

# PHẦN 1 CƠ SỞ

1. Lưu đồ thuật toán.
2. Mảng một chiều
3. Ma trận
4. Trừu tượng hóa dữ liệu
5. Con trỏ
6. Đệ quy
7. Tập tin

# PHẦN 2

## CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

# CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

**CHƯƠNG I : TỔNG QUAN VỀ CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

**CHƯƠNG II : MỘT SỐ THUẬT TOÁN TÌM KIẾM VÀ SẮP XẾP**

**CHƯƠNG III : DANH SÁCH LIÊN KẾT - NGẮN XẾP VÀ HÀNG ĐỢI**

**CHƯƠNG IV : CÂY**

**CHƯƠNG V : MỘT SỐ ỨNG DỤNG**

## **Các bước cơ bản khi tiến hành giải các bài toán tin học**

1. Xác định bài toán
2. Tìm cấu trúc dữ liệu để biểu diễn bài toán
3. Tìm thuật toán
4. Lập trình
5. Kiểm thử
6. Tối ưu

## 1. Xác định bài toán

**Dữ liệu vào → Xử lý → Kết quả ra**

Khác với các bài toán thuần túy toán học chỉ cần xác định rõ giả thiết và kết luận chứ không cần xác định yêu cầu về lời giải.

Tuy nhiên những bài toán tin học ứng dụng trong thực tế chỉ cần tìm lời giải tốt tới mức nào đó, thậm chí tồi ở mức chấp nhận được nếu lời giải tốt đòi hỏi thời gian và chi phí.

## 2. Tìm cấu trúc dữ liệu để biểu diễn bài toán

Các tiêu chuẩn khi lựa chọn CTDL

- CTDL phải biểu diễn được đầy đủ các thông tin nhập và xuất của bài toán
- CTDL phải phù hợp với thao tác của thuật toán mà ta lựa chọn
- CTDL phải cài đặt được trên máy tính với ngôn ngữ lập trình đang sử dụng

## 1. Tìm thuật toán

### *Các đặc trưng của thuật toán*

1. Tính đơn nghĩa
2. Tính dừng
3. Tính đúng
4. Tính phổ dụng
5. Tính khả thi

# CHƯƠNG I TỔNG QUAN VỀ CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

## I. KHÁI NIỆM VỀ CẤU TRÚC DỮ LIỆU

### 1. Cấu trúc dữ liệu là gì ? :

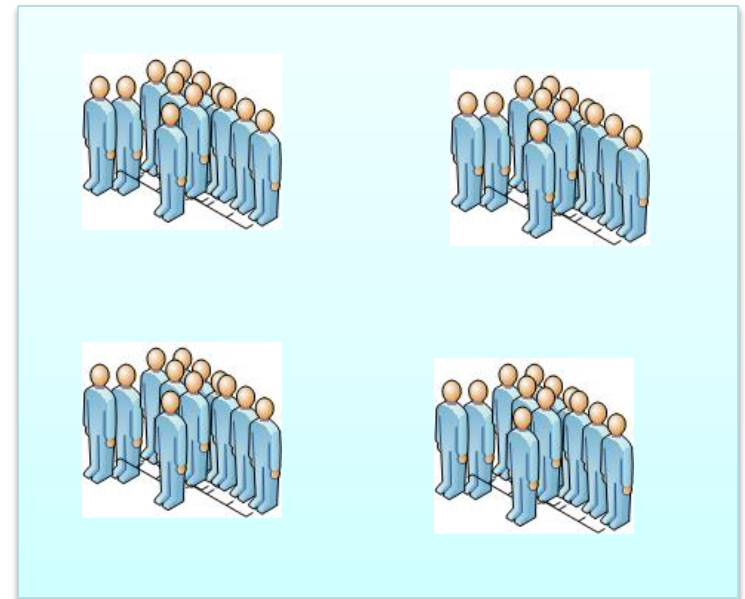
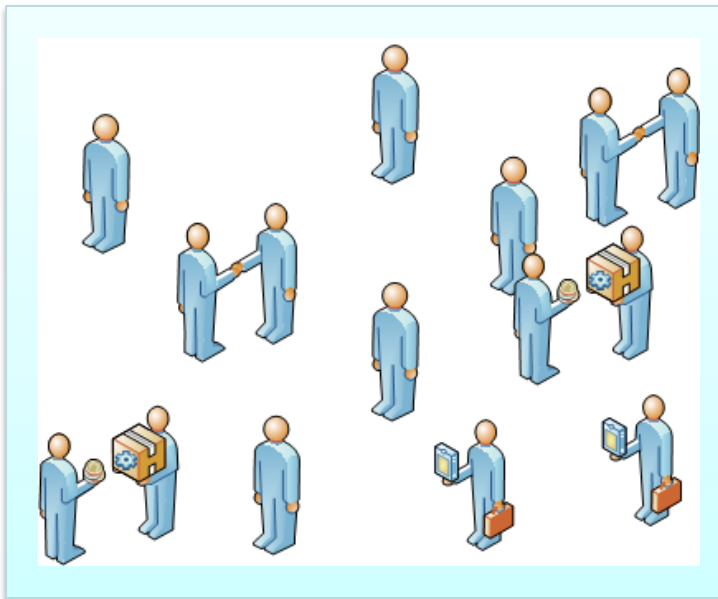




