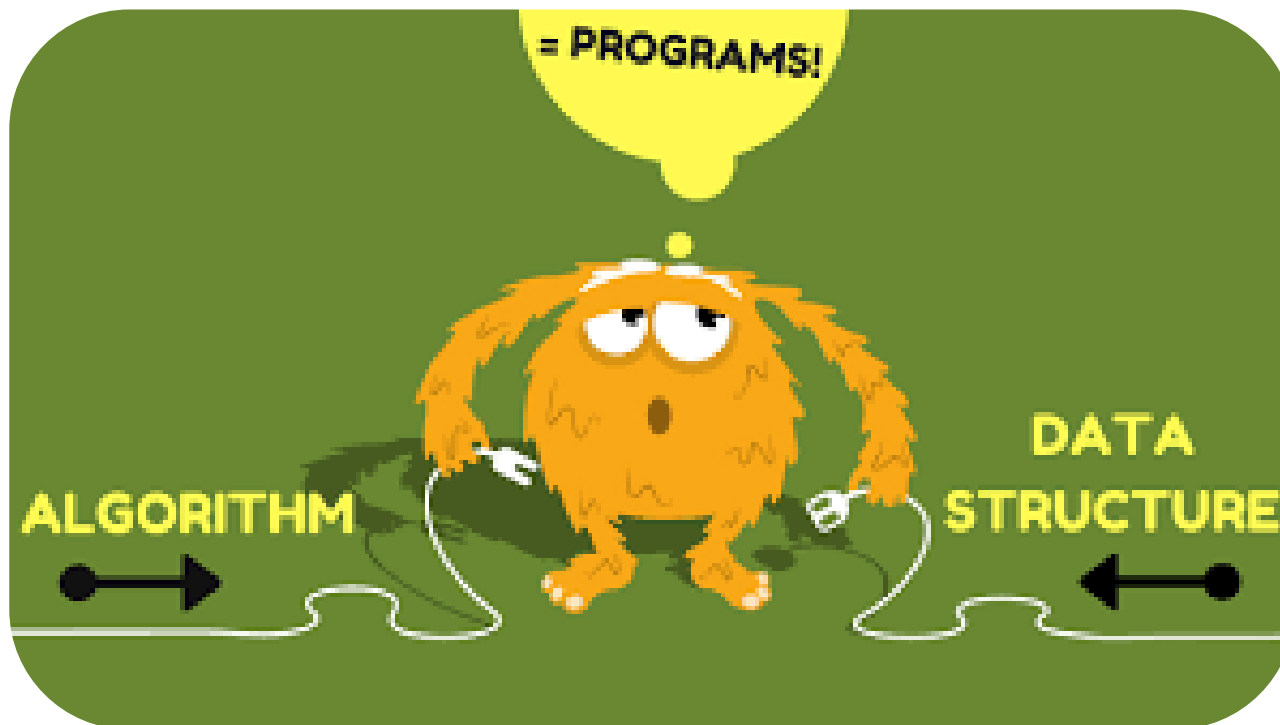


TỔNG QUAN CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT



MỤC TIÊU BÀI HỌC

- Tìm hiểu khái niệm cấu trúc dữ liệu.
- Tìm hiểu khái niệm giải thuật.
- Mối liên hệ giữa cấu trúc dữ liệu và giải thuật.



1

DỮ LIỆU VÀ CẤU TRÚC DỮ LIỆU



➤ **Tại sao sử dụng máy tính để xử lý dữ liệu**

- ✓ Nhanh hơn, chính xác hơn.
- ✓ Giải quyết nhiều bài toán đòi hỏi khối lượng tính toán cực lớn, hoặc những bài toán phức tạp với khối lượng dữ liệu lớn.

➤ **Phương pháp?**

- ✓ Nhờ vào các thuật toán hiệu quả, thông minh -> **chi phí thấp**.

