, s p  
x p

# Một ví dụ về Cấu trúc



Hình các s nguyên ch a c t ch c

Hình các s nguyên ã c t ch c  
trong m t m ng

# Khái niệm Cấu trúc dữ liệu

- Cấu trúc dữ liệu là gì?
  - Cấu trúc dữ liệu (data structure) là một phương thức để lưu trữ và tổ chức dữ liệu trong máy tính để xử lý hiệu quả.

# Các kiểu cấu trúc dữ liệu

- Dữ liệu không có cấu trúc (kiểu dữ liệu nguyên hay còn gọi là kiểu dữ liệu cơ sở):
  - Mối liên hệ giữa dữ liệu là một phần tử đơn lẻ
  - Ví dụ: Integer, Char, Boolean,...
- Dữ liệu có cấu trúc:
  - Các cấu trúc thành bởi các phần tử dữ liệu cơ sở
  - Ví dụ: Mảng (array), chuỗi (string), danh sách (collection), bản ghi (record), đối tượng (object)



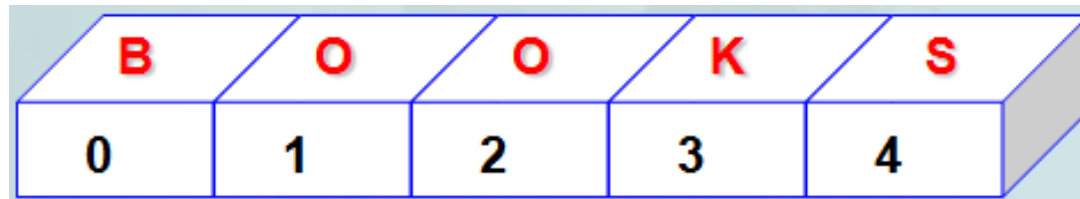
- Ví dụ: một số kiểu dữ liệu cơ bản trong Visual Basic:

| Tên kiểu       | Kích thước                                 | Miền giá trị  |
|----------------|--|---|
| <b>Byte</b>    | 1 byte                                     | 0 -> 255 (không dấu)  |
| <b>Boolean</b> | Tùy thuộc vào ngôn ngữ (thực ra là 1 byte) | True hoặc False   |
| <b>Integer</b> | 4 byte                                     | -2,147,483,648 -> 2,147,483,647 (có dấu)                                      |
| <b>Long</b>    | 8 byte                                     | -9,223,372,036,854,775,808 -> 9,223,372,036,854,775,807 (9.2...E+18) (có dấu) |
| <b>Date</b>    | 8 byte                                     | 0:00:00 ngày 1/1/0001 tới 11:59:59 ngày 31/12/9999                            |
| <b>Char</b>    | 2 byte                                     | 0 -> 65535 (không dấu)  |

# Kiểm li\_u có c\_u trúc

- Kiểm chu\_i kí\_t :

- Ví\_d : chu\_i kí\_t "BOOKS"



# Kiến trúc dữ liệu có cấu trúc

## ■ Kiểu mảng (array):

- Ví dụ mảng 1 chiều

|   |    |    |    |    |    |
|---|----|----|----|----|----|
| 4 | 14 | 22 | 38 | 27 | 15 |
| 0 | 1  | 2  | 3  | 4  | 5  |

- Ví dụ mảng 2 chiều

|   | 0      | 1      | 2      | 3      | 4      | 5      |
|---|--------|--------|--------|--------|--------|--------|
| 0 | [0][0] | [0][1] | [0][2] | [0][3] | [0][4] | [0][5] |
| 1 | [1][0] | [1][1] | [1][2] | [1][3] | [1][4] | [1][5] |
| 2 | [2][0] | [2][1] | [2][2] | [2][3] | [2][4] | [2][5] |

## Ví dụ cấu trúc dữ liệu

- Ví dụ về cấu trúc dữ liệu lưu trữ thông tin cho các chương trình máy tính có ý nghĩa rất quan trọng
- Ví dụ ta có một bảng thông tin như sau:

| H Tên    | Tuổi | SBD | Toán |
|----------|------|-----|------|
| Nguyễn A | 18   | 1A  | 10   |
| Trần B   | 19   | 2A  | 6    |
| Vũ D     | 18   | 3A  | 8    |

- Nếu ghép các dữ liệu trên cùng một cột thành cùng một cấu trúc thì ta có 4 mảng như sau:

|          |    |    |    |
|----------|----|----|----|
| Nguyễn A | 18 | 1A | 10 |
| Trần B   | 19 | 2A | 6  |
| Vũ D     | 18 | 3A | 8  |