# CHƯƠNG I TỔNG QUAN VỀ CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

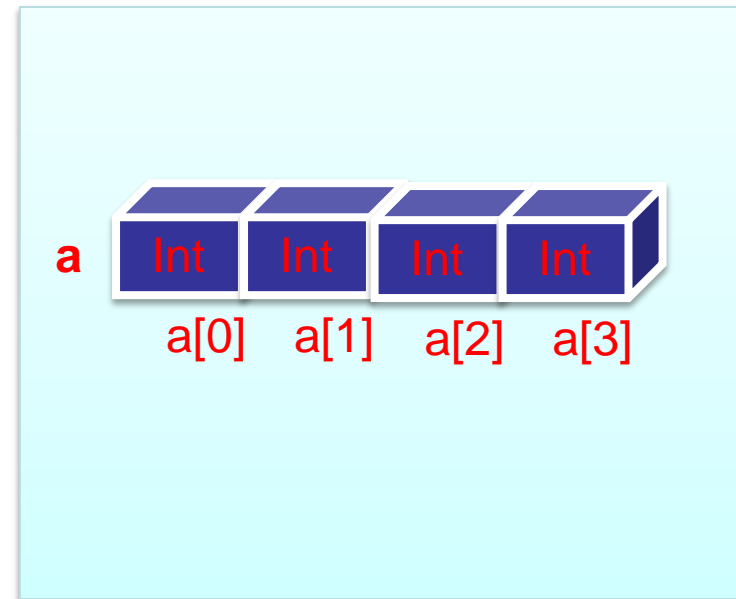
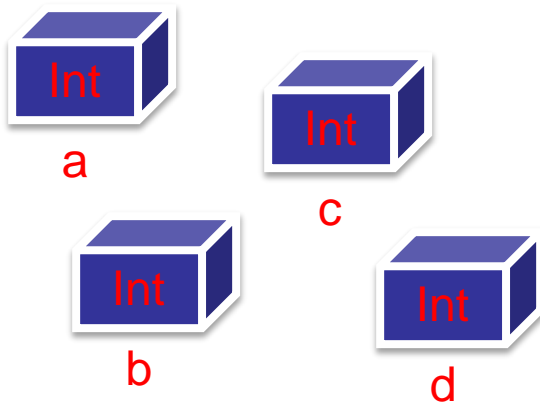
## I. KHÁI NIỆM VỀ CẤU TRÚC DỮ LIỆU

### 1. Cấu trúc dữ liệu là gì ? :



## I. KHÁI NIỆM VỀ CẤU TRÚC DỮ LIỆU

### 1. Cấu trúc dữ liệu là gì ? :



# CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

- Hạt nhân của các chương trình máy tính là sự  
***Lưu trữ và Xử lý thông tin.***
- Việc tổ chức dữ liệu như thế nào có ảnh hưởng rất lớn đến :
  - ❖ *Cách thức xử lý dữ liệu đó*
  - ❖ *Tốc độ thực thi*
  - ❖ *Sự chiếm dụng bộ nhớ của chương trình.*

## I. KHÁI NIỆM VỀ CẤU TRÚC DỮ LIỆU

### 1. Cấu trúc dữ liệu là gì ? :

- ❖ Tổ chức dữ liệu để lưu trữ.
- ❖ Mô hình dữ liệu để biểu diễn thông tin

#### □ *Dữ liệu không có cấu trúc (đơn giản):*

- Int, Char, Boolean, Float...
- Mỗi đối tượng dữ liệu là một phần tử đơn lẻ.