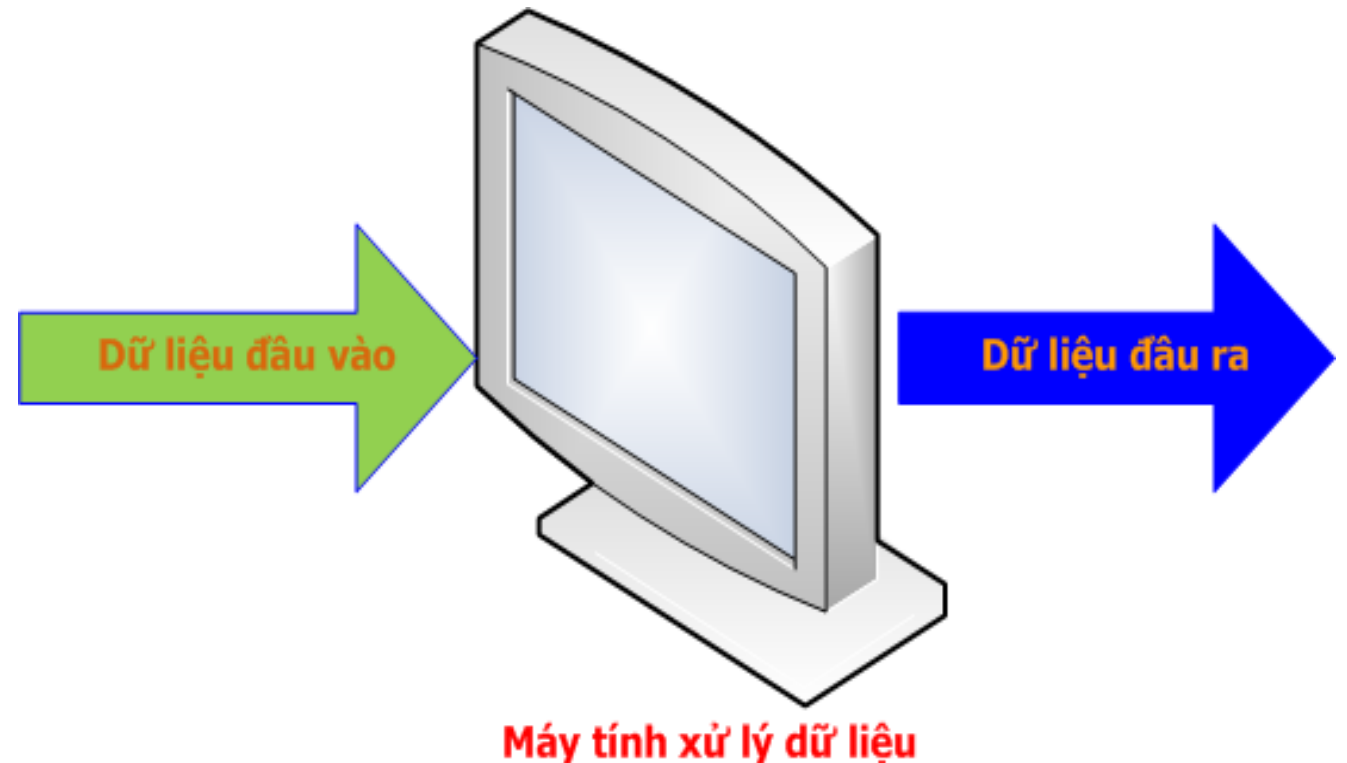
2

CÁC KIỂU CẤU TRÚC DỮ LIỆU



KHÁI NIỆM DỮ LIỆU

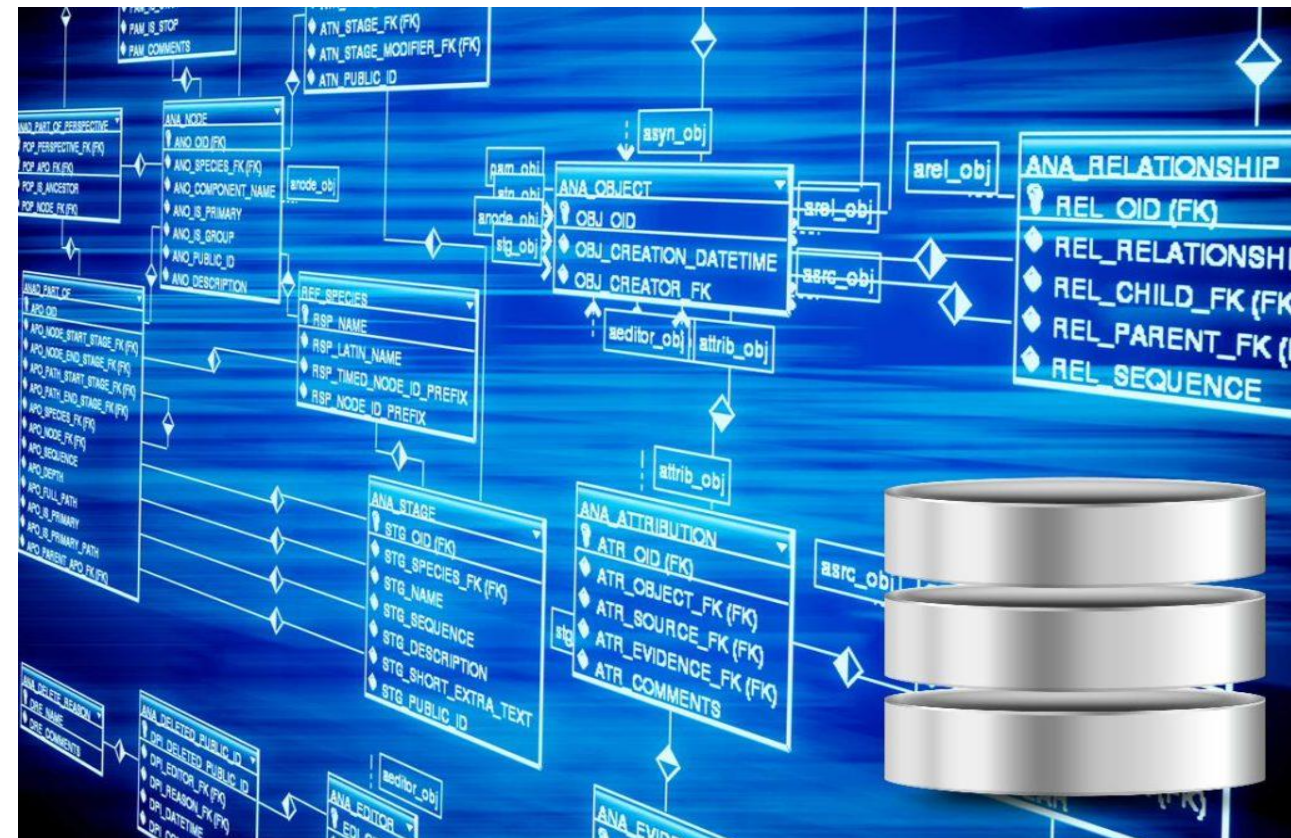
- Trong tin học: Dữ liệu để biểu diễn các thông tin cần thiết cho bài toán.
- Các dữ liệu máy tính gồm: dữ liệu đầu vào, dữ liệu đầu ra.



KHÁI NIỆM CẤU TRÚC DỮ LIỆU

➤ Cấu trúc dữ liệu là gì?

- ✓ Cấu trúc dữ liệu (data structure) là một phương thức cụ thể để **lưu trữ** và **tổ chức** dữ liệu trong máy tính để việc xử lý **hiệu quả**.