## Ví dụ cấu trúc dữ liệu

- Nếu ghép các dữ liệu trên cùng một hàng lại thành một cấu trúc ta có cấu trúc bản ghi (Toàn bộ bản ghi là một mảng các bản ghi) như sau (cấu trúc kiểu file):

|                          |
|--------------------------|
| Nguyễn An   18   1A   10 |
| Trần B   19   2A   6     |
| V D   18   3A   8        |

## Ví dụ cấu trúc dữ liệu

- Nút chức năng định nghĩa (object) có 3 tính năng

| Sinh viên   |
|---|
| H tên: Nguyễn An<br>Tu i: 18<br>SBD: 1A<br>Toán: 10 |
| (các phép tính)                                     |

| Sinh viên                                       |
|---|
| H tên: Trần B<br>Tu i: 19<br>SBD: 2A<br>Toán: 6 |
| (các phép tính)                                 |

| Sinh viên                                     |
|---|
| H tên: Vũ D<br>Tu i: 18<br>SBD: 3A<br>Toán: 8 |
| (các phép tính)                               |

- M t CTDL t t ph i th a m ă n:
  - Ph n ă n h ứ n g th c t
  - Ph ứ h p v i c ă c th ă o t ă c trê n ó
  - T i t k i m t ă i n g u y ệ n h ă th n g

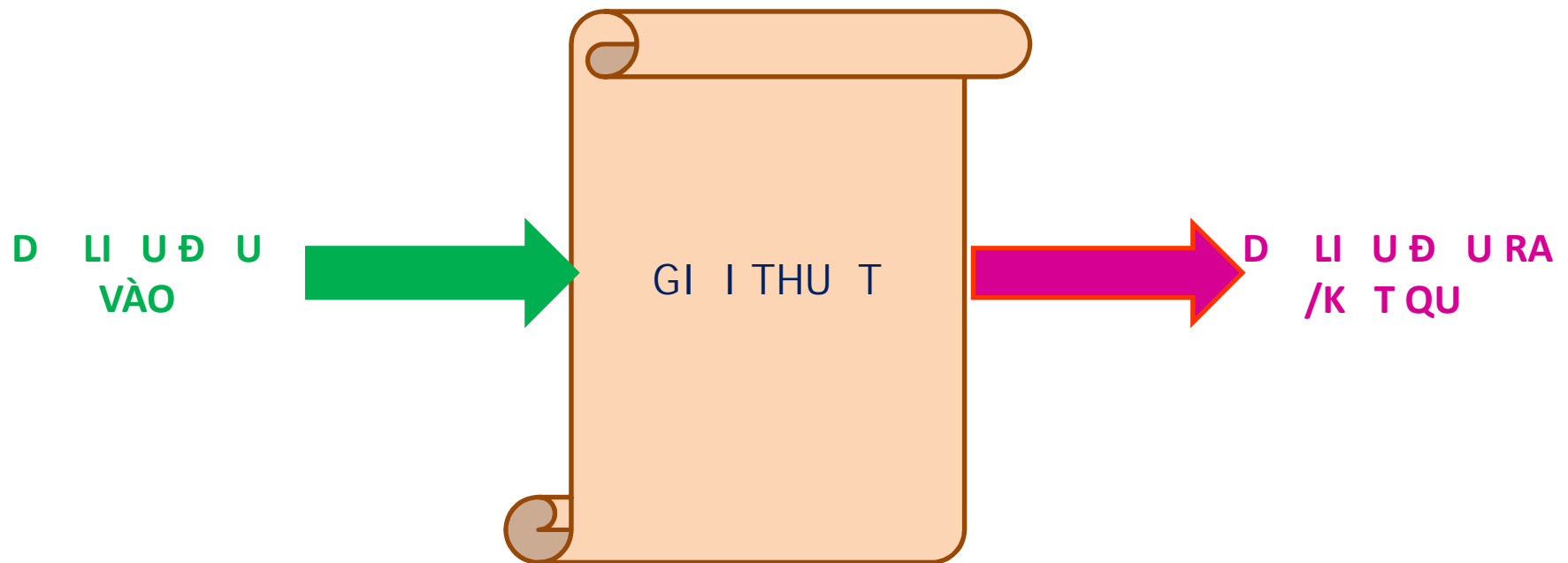
## Vai trò của cấu trúc dữ liệu

- CTDL đóng vai trò quan trọng trong việc kiểm soát thuật toán (còn gọi là thuật gì đó hay gì đó thuật) để đưa ra cách giải quyết bài toán.
- CTDL hỗ trợ cho các thuật toán thao tác trên dữ liệu có cấu trúc.