#### □ *Dữ liệu có cấu trúc:*

- Được cấu thành bởi các phần tử dữ liệu đơn giản.
- Mảng, Chuỗi, Danh sách, Tập tin.

# CHƯƠNG I TỔNG QUAN VỀ CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

## I. KHÁI NIỆM VỀ CẤU TRÚC DỮ LIỆU

### 2. Một số ví dụ

A

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

Array 1 chiều

# CHƯƠNG I TỔNG QUAN VỀ CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

## I. KHÁI NIỆM VỀ CẤU TRÚC DỮ LIỆU

### 2. Một số ví dụ

		Cột					
		0	1	2	3	4	5
Dòng	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]

Array 2 chiều

# CHƯƠNG I TỔNG QUAN VỀ CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

## I. KHÁI NIỆM VỀ CẤU TRÚC DỮ LIỆU

### 2. Một số ví dụ

Sinh viên (H\_ Tên, MSSV, nsinh)

```
struct    <TenCauTruc >
{
    <Kiểu> Biến 1;
    <Kiểu> Biến 2;
};
```

```
typedef struct SINH_VIÊN
{
    char    H_Tên [ 255 ];
    char    MSSV [3];
    int     năm_sinh;
};
```

Danh sách

## I. KHÁI NIỆM VỀ CẤU TRÚC DỮ LIỆU

### 3. Vai trò cấu trúc dữ liệu trong lập trình

**Cấu Trúc Dữ Liệu + Giải Thuật = Chương trình**  
**(Data Structures + Algorithms = Program)**



## II. GIẢI THUẬT

### 1. Khái niệm giải thuật :

Khái niệm giải thuật hay thuật giải mà nhiều khi còn được gọi là thuật toán dùng để chỉ phương pháp hay cách thức (method) để giải quyết vấn đề.

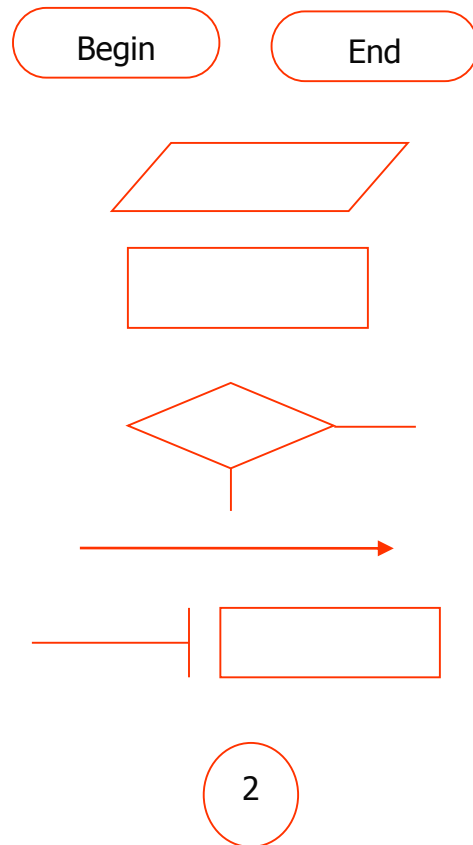
Giải thuật có thể được minh họa bằng ngôn ngữ tự nhiên (natural language), bằng lưu đồ (flow chart) hoặc bằng mã giả (pseudo code).