MỘT VÍ DỤ VỀ CẤU TRÚC

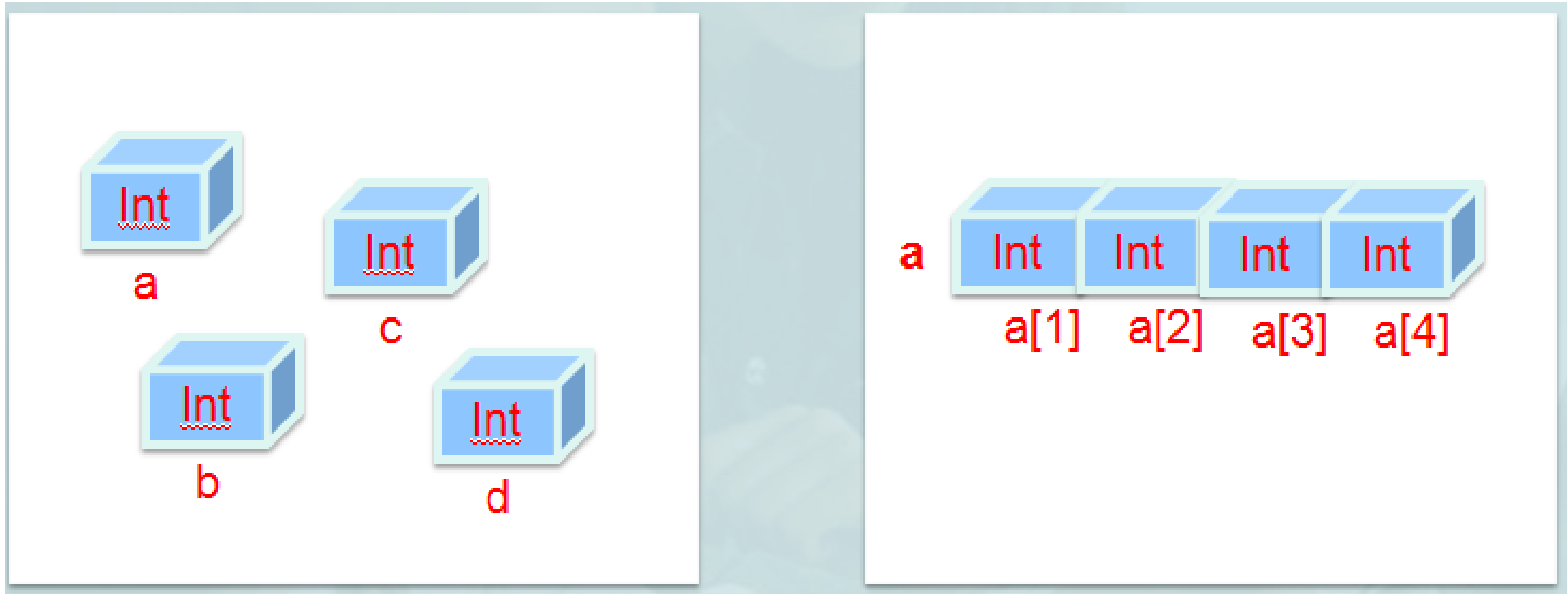


Hình 1: các cuốn sách chưa được tổ chức, sắp xếp



Hình 2: các cuốn sách đã được tổ chức, sắp xếp

MỘT VÍ DỤ VỀ CẤU TRÚC



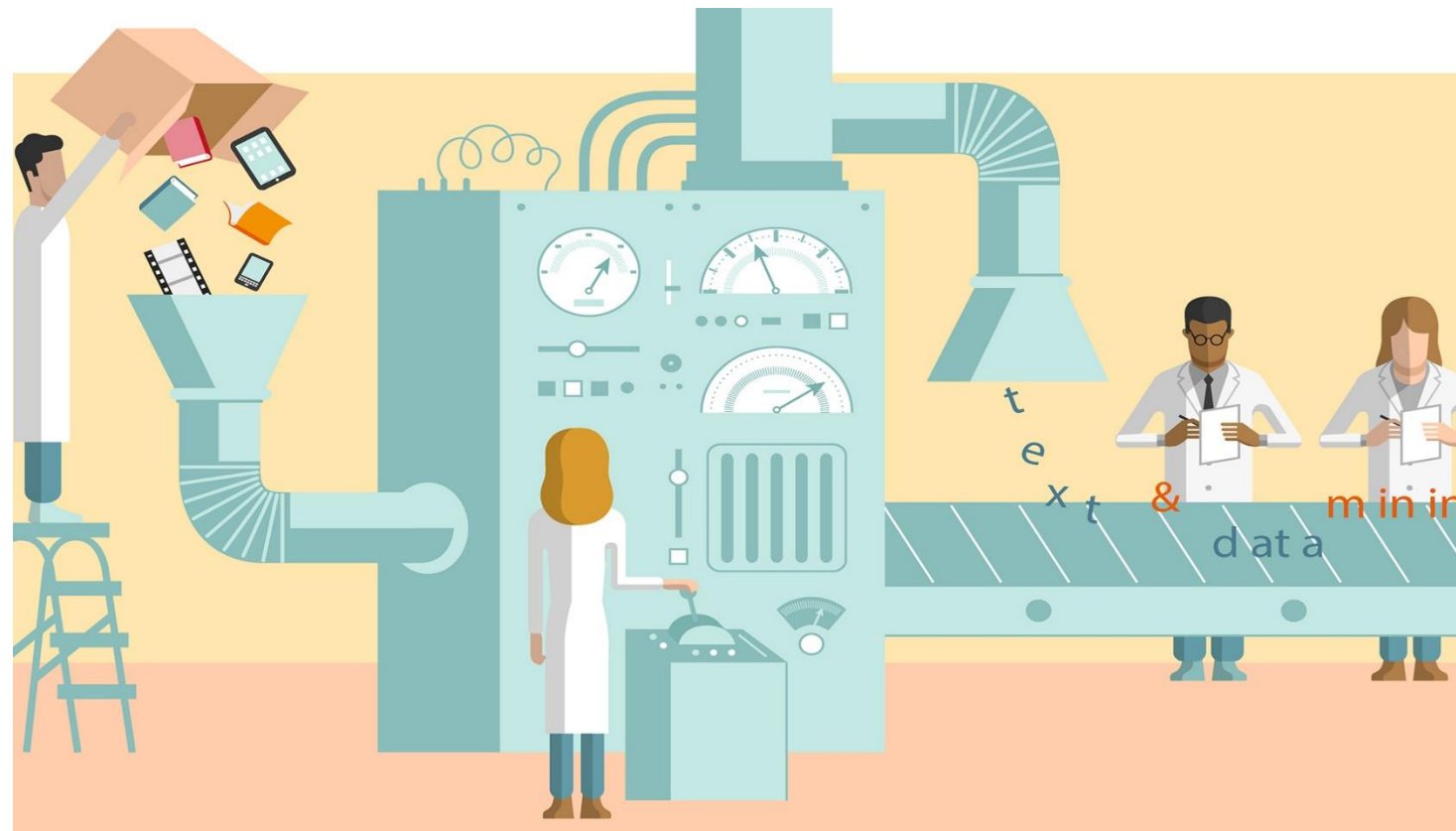
Hình 1: các số nguyên chưa được tổ chức

Hình 2: các số nguyên đã được tổ chức trong một mảng

KHÁI NIỆM CẤU TRÚC DỮ LIỆU

■ Cấu trúc dữ liệu là gì?

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



CÁC KIỂU CẤU TRÚC DỮ LIỆU

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

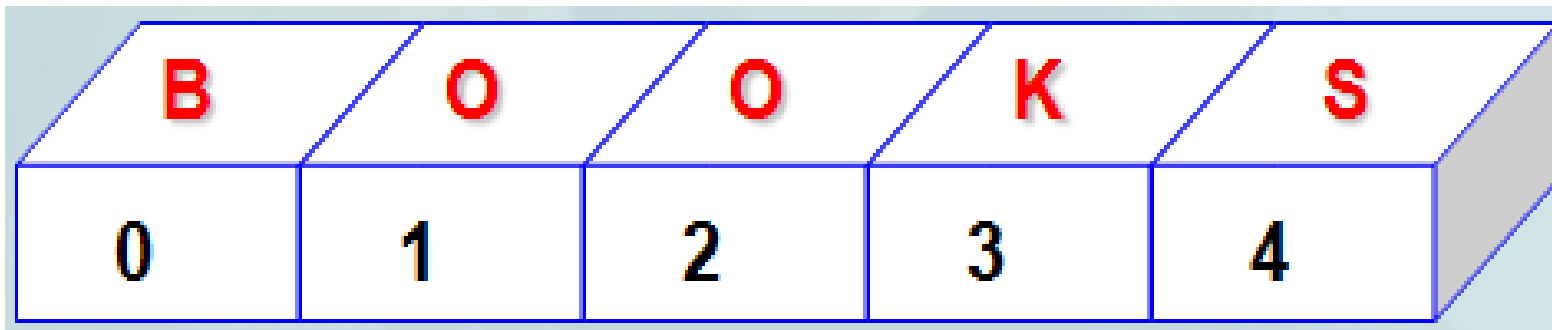
➤ Ví dụ: một số kiểu dữ liệu cơ sở được định nghĩa trong Visual Basic:

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

Kiểu dữ liệu có cấu trúc

➤ Kiểu chuỗi kí tự:

- Ví dụ: chuỗi kí tự "BOOKS"



KIỂU 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

- Việc tổ chức CTDL để lưu trữ dữ liệu phục vụ cho các chương trình máy tính có ý nghĩa rất quan trọng.
- Ví dụ ta có một bảng thông tin như sau:

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

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

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

VÍ DỤ CẤU TRÚC DỮ LIỆU

➤ Nếu tổ chức dưới dạng đối tượng (object) sẽ có 3 đối tượng

Sinh viên
Họ tên: Nguyễn An Tuổi: 18 SBD: 1A Toán: 10
(các phương thức)

Sinh viên
Họ tên: Trần B Tuổi: 19 SBD: 2A Toán: 6
(các phương thức)

Sinh viên
Họ tên: Vũ D Tuổi: 18 SBD: 3A Toán: 8
(các phương thức)

TIÊU CHUẨN CỦA CẤU TRÚC DỮ LIỆU

➤ Một CTDL tốt phải thỏa mãn:

- Phản ánh đúng thực tế
- Phù hợp với các thao tác trên đó
- Tiết kiệm tài nguyên hệ thống



VAI TRÒ CỦA CẤU TRÚC DỮ LIỆU

- CTDL đóng vai trò quan trọng trong việc kết hợp thuật toán (còn gọi là thuật giải hay giải thuật) để đưa ra cách giải quyết bài toán.
- CTDL hỗ trợ cho các thuật toán thao tác trên đối tượng được hiệu quả hơn.