# Khái niệm ghi thu\_t

- Là t p h u h n có th t các b c tác ng lên d li u nào ó sau m t s h u h n l n th c hi n s cho ta k t qu .



## Các đặc trưng của giải thuật

- Có dữ liệu vào (Input)
- Có dữ liệu kết quả ra (Output)
- Tính Chính xác (Precision): Các bước của giải thuật mô tả chính xác.
- Tính Hữu hạn (Finiteness): Giải thuật phải trả ra sau một số hữu hạn bước với mọi dữ liệu vào.

## Các đặc trưng của giải thuật

- Tính đơn trị (Uniqueness): Các thuật ngữ trung gian của thuật toán thể hiện giải thuật xác định một cách đơn trị và chi phí thu nhập vào và các thuật ngữ của các bước thực hiện.
- Tính Tổng quát (Generality): Giải thuật có thể áp dụng giải nhiều bài toán có dạng mẫu cho.

# Các cách biểu diễn giải thuật

## ■ Các cách biểu diễn giải thuật:

- Ngôn ngữ tự nhiên
- Lưu đồ (flow chart)
- Mã giả (Pseudo code)
- Ngôn ngữ lập trình

## Biểu diễn ngôn ngữ tự nhiên

- Liệt kê thuật toán các bước ngôn ngữ tự nhiên biểu diễn thuật toán.
- Ưu điểm:
  - Đơn giản, không cần kiến thức về cách biểu diễn (mã nguồn, lưu trữ, ...)
- Nhược điểm:
  - Dài dòng, không cấu trúc.
  - Đôi lúc khó hiểu, không diễn đạt thuật toán.

## ■ Ngôn ngữ t a ngôn ngữ l p trình:

- Dùng cú trúc chu n hóa, ch ng h n t a Pascal, C.
- Dùng các ký hi u toán h c, bi n, hàm.

## ■ u i m:

- c ng k nh h n l u kh i.

## ■ Nh c i m:

- Không tr c quan b ng l u kh i.