## II. GIẢI THUẬT

### 2. Cách viết một giải thuật ( Biểu diễn giải thuật )

- A. Ngôn ngữ tự nhiên
- B. Lưu đồ (flow chart)
- c. Mã giả (pseudo code)

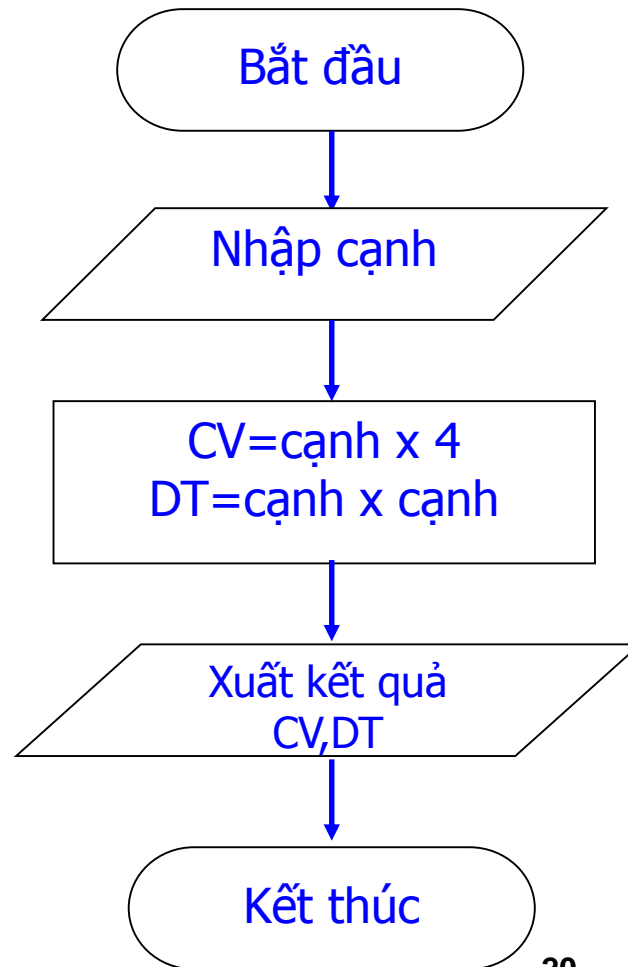
# CHƯƠNG I TỔNG QUAN VỀ CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT



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

# CHƯƠNG I TỔNG QUAN VỀ CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Ví dụ: Hãy dùng lưu đồ mô tả bài toán tính Chu vi và Diện tích hình vuông khi người sử dụng cho biết số đo cạnh của nó.