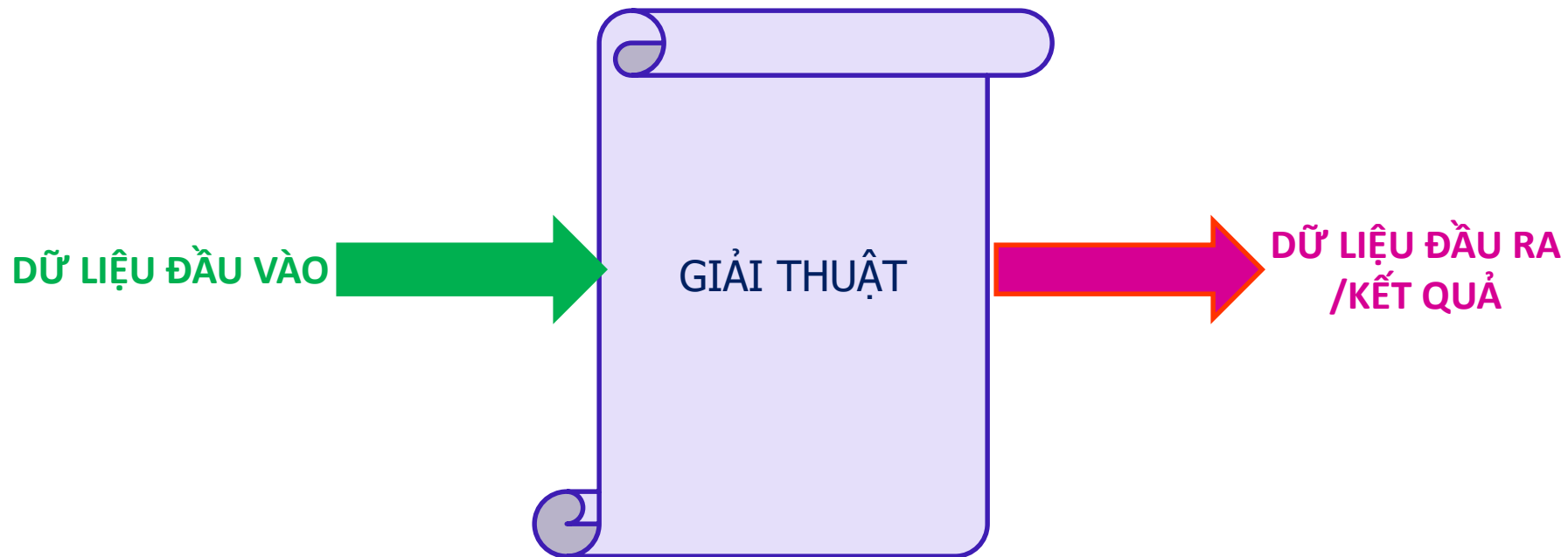
3

KHÁI NIỆM VỀ GIẢI THUẬT



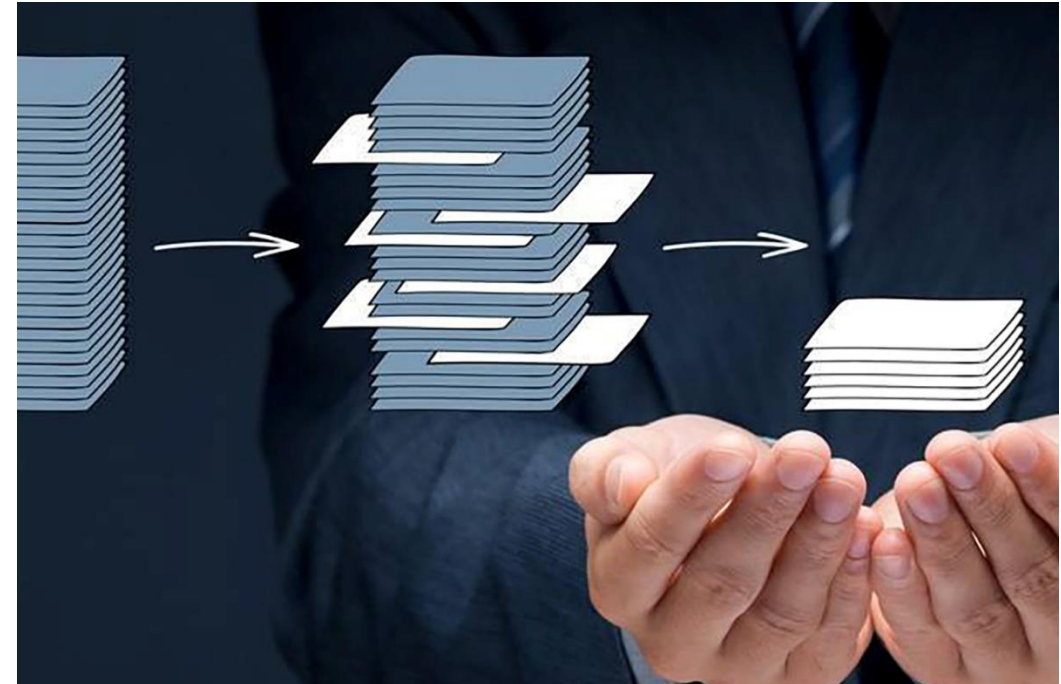
KHÁI NIỆM GIẢI THUẬT

- Là tập hữu hạn có thứ tự các bước tác động lên dữ liệu nào đó để sau một số hữu hạn lần thực hiện sẽ cho ta kết quả.



CÁC ĐẶC TRƯNG CỦA GIẢI THUẬT

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



CÁC ĐẶC TRƯNG CỦA GIẢI THUẬT

- **Tính Đơn trị:** Các kết quả trung gian của từng bước thực hiện giải thuật được xác định một cách đơn trị và chỉ phụ thuộc đầu vào và các kết quả của các bước trước.
- **Tính Tổng quát:** Giải thuật có thể áp dụng để giải mọi bài toán có dạng đã cho.



4

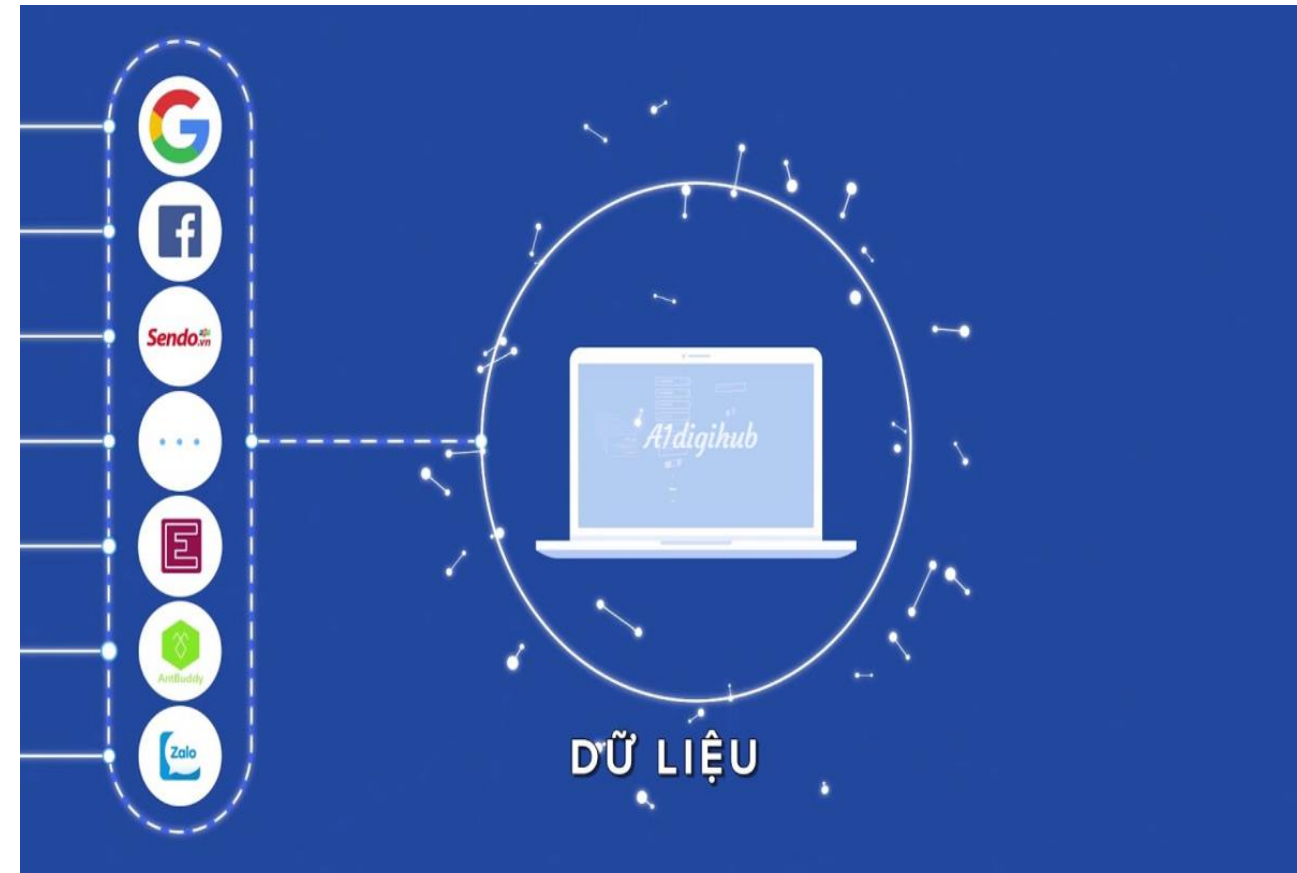
CÁC CÁCH BIỂU DIỄN GIẢI THUẬT



CÁC CÁCH BIỂU DIỄN GIẢI THUẬT

➤ Các cách biểu diễn giải thuật:

- ✓ Ngôn ngữ tự nhiên
- ✓ Lưu đồ
- ✓ Mã giả Ngôn ngữ lập trình



BIỂU DIỄN BẰNG NGÔN NGỮ TỰ NHIÊN

➤ Liệt kê tuần tự các bước bằng ngôn ngữ tự nhiên để biểu diễn thuật toán.

➤ Ưu điểm:

✓ Đơn giản, không cần kiến thức về cách biểu diễn (mã giả, lưu đồ,...)

➤ Nhược điểm:

✓ Dài dòng, không cấu trúc.

✓ Đôi lúc khó hiểu, không diễn đạt được thuật toán.

• 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 $\text{max} = c$;

➤ Ngôn ngữ tựa ngôn ngữ lập trình:

- ✓ Dùng cấu trúc chuẩn hóa, chẳng hạn tựa Pascal, C.
- ✓ Dùng các ký hiệu toán học, biến, hàm.

➤ Ưu điểm:

- ✓ Dễ công kênh hơn lưu đồ khối.

➤ Nhược điểm:

- ✓ Không trực quan bằng lưu đồ khối.