## ■ Bài toán lu c tr ng:

B c 1: L y tr ng t gi

B c 2: un n c trong n i n khi sôi

B c 3: B tr ng vào n i n c

B c 4: i 3 phút cho tr ng chín

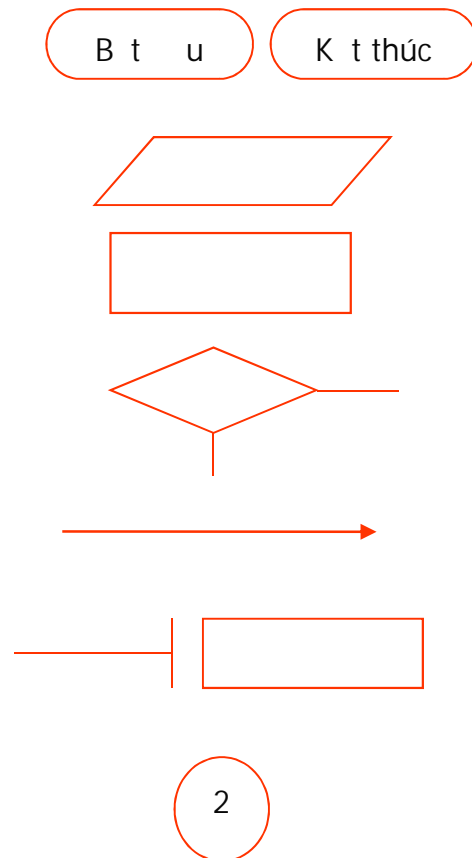
B c 5: V t tr ng ra

B c 6: Bóc v tr ng

B c 7: t vào a

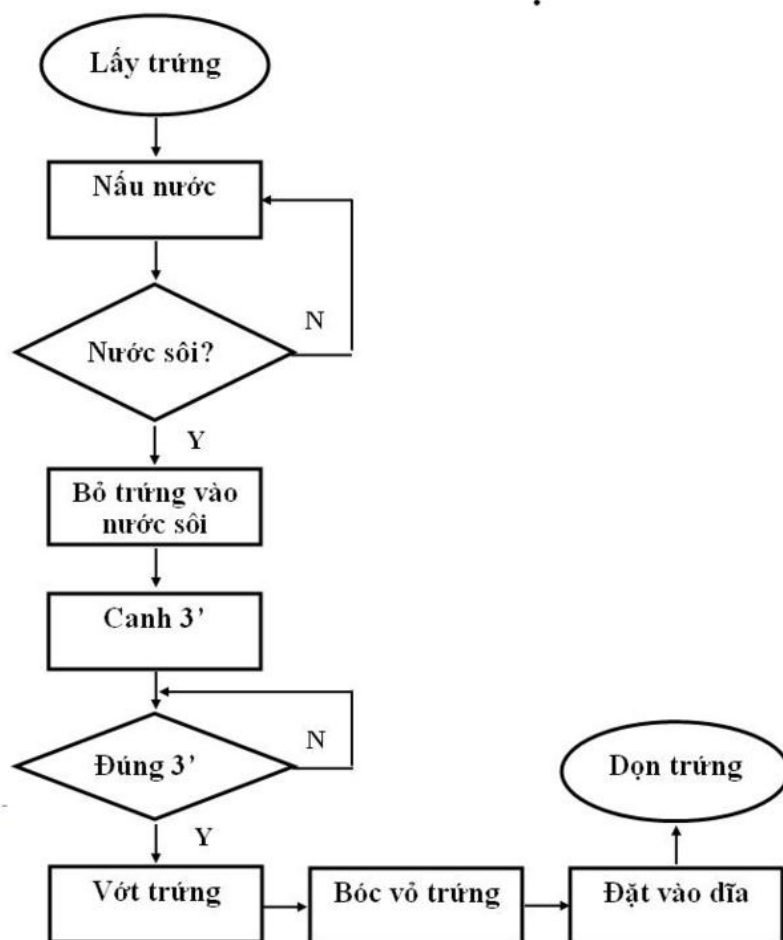
B c 8: D n v tr ng

# Biểu diễn bằng Lưu



- Hình thức / Kết thúc gì / Thu /
- Thao tác nhập / xuất dữ liệu
- Thao tác xử lý
- Lưu khi rẽ nhánh
- Kết nối trình
- Chú thích
- Ký hiệu kết nối cùng trang hay sang trang khác