## III. ĐỘ PHỨC TẠP CỦA GIẢI THUẬT (Algorithm Efficiency)

### 1. Dạng tổng quát

$f(n)$

$n$  : kích cỡ đầu vào của dữ liệu

## III. ĐỘ PHỨC TẠP CỦA GIẢI THUẬT (Algorithm Efficiency)

### 2. Vòng lặp đơn

```
1  i = 1
2  while ( i <= 1000 )
    {
        application code
        i = i + 1
    }
```

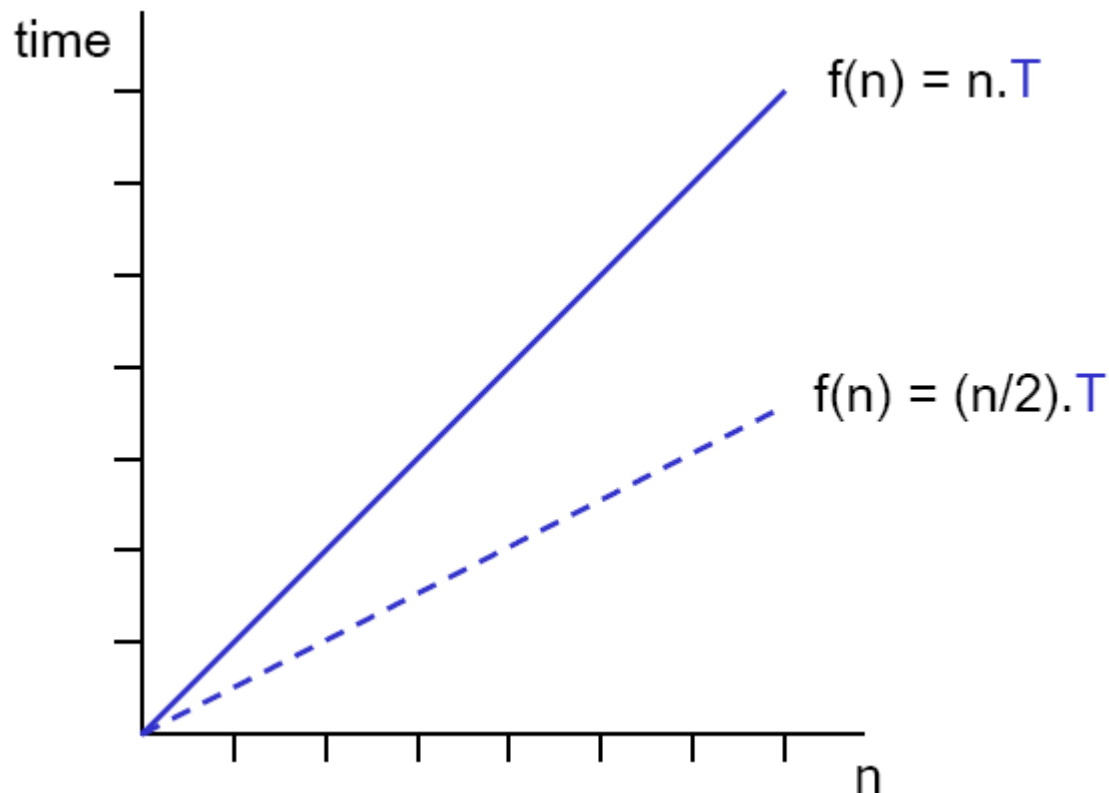
1000

```
1  i = 1
2  while ( i <= 1000 )
    {
        application code
        i = i + 2
    }
```

500

## III. ĐỘ PHỨC TẠP CỦA GIẢI THUẬT (Algorithm Efficiency)

### 2. Vòng lặp đơn



## III. ĐỘ PHỨC TẠP CỦA GIẢI THUẬT (Algorithm Efficiency)

### 2. Vòng lặp đơn

```
1  i = 1
2  while ( i <= 1000 )
    {
        application code
        i = i + 1
    }
```

1000

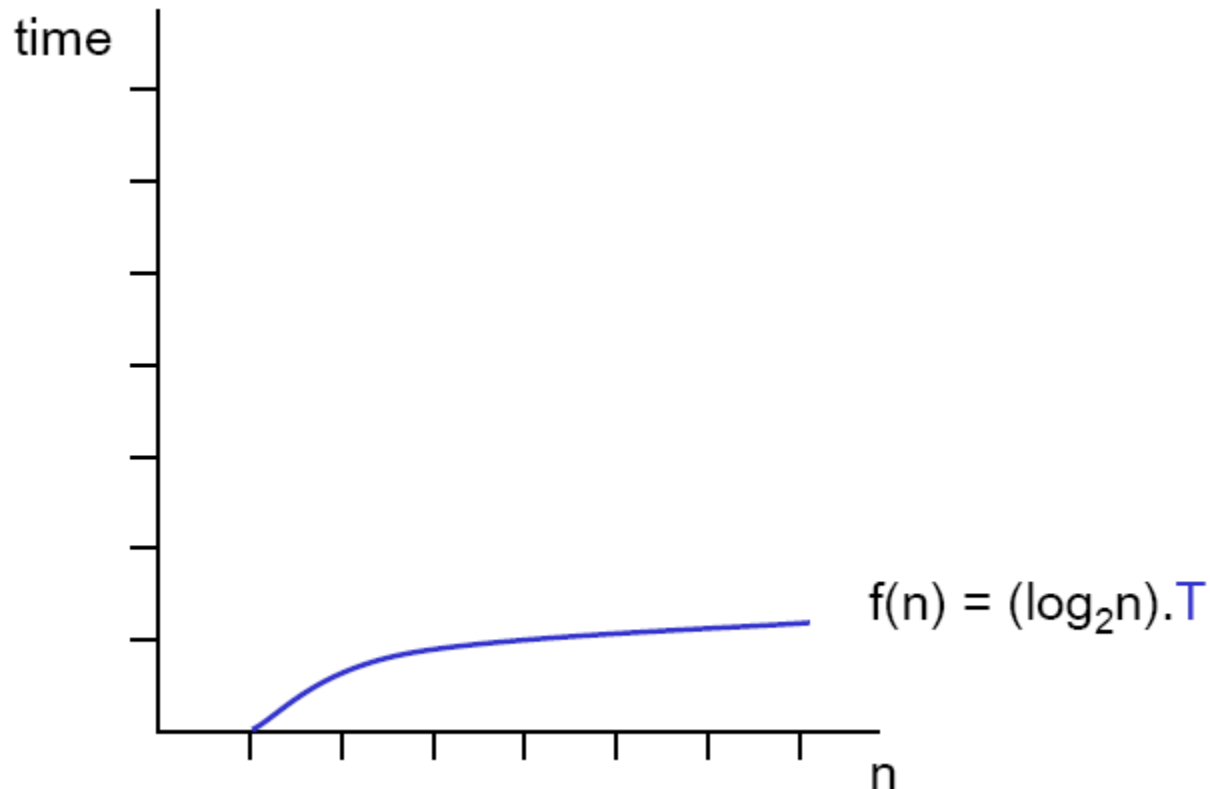
```
1  i = 1
2  while ( i <= 1000 )
    {
        application code
        i = i x 2
    }
```