- Là một sự kết hợp giữa ngôn ngữ tự nhiên với các cấu trúc câu lệnh của một ngôn ngữ lập trình.
- Ví dụ: Giải thuật giải phương trình bậc nhất $ax+b=0$.
 - Nhập vào a, b
 - If $a==0$ then
 - If $b==0$ then
Kết luận phương trình vô số nghiệm
 - else
Kết luận phương trình vô nghiệm
 - else
Kết luận phương trình có nghiệm $x=-b/a$

➤ Bài toán luộc trứng:

Bước 1: Lấy trứng từ giỏ

Bước 2: Đun nước trong nồi đến khi sôi

Bước 3: Bỏ trứng vào nồi nước

Bước 4: Đợi 3 phút cho trứng chín

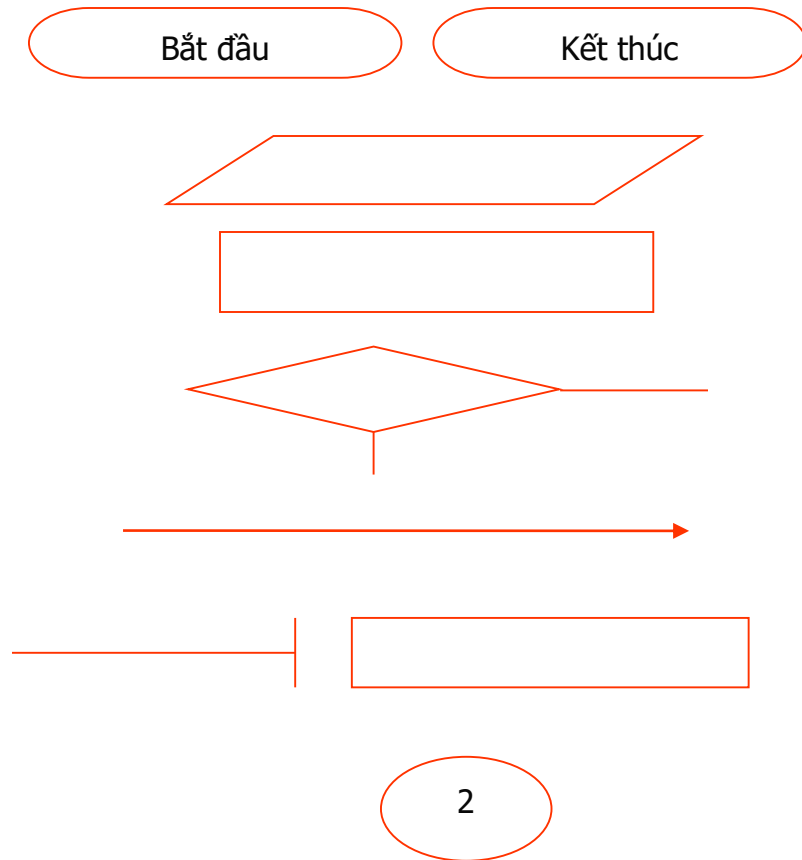
Bước 5: Vớt trứng ra

Bước 6: Bóc vỏ trứng

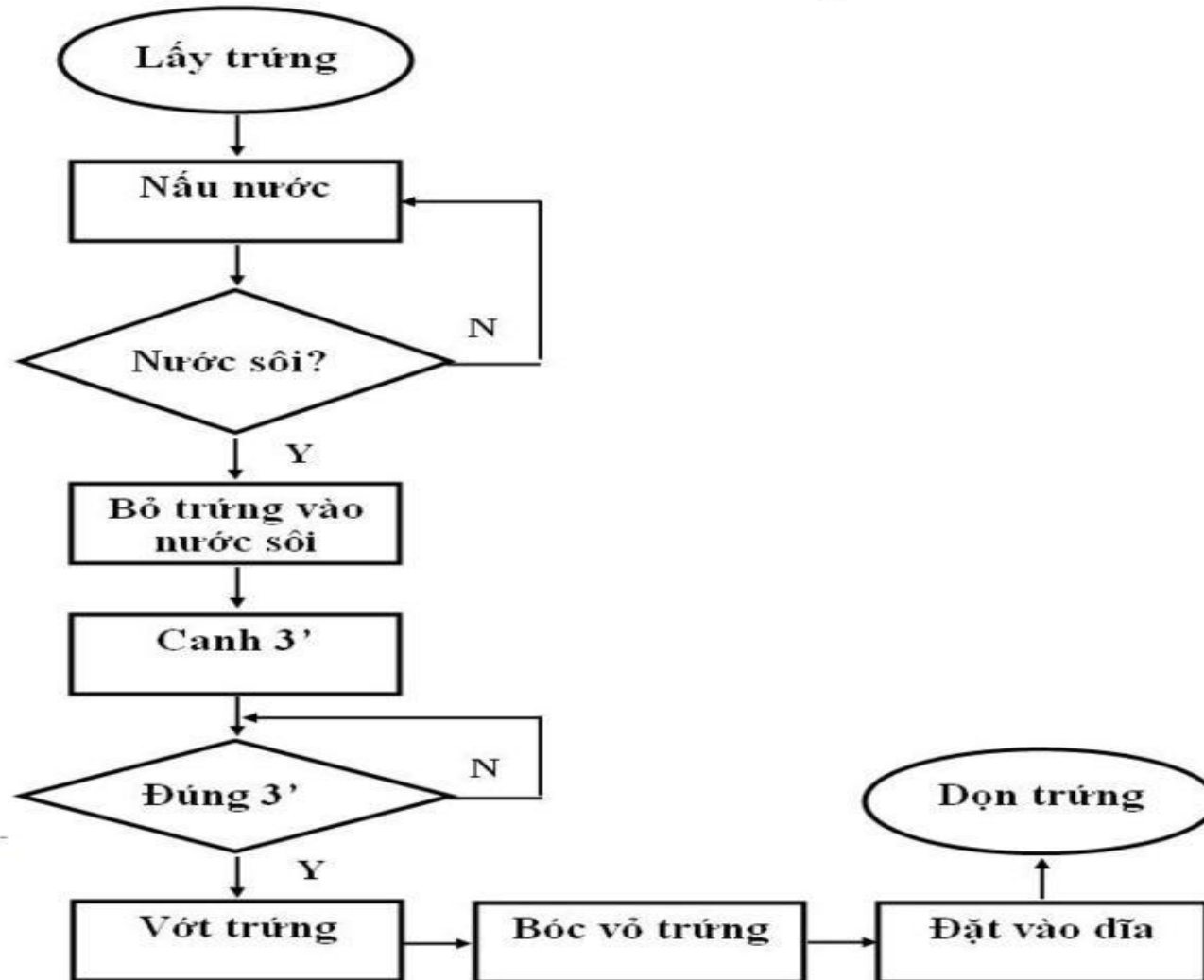
Bước 7: Đặt vào đĩa

Bước 8: Dọn vỏ trứng

BIỂU DIỄN BẰNG LƯU ĐỒ

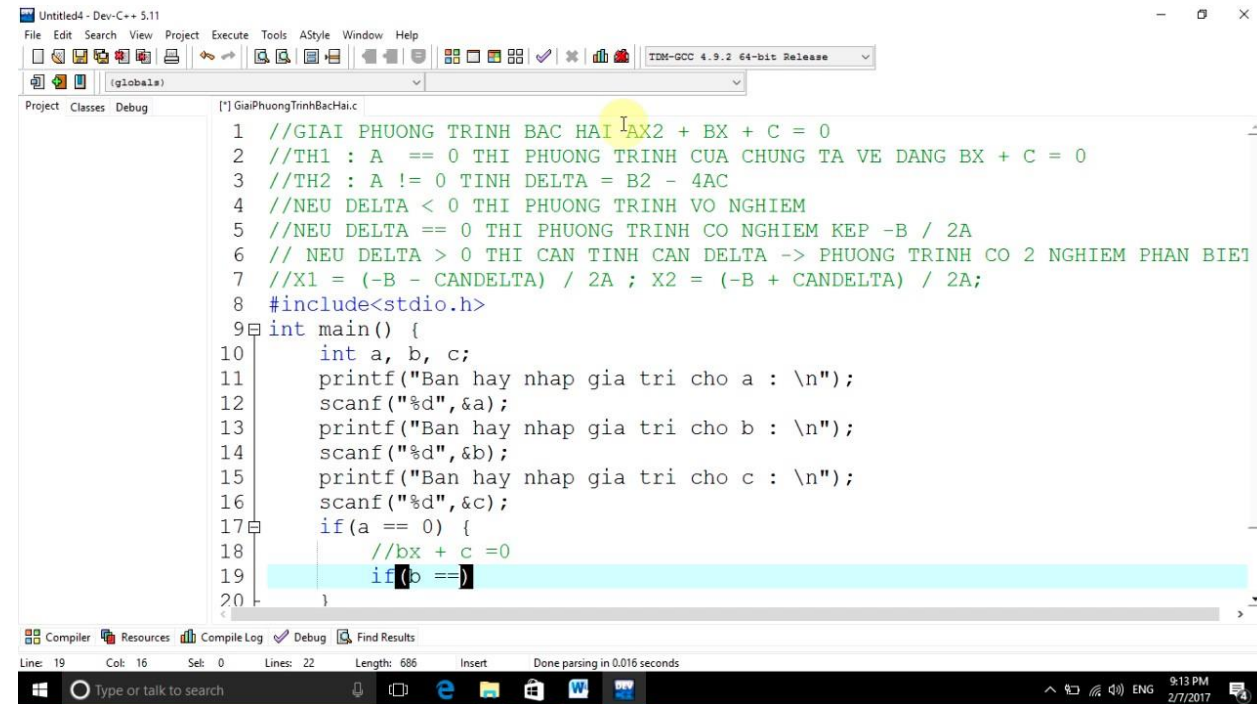


- Điểm bắt đầu / Kết thúc giải thuật
- Thao tác nhập/ xuất dữ liệu
- Thao tác xử lý
- Điều khiển rẽ nhánh
- Đường tiến trình
- Chú thích
- Ký hiệu kết nối cùng trang hay sang trang khác