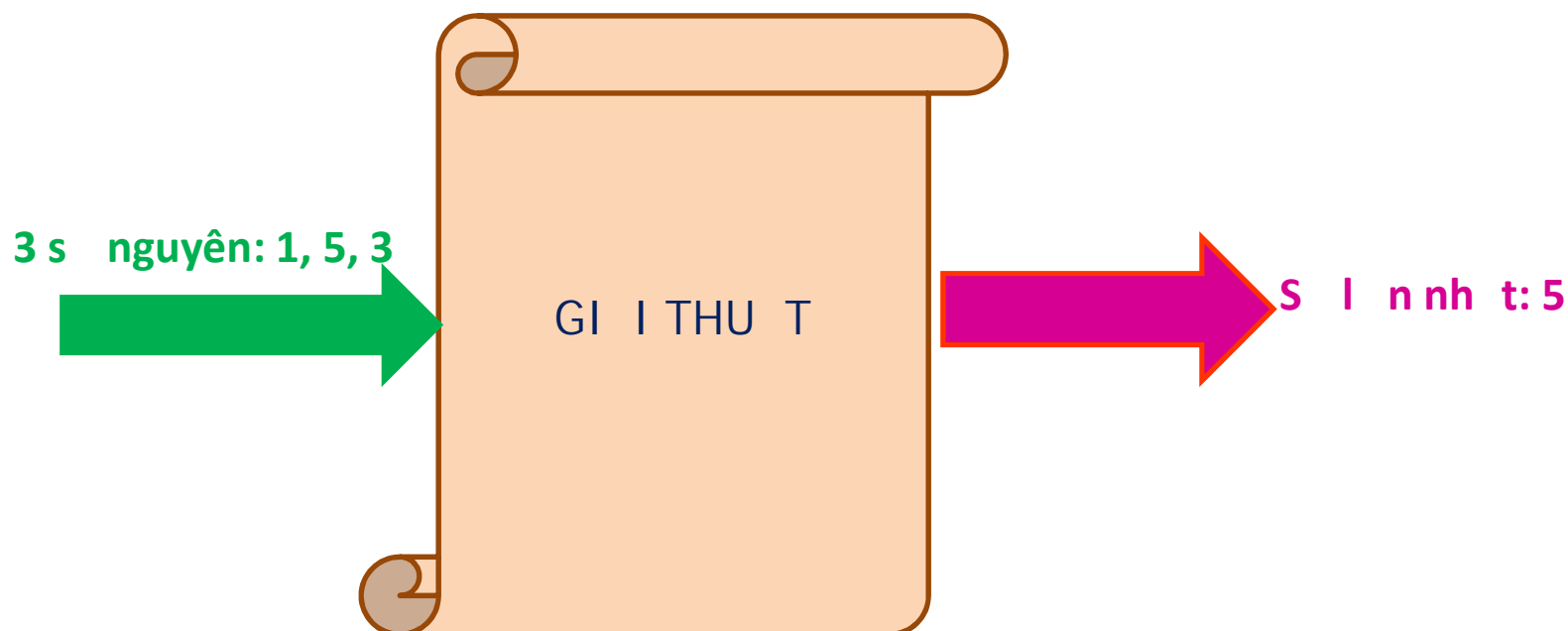




## Biểu diễn bằng ngôn ngữ lập trình

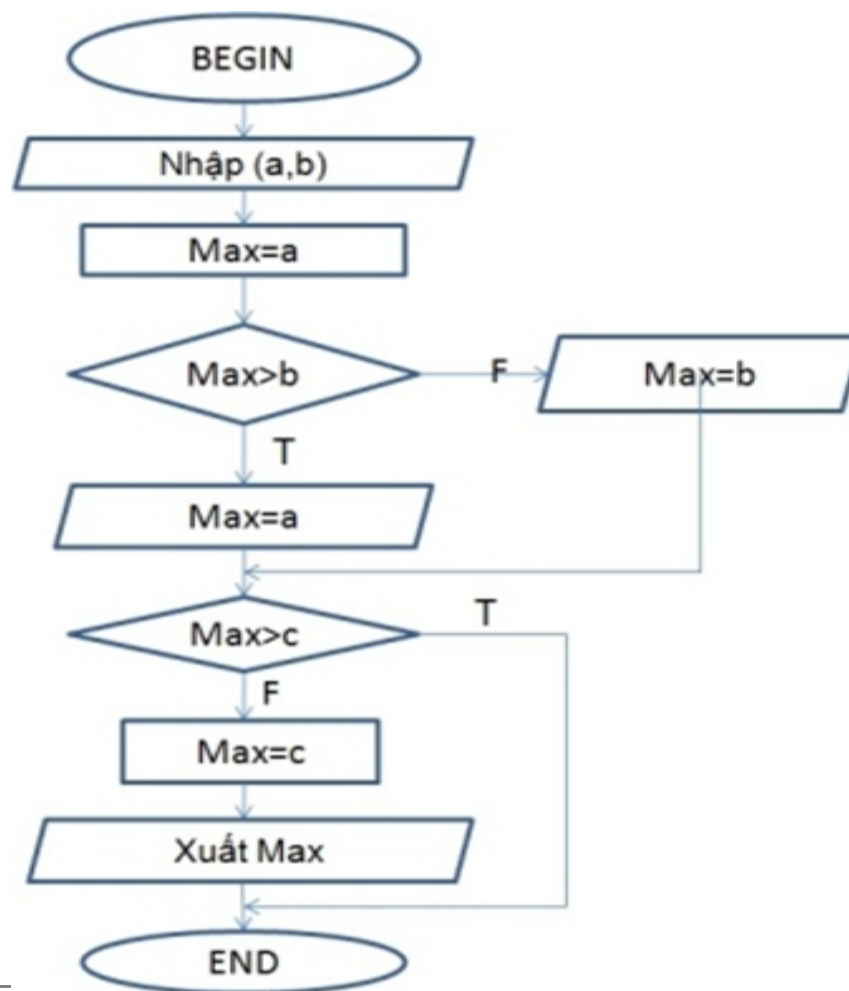
- Dùng ngôn ngữ máy tính (C, Pascal,...) để diễn tả thuật toán, CTDL thành câu lệnh.
- Có khả năng lập trình theo hình thức text và thực hành.
- Dùng phương pháp tinh chế để chuyển đổi bài toán sang mã chương trình cụ thể.

- Bài toán: Cho 3 số nguyên a, b, c. Mô tả giải thuật tìm số lớn nhất trong 3 số đã cho.
- Phân tích:
  - **Đầu vào:** 3 số nguyên a, b, c.
  - **Đầu ra:** số lớn nhất trong 3 số.