$\log_2 1000$



## III. ĐỘ PHỨC TẠP CỦA GIẢI THUẬT (Algorithm Efficiency)

### 2. Vòng lặp đơn



## III. ĐỘ PHỨC TẠP CỦA GIẢI THUẬT (Algorithm Efficiency)

### 3. Vòng lặp lồng nhau

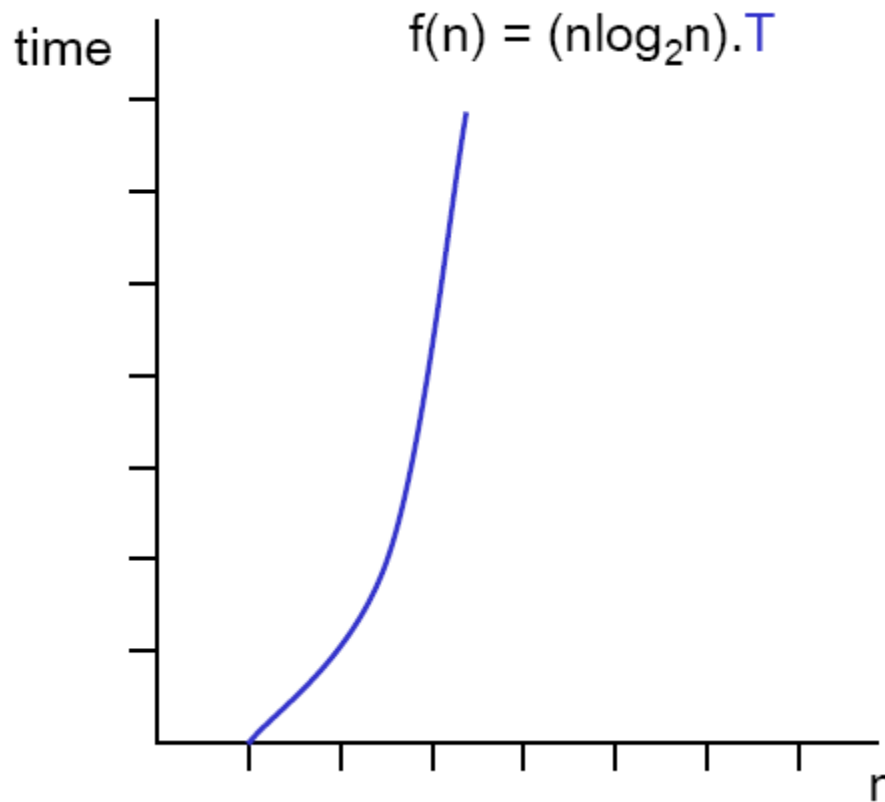
Số lần lặp = Outer loop x Inner loop

```
1  i = 1
2  While ( i <= 10 )
    {
        j = 1
        while (j <= 10)
        {
            application code
            j = j x 2
        }
    }
3  i = i + 1
```