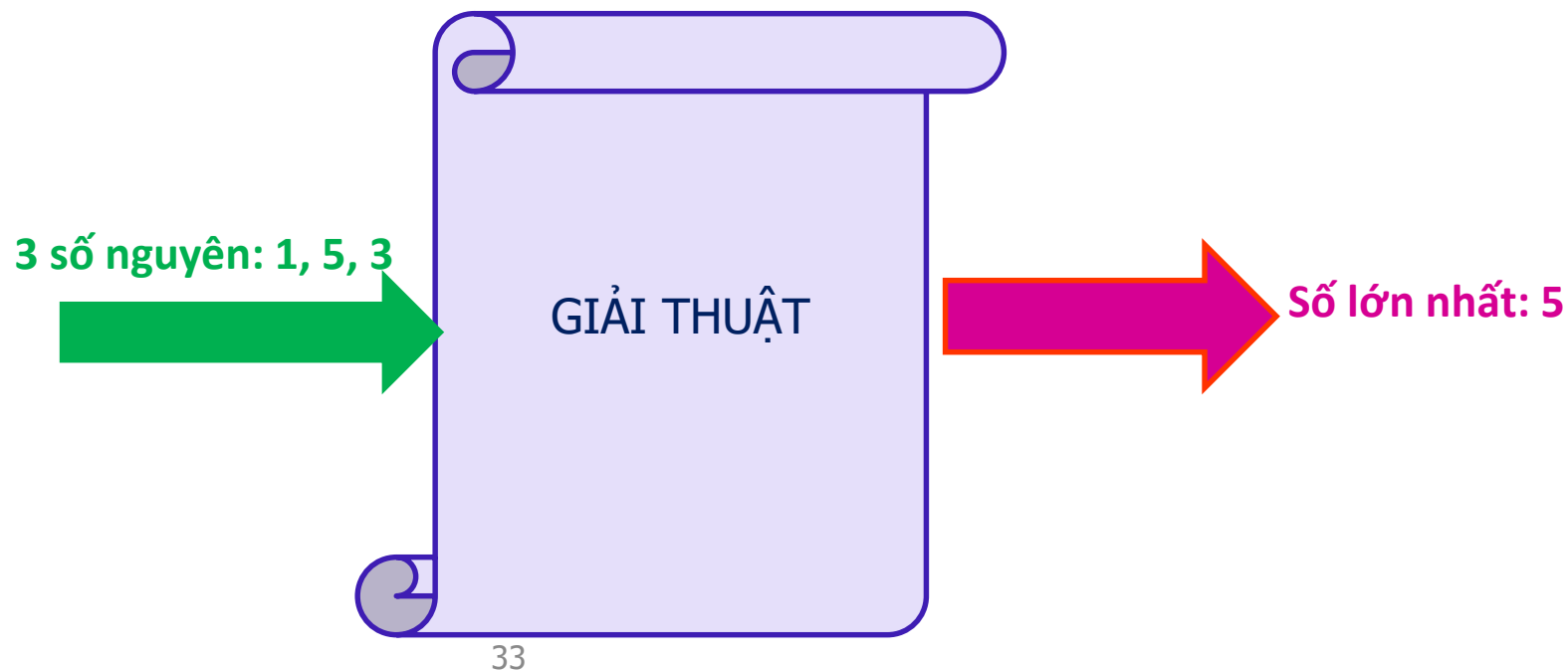
BIỂU DIỄN BẰNG NGÔN NGỮ LẬP TRÌNH

- Dùng ngôn ngữ máy tính (C, Pascal,...) để diễn tả thuật toán, CTDL thành câu lệnh.
- Để có kỹ năng lập trình đòi hỏi cần học tập và thực hành.
- Dùng phương pháp tinh chế từng bước để chuyển hoá bài toán sang mã chương trình cụ thể.



```
1 //GIAI PHUONG TRINH BAC HAI  $AX^2 + BX + C = 0$ 
2 //TH1 : A == 0 THI PHUONG TRINH CUA CHUNG TA VE DANG  $BX + C = 0$ 
3 //TH2 : A != 0 TINH DELTA =  $B^2 - 4AC$ 
4 //NEU DELTA < 0 THI PHUONG TRINH VO NGHIEM
5 //NEU DELTA == 0 THI PHUONG TRINH CO NGHIEM KEP  $-B / 2A$ 
6 // NEU DELTA > 0 THI CAN TINH CAN DELTA -> PHUONG TRINH CO 2 NGHIEM PHAN BIEI
7 //X1 =  $(-B - \text{CANDELTA}) / 2A$  ; X2 =  $(-B + \text{CANDELTA}) / 2A$ ;
8 #include<stdio.h>
9 int main() {
10     int a, b, c;
11     printf("Ban hay nhap gia tri cho a : \n");
12     scanf("%d",&a);
13     printf("Ban hay nhap gia tri cho b : \n");
14     scanf("%d",&b);
15     printf("Ban hay nhap gia tri cho c : \n");
16     scanf("%d",&c);
17     if(a == 0) {
18         //bx + c = 0
19         if(b == 0) {
20             //c = 0
```

- 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.



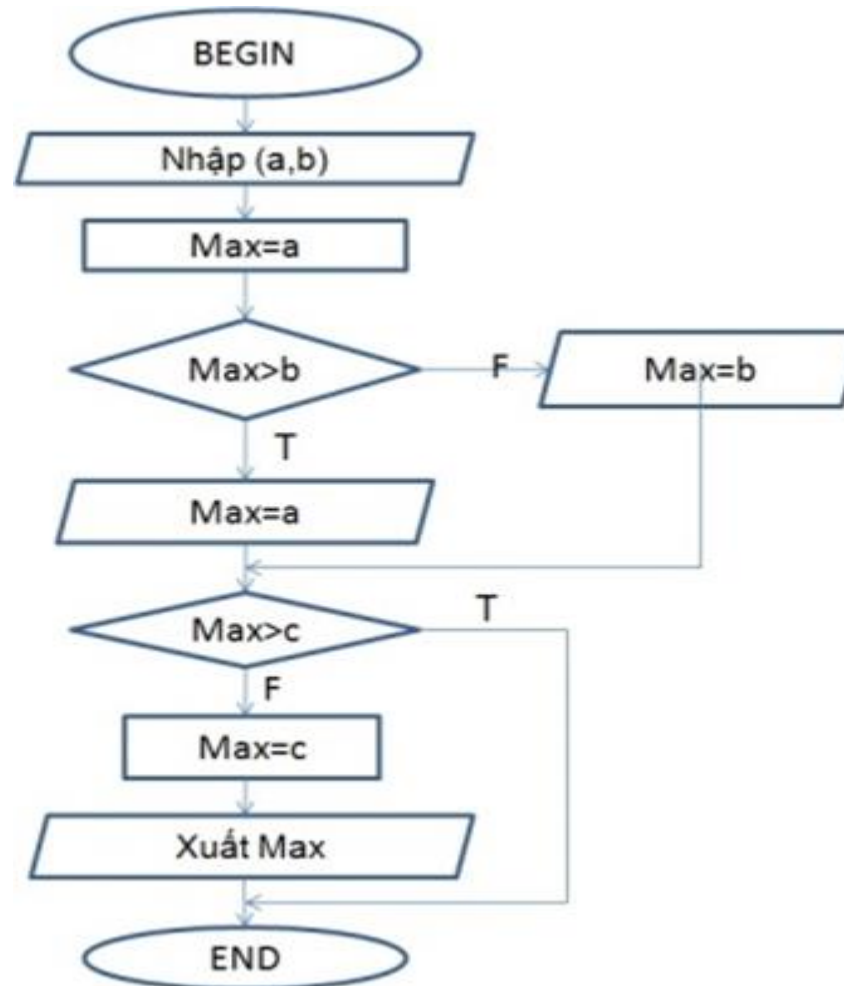
➤ Dùng ngôn ngữ tự nhiên mô tả giải thuật tìm số lớn nhất trong 3 số a, b, c :

✓ **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 đồ mô tả giải thuật tìm số lớn nhất trong 3 số a, b, c:



➤ Theo dõi quá trình thực hiện của thuật toán với giá trị cụ thể 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 ko thực hiện gán