- Dùng ngôn ngữ tự nhiên mô tả giải thuật tìm số lớn nhất trong 3 số  $a, b, c$ :
  - Bước 1. Gán  $\text{max} = a$ ;
  - Bước 2. Nếu  $b > \text{max}$  thì gán  $\text{max} = b$ ;
  - Bước 3. Nếu  $c > \text{max}$  thì gán  $\text{max} = c$ ;

- Dùng lưu mô tả thuật toán tìm số lớn nhất trong 3 số a, b, c:



Theo dõi quá trình thực hiện các thuật toán với giá trị của a, b, c.

- $a := 1;$
- $b := 5;$
- $c := 3;$

 $a = 1$ 
 $b = 5$ 
 $c = 3$ 

Bước 1. Gán giá trị của a vào biến max

 $\text{max} = 1$ 

Bước 2. Do  $b > \text{max}$  ( $5 > 1$ ) nên max gán bằng b

 $\text{max} = 5$ 

Bước 3. Do  $c < \text{max}$  ( $3 < 5$ ) nên không thực hiện gán

 $\text{max} = 5$

■ Một vài nhận xét:

- Giá trị thu được có **tính chính xác**: Các bước của giá trị thu được mô tả chính xác.
- Giá trị thu được có **tính duy nhất**: với cùng vào đã xác định, kết quả tìm kiếm của giá trị thu được xác định duy nhất
- Giá trị thu được có **tính hữu hạn**: Giá trị thu được kết thúc sau 3 bước và đưa ra lời giải của bài toán
- Giá trị thu được có **tính tổng quát**: luôn đưa ra giá trị của số lần nhúng trong 3 số bất kỳ

## ph c t\_p gi i thu\_t

### ■ M t thu t toán hi u qu :

- Chi phí c n s d ng tài nguyên th p: B nh , th i gian s d ng CPU, ...

### ■ Ph ng pháp ánh giá ph c t\_p c a gi i thu t:

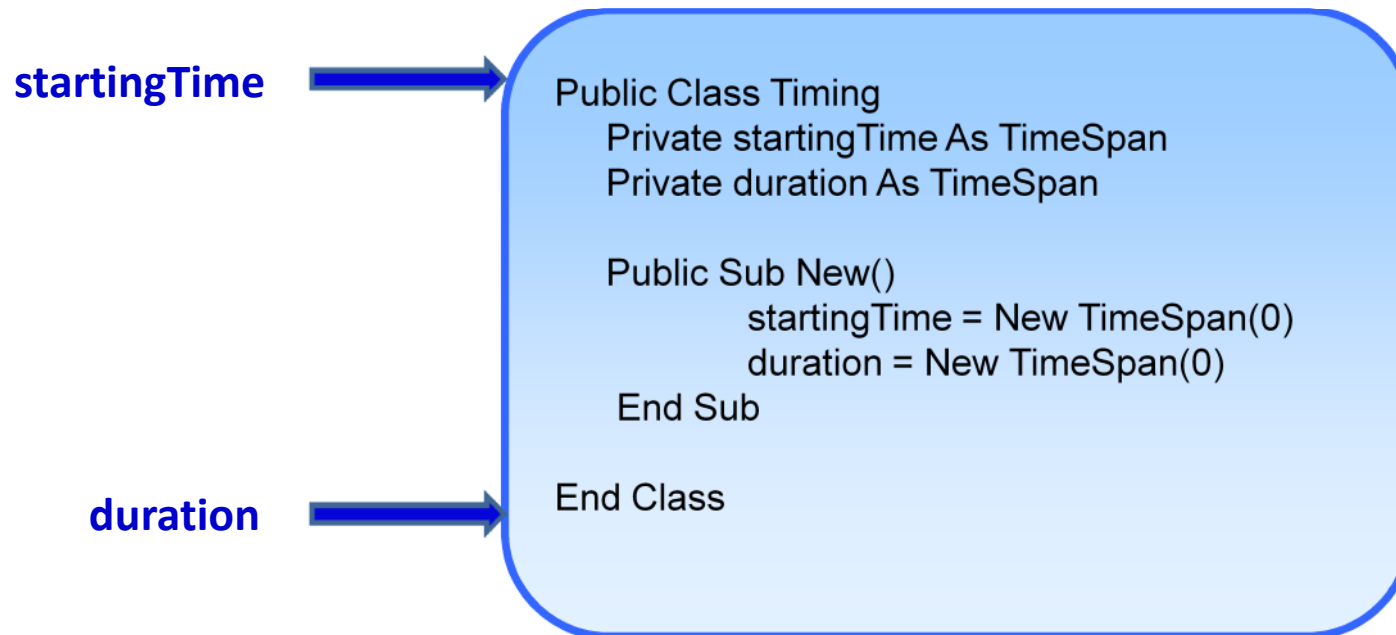
- D a trên th i gian th c hi n gi i thu t n khi ra c k t qu .
- D a trên s phép toán th c hi n gi i thu t