$n \log_2 n$

## III. ĐỘ PHỨC TẠP CỦA GIẢI THUẬT (Algorithm Efficiency)

### 3. Vòng lặp lồng nhau



## III. ĐỘ PHỨC TẠP CỦA GIẢI THUẬT (Algorithm Efficiency)

### 4. Đánh giá tổng kết

$\log_2 n$     $n$     $n \log_2 n$     $n^2$     $n^3$     $\dots$     $n^k$     $\dots$     $2^n$     $n!$

## III. ĐỘ PHỨC TẠP CỦA GIẢI THUẬT (Algorithm Efficiency)

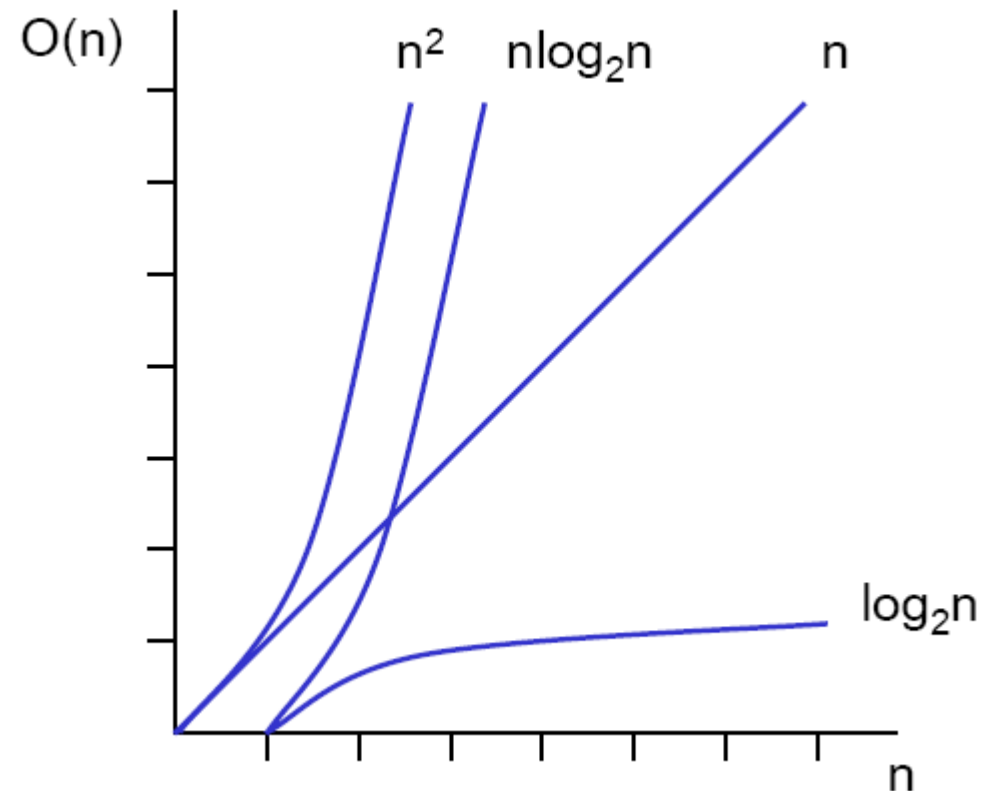
### 4. Đánh giá tổng kết

**$n = 10,000$**

Efficiency	Big-O	Iterations	Est. Time
logarithmic	$O(\log_2 n)$	14	microseconds
linear	$O(n)$	10,000	.1 seconds
linear logarithmic	$O(n \log_2 n)$	140,000	2 seconds
quadratic	$O(n^2)$	$10,000^2$	15-20 min.
polynomial	$O(n^k)$	$10,000^k$	hours
exponential	$O(2^n)$	$2^{10,000}$	intractable
factorial	$O(n!)$	$10,000!$	intractable

## III. ĐỘ PHỨC TẠP CỦA GIẢI THUẬT (Algorithm Efficiency)

### 4. Đánh giá tổng kết



# CHƯƠNG I TỔNG QUAN VỀ CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