$\text{max} = 5$

➤ Một vài nhận xét:

- ✓ Giải thuật có **tính chính xác**: Các bước của giải thuật được mô tả chính xác.
- ✓ Giải thuật có **tính duy nhất**: với đầu vào đã xác định, kết quả tại mỗi bước của giải thuật được xác định duy nhất
- ✓ Giải thuật có **tính hữu hạn**: Giải thuật kết thúc sau 3 bước và đưa ra lời giải của bài toán
- ✓ Giải thuật có **tính tổng quát**: luôn đưa ra giá trị của số lớn nhất trong 3 số bất kì

5

ĐỘ PHỨC TẠP CỦA THUẬT TOÁN



ĐỘ PHỨC TẠP GIẢI THUẬT

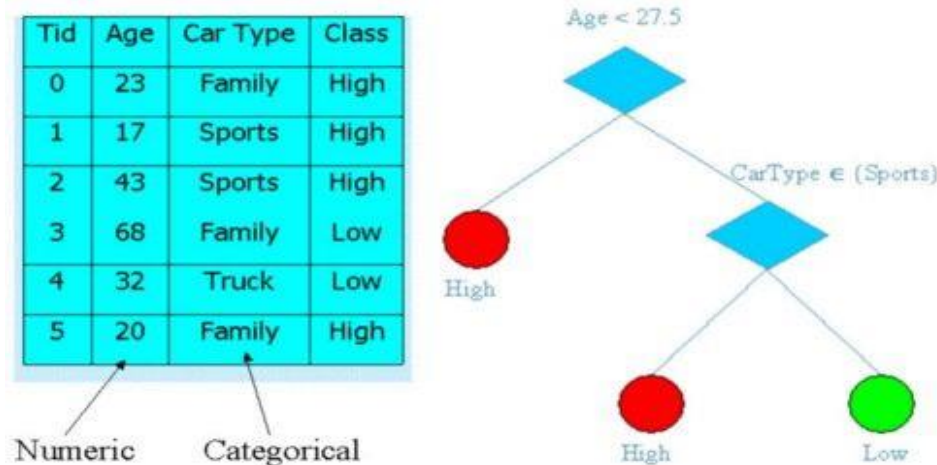
➤ Một thuật toán hiệu quả:

✓ Chi phí cần sử dụng tài nguyên thấp: Bộ nhớ, thời gian sử dụng CPU, ...

➤ Phương pháp đánh giá độ phức tạp của giải thuật:

✓ Dựa trên thời gian để thực hiện giải thuật đến khi ra được kết quả.

✓ Dựa trên số phép toán để thực hiện giải thuật



1) Age < 27.5 \Rightarrow High

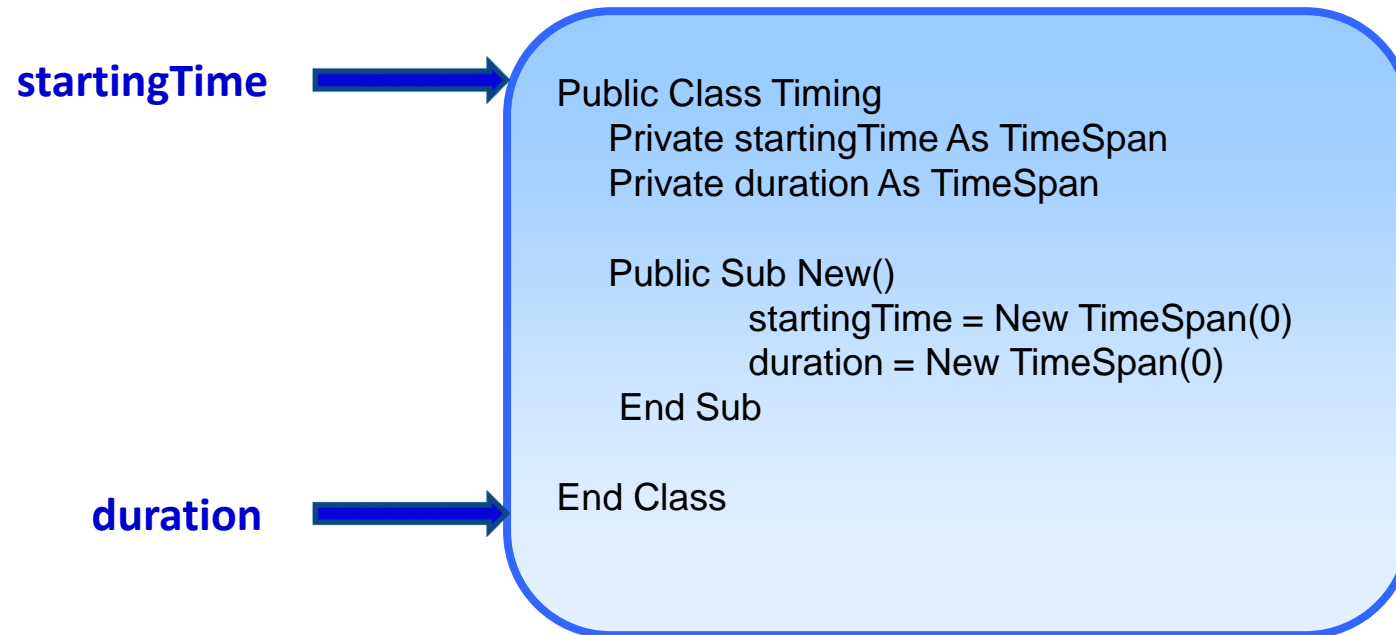
2) Age \geq 27.5 and
CarType = Sports \Rightarrow High

3) Age \geq 27.5 and
CarType \neq Sports \Rightarrow Low

TÍNH THỜI GIAN THỰC HIỆN GIẢI THUẬT

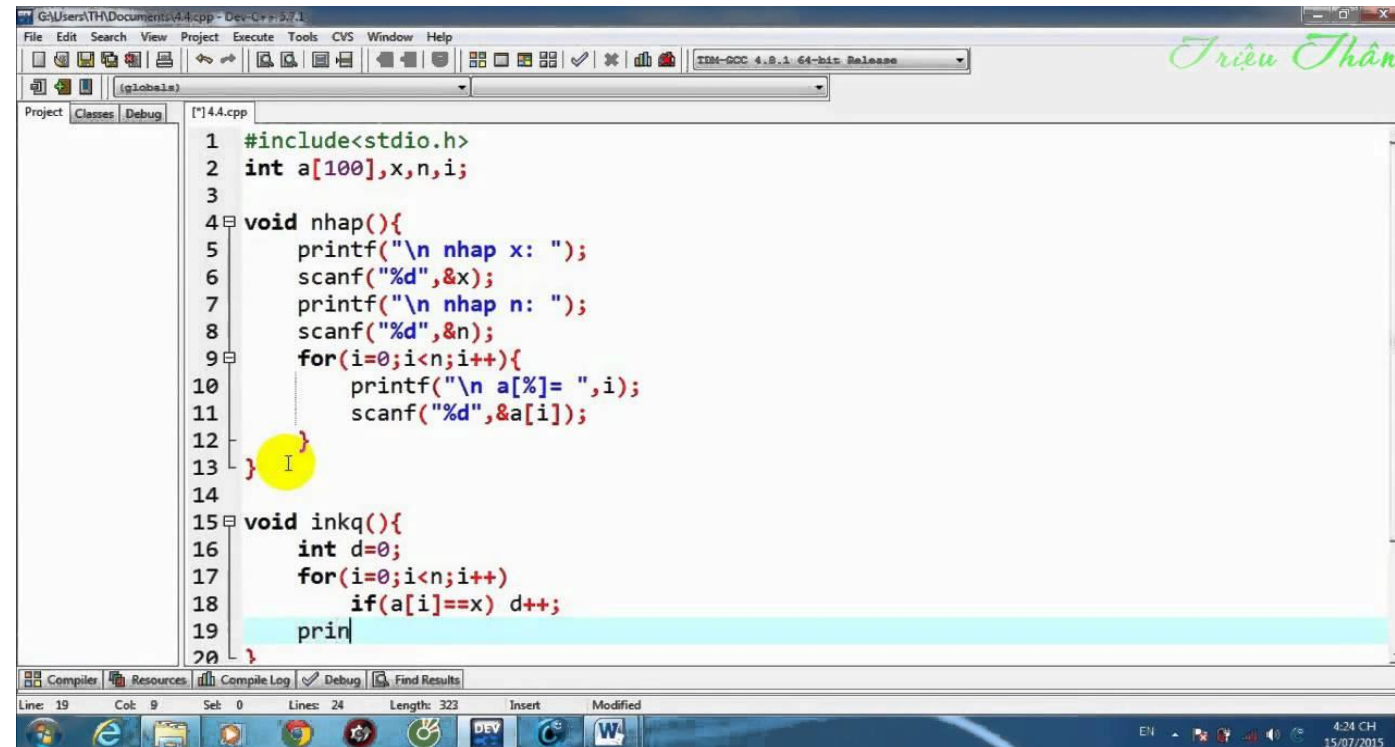
- Thời gian thực hiện giải thuật hoàn toàn có thể đo bằng cách lập trình.
- Ví dụ cài đặt lớp Timing trong VB.Net:
 - ✓ Cho phép tính thời gian thực hiện đoạn code chạy trong chương trình
 - ✓ Lớp Timing có 2 thành phần dữ liệu:
 - `startingTime`: lưu trữ thời gian bắt đầu thực hiện đoạn code
 - `duration`: thời gian kết thúc của đoạn code
 - Thời gian thực hiện = `startingTime - duration`