## Tính thời gian thực hiện ghi nhận

- Thời gian thực hiện ghi nhận hoàn toàn có thể đo bằng cách lập trình.
- Ví dụ cài đặt lập Timing trong VB.Net:
  - Cho phép tính thời gian thực hiện của code chạy trong chương trình
  - Lập Timing có 2 thành phần dữ liệu:
    - **startingTime**: Lưu trữ thời gian bắt đầu thực hiện của code
    - **duration**: thời gian kết thúc của code
    - **Thời gian thực hiện** = **startingTime** - **duration**

# Tính thời gian thực hiện ghi dữ liệu



- $n$  - Kích thước vào của dữ liệu
- Mô tả phức tạp thuật toán qua một hàm  $O(n)$
- Hai nguyên tắc đánh giá:
  - Nguyên tắc cộng
  - Nguyên tắc nhân

■ Ví dụ :

```
1 i = 1
2 loop ( i <= 1000 )
    1 application code
    2 i = i + 1
```

$O(n) = 1000$

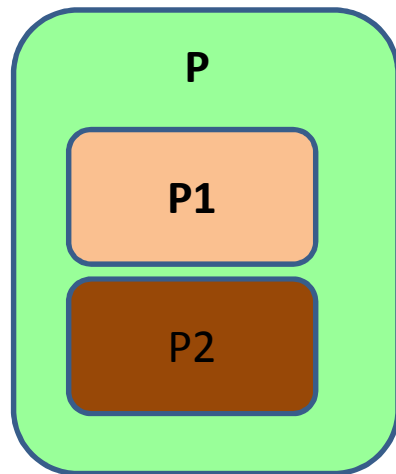
```
1 i = 1
2 loop ( i <= 1000 )
    1 application code
    2 i = i + 2
```

$O(n) = 500$

# Tính số phép toán thực hiện giới thiệu

## ■ Nguyên tắc cộng:

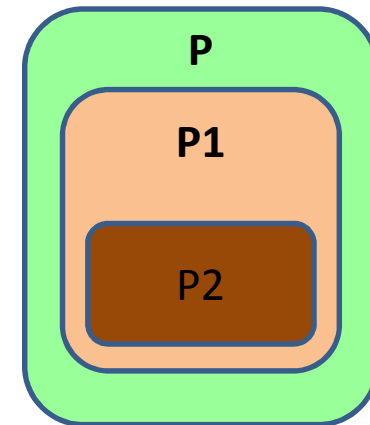
- **Nếu:** chương trình P gồm 2  
o n chương trình P1  
( $O_1(n)$ ) và P2 ( $O_2(n)$ ) ch y  
n i ti p nhau



- **Thời:** ph c t p c a CT P:  
 $O(n) = \max(O_1(n), O_2(n))$

## ■ Nguyên tắc nhân:

- **Nếu:** chương trình P gồm 2  
o n chương trình P1  
( $O_1(n)$ ) và P2 ( $O_2(n)$ ) l ng  
nhau



- **Thời:** ph c t p c a CT P:  
 $O(n) = O_1(n).O_2(n)$

- Hằng số :  $O(c)$
- $\log N$  :  $O(\log N)$
- $N$  :  $O(N)$
- $N \log N$  :  $O(N \log N)$
- $N^2$  :  $O(N^2)$
- $N^3$  :  $O(N^3)$
- $2^N$  :  $O(2^N)$
- $N!$  :  $O(N!)$



ph c t p t ng d n

# Mối liên hệ giữa CTDL và GT

- Ghi nhớ thuật = phép xử lý.
- Ý tưởng của ghi nhớ thuật chính là dùng lưu trữ để chia thành các cấu trúc.

- CTDL & GT gắn chặt với nhau. Niklaus Wirth đã từng nói:

Cấu trúc dữ liệu + Ghi nhớ thuật = Chương trình

- Nếu thay đổi cấu trúc dữ liệu thì ghi nhớ thuật cũng thay đổi theo.

# Mối liên hệ giữa CTDL và GT

- Ví dụ quản lý danh bạ điện thoại
- Dữ liệu gồm:
  - Họ và tên
  - Số điện thoại

| Họ và tên    | Số T      |
|--------------|-----------|
| Nguyễn Văn D | 098123456 |
| Võ Thị B     | 091557799 |
| Trần Xuân A  | 090333999 |
| Quách Thái C | 093886868 |



# Mối liên hệ giữa CTDL và GT

## ■ Bài toán tìm kiếm thông tin theo họ tên:

- Nếu danh bạ không có tổ chức gì cả thì đơn giản nhất là tìm tu n t t u n cụ i
- Nếu danh bạ (H và tên) **t ch c theo th t** a,b,c thì ta có th tìm ki m theo gi i thu t tìm ki m nh phân (tìm ki u t i n: chia ôi và tìm trên n a có ch a k t qu )-> **th i gian tìm ki m nhanh h n.**

| H và tên     | S T       |
|--------------|-----------|
| Nguyễn Văn D | 098123456 |
| V Th B       | 091557799 |
| Tr n Xuân A  | 090333999 |
| Quách Thái C | 093886868 |

Danh bạ ch a s p x p

| H và tên     | S T       |
|--------------|-----------|
| Tr n Xuân A  | 090333999 |
| V Th B       | 091557799 |
| Quách Thái C | 093886868 |
| Nguyễn Văn D | 098123456 |

Danh bạ đã s p x p

# Mối liên hệ giữa CTDL và GT

- **Nu:** danh bạ và xếp thứ tự và có một bảng mục lục
    - A - Trang 10
    - B - Trang 40
    - C - Trang 100
  - **Thì:** Tìm trong bảng mục lục trực tiếp. Nếu thì chỉ cần tìm trong một vài n
- > thời gian thực hiện bài toán nhanh hơn.

| H và tên     | S T       |
|--------------|-----------|
| Trần Xuân A  | 098123456 |
| Vũ Thị A     | 091662288 |
| Quách Thái A | 090333999 |
| Nguyễn Văn A | 093886668 |

| H và tên     | S T       |
|--------------|-----------|
| Trần Xuân B  | 012321432 |
| Vũ Thị B     | 094325325 |
| Quách Thái B | 091987412 |
| Nguyễn Văn B | 096666666 |

# Hướng tiếp cận CTDL&GT trong môn này

- Hướng tiếp cận thông thường: tìm hiểu về mặt lý thuyết và chi tiết cài đặt các cấu trúc dữ liệu và giải thuật
  - Điểm mạnh: hiểu sâu về bản chất, có thể xây dựng CTDL&GT cho những bài toán phức tạp
  - Điểm yếu: khó hiểu, tính năng động thấp đòi hỏi sinh viên nắm vững các lý thuyết và sâu về công cụ lập trình
- Hướng tiếp cận mới:
  - Điểm mạnh:
    - Hiểu các khái niệm CTDL&GT
    - Tạo dựng ngay các cấu trúc CTDL phức tạp nhờ các API của ngôn ngữ lập trình cài đặt sẵn
    - Áp dụng vào ứng dụng ngay lập tức
  - Điểm yếu:
    - Phức tạp về mặt ngôn ngữ lập trình phức tạp
    - Hạn chế trong việc có thể cài đặt CTDL&GT mới

# Hình thức tiếp cận CTDL&GT trong môn này

- Trong thực tế, các CTDL có sẵn trong API của ngôn ngữ lập trình đáp ứng gần như 90% nhu cầu về lập trình với CTDL&GT thực tế.
- Môn này tiếp cận CTDL&GT theo hình thức thực tiễn, và giới thiệu lý thuyết, hình thức cài đặt, và hình thức sử dụng các API có sẵn.
- Chẳng hạn, các CTDL của API ngôn ngữ lập trình VB.NET (bản 2008) sẽ được sử dụng minh họa cho môn này.

- Cấu trúc dữ liệu là cách tổ chức các dữ liệu thành các cấu trúc như mảng, chuỗi, file...
- Giải thuật là tập hợp các bước tác động lên dữ liệu nào đó sau một số bước nhằm tìm ra kết quả.
- Việc lựa chọn cấu trúc dữ liệu và giải thuật cho bài toán là rất quan trọng.