TÍNH THỜI GIAN THỰC HIỆN GIẢI THUẬT



TÍNH SỐ PHÉP TOÁN ĐỂ THỰC HIỆN GIẢI THUẬT

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



The screenshot shows a C++ IDE window titled "G:\Users\TH\Documents\4.4.cpp - Dev-C++ 5.11". The code is as follows:

```
1 #include<stdio.h>
2 int a[100],x,n,i;
3
4 void nhap(){
5     printf("\n nhap x: ");
6     scanf("%d",&x);
7     printf("\n nhap n: ");
8     scanf("%d",&n);
9     for(i=0;i<n;i++){
10         printf("\n a[%]= ",i);
11         scanf("%d",&a[i]);
12     }
13 }
14
15 void inkq(){
16     int d=0;
17     for(i=0;i<n;i++){
18         if(a[i]==x) d++;
19     }
20     prin
```

The IDE interface includes a menu bar (File, Edit, Search, View, Project, Execute, Tools, CVS, Window, Help), a toolbar, and a status bar at the bottom showing "Line: 19, Col: 9, Sel: 0, Lines: 24, Length: 323". A watermark "Trần Thành" is visible in the top right corner.

TÍNH SỐ PHÉP TOÁN ĐỂ THỰC HIỆN GIẢI THUẬT

■ Ví dụ:

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

$O(n) = 1000$

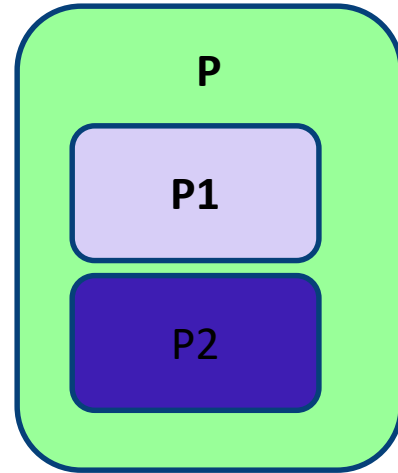
```
1 i = 1
2 loop ( i <= 1000 )
    1 application code
    2 i = i + 2
```

$O(n) = 500$

TÍNH SỐ PHÉP TOÁN ĐỂ THỰC HIỆN GIẢI THUẬT

■ Nguyên tắc cộng:

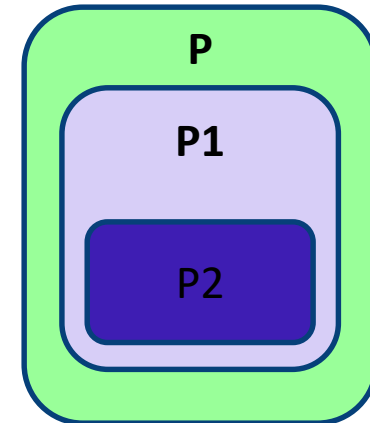
- **Nếu:** chương trình P gồm 2 đoạn chương trình P1 ($O_1(n)$) và P2 ($O_2(n)$) chạy nối tiếp nhau



- **Thì:** độ phức tạp của CT P: $O(n) = \max(O_1(n), O_2(n))$

■ Nguyên tắc nhân:

- **Nếu:** chương trình P gồm 2 đoạn chương trình P1 ($O_1(n)$) và P2 ($O_2(n)$) lồng nhau



- **Thì:** độ phức tạp của CT P: $O(n) = O_1(n).O_2(n)$

MỘT SỐ VÍ DỤ

- 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 : $O(2^N)$
- $N!$: $O(N!)$



Độ phức tạp tăng dần

MỐI LIÊN HỆ GIỮA CTDL VÀ GT

- Giải thuật = **phép xử lý**.
- Đối tượng của giải thuật chính là **dữ liệu** được tổ chức thành **các cấu trúc**.
- CTDL & GT gắn chặt với nhau.
 - ✓ **Cấu trúc dữ liệu + Giải thuật = Chương trình**
- Nếu thay đổi cấu trúc dữ liệu thì giải thuật cũng sẽ 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	SĐT
Nguyễn Văn D	098123456
Vũ Thị B	091557799
Trần Xuân A	090333999
Quách Thái C	093886868

Mối liên hệ giữa CTDL và GT

➤ Bài toán tìm số điện thoại theo họ tên:

- ✓ Nếu danh bạ không có tổ chức gì cả thì dẫn đến giải thuật là tìm tuần tự từ đầu đến cuối
- ✓ Nếu danh bạ (Họ và tên) **tổ chức theo thứ tự** a,b,c thì ta có thể tìm kiếm theo giải thuật tìm kiếm nhị phân (tìm kiểu từ điển: chia đôi và tìm trên nửa có chứa kết quả)-> **thời gian tìm kiếm nhanh hơn.**

Họ và tên	SĐT
Nguyễn Văn D	098123456
Vũ Thị B	091557799
Trần Xuân A	090333999
Quách Thái C	093886868

Danh bạ chưa sắp xếp

Họ và tên	SĐT
Trần Xuân A	090333999
Vũ Thị B	091557799
Quách Thái C	093886868
Nguyễn Văn D	098123456

Danh bạ đã sắp xếp

MỐI LIÊN HỆ GIỮA CTDL VÀ GT

■ **Nếu:** danh bạ vừa xếp thứ tự vừa có một bảng mục lục

- A - Trang 10
- B - Trang 40
- C – Trang 100

■ **Thì:** Tìm trong bảng mục lục trước. Nếu thấy thì chỉ cần tìm trong một vắn

-> thời gian thực hiện bài toán nhanh hơn.

Họ và tên	SĐT
Trần Xuân A	098123456
Vũ Thị A	091662288
Quách Thái A	090333999
Nguyễn Văn A	093886668

Họ và tên	SĐT
Trần Xuân B	012321432
Vũ Thị B	094325325
Quách Thái B	091987412
Nguyễn Văn B	096666666

HƯỚNG TIẾP CẬN CTDL> 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> cho những bài toán phức tạp
- ✓ Điểm yếu: khó hiểu, tính ứng dụng thấp do đòi hỏi sinh viên nắm vững cả lý thuyết và sâu về công cụ lập trình

➤ Hướng tiếp cận mới:

- ✓ Điểm mạnh:
 - Dễ hiểu các khái niệm CTDL>
 - Áp dụng vào ứng dụng ngay lập tức
- ✓ Điểm yếu:
 - Phải gắn với một ngôn ngữ lập trình cụ thể
 - Hạn chế trong việc có thể tự cài đặt CTDL> mới

HƯỚNG TIẾP CẬN CTDL> TRONG MÔN NÀY

- Trong thực tế, các CTDL có sẵn trong API của ngôn ngữ lập trình **đáp ứng** được đến hơn **90%** nhu cầu về lập trình với CTDL> thực tế.
- Môn này *tiếp cận CTDL> theo hướng thực tiễn*, vừa giải thích lý thuyết, hướng dẫn cài đặt, vừa hướng dẫn sử dụng các API có sẵn.
- Cụ thể, các CTDL của API ngôn ngữ lập trình **C** sẽ được sử dụng để minh họa cho môn này.

- Cấu trúc dữ liệu là cách tổ chức các dữ liệu cơ bản thành các cấu trúc như mảng, chuỗi, file...
- Giải thuật là tập hữu hạn có thứ tự các bước tác động lên dữ liệu nào đó để sau một số hữu hạn lần thực hiện sẽ cho ta kết quả.
- Việc lựa chọn cấu trúc dữ liệu và giải thuật cho bài toán là rất quan trọng.



THANK
YOU!

