

# CTT009

## Thuật toán

Lê Thị Nhân  
[lt Nhan@fit.hcmus.edu.vn](mailto:lt Nhan@fit.hcmus.edu.vn)



KHOA CÔNG NGHỆ THÔNG TIN  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

# Nội dung

- ☐ Khái niệm
- ☐ Phát triển thuật toán
- ☐ Biểu diễn thuật toán
- ☐ Đánh giá thuật toán





# KHÁI NIỆM



# Giới thiệu

- Chúng ta đã học cách thức
  - Chuyển đổi các biểu diễn số từ dạng này sang dạng khác
  - Phát hiện và sửa lỗi trong dữ liệu
  - Nén và giải nén các tập dữ liệu
  - ...



# Giới thiệu

## □ CPU tuân theo chu kỳ máy tính

As long as the halt instruction has not been executed continue to execute the following steps:

- a. Fetch an instruction.
- b. Decode the instruction.
- c. Execute the instruction.

Thuật toán đơn giản

Source: *Computer Science - An Overview, 12e*

# Định nghĩa

- Thuật toán là một ***tập có thứ tự*** các bước ***rõ ràng, có thể thực thi*** mà xác định được ***quá trình dừng***
- Thuật toán/giải thuật là một tập hợp hữu hạn của các chỉ thị hay phương cách được định nghĩa cho việc hoàn tất một số việc từ một trạng thái ban đầu cho trước dẫn đến kết quả mong muốn
- Một bài toán có thể được giải quyết bởi các thuật toán khác nhau

# Ví dụ

- Thuật toán để giải phương trình bậc nhất  
 $P(x): ax + b = c$  (với  $a, b, c$  là các số thực)  
có thể thực hiện qua một số bước sau

Nếu  $a = 0$

$b = c$  thì  $P(x)$  có nghiệm bất kì

$b \neq c$  thì  $P(x)$  vô nghiệm

Nếu  $a \neq 0$

$P(x)$  có duy nhất một nghiệm  $x = (c - b)/a$



# Tầm quan trọng của thuật toán

## □ Theo Boolos và Jeffrey

- Không có con người nào có thể viết đủ nhanh, đủ dài, đủ nhỏ để liệt kê tất cả các thành phần của một tập rất lớn gần như vô hạn mà chỉ bằng cách lần lượt viết ra tên của chúng theo một số quy ước
- Tuy nhiên, con người có thể đưa ra cách thức để xác định phần tử thứ  $n$  bất kì
- Từ đó, cách thức này sẽ được thực hiện bởi các máy điện toán





# Câu hỏi về thuật toán

- ☐ Các nhà phát triển thuật toán thường tự đặt 4 câu hỏi phản biện (critical) khi họ đánh giá các thuật toán
  - ☐ Có phải thuật toán giải quyết bài toán đã được nêu ra?
  - ☐ Có phải thuật toán rõ ràng, rành mạch?
  - ☐ Thuật toán có đưa ra một kết xuất?
  - ☐ Thuật toán có kết thúc trong một khoảng thời gian hợp lý?



# PHÁT TRIỂN THUẬT TOÁN



# Quá trình phát triển thuật toán

- ☐ Xác định đầu vào
- ☐ Xác định tiến trình thực hiện
- ☐ Xác định đầu ra
- ☐ Phát triển lược đồ HIPO
- ☐ Xác định các thành phần (module) liên quan



# Quá trình phát triển thuật toán

- ☐ Xác định đầu vào
  - ☐ Thuật toán cần dữ liệu gì ?
  - ☐ Như thế nào để có dữ liệu đó ?
  - ☐ Định dạng dữ liệu như thế nào ?
- ☐ Xác định tiến trình thực hiện
- ☐ Xác định đầu ra
- ☐ Phát triển lược đồ HIPO
- ☐ Xác định các thành phần (module) liên quan



# Quá trình phát triển thuật toán

- ☐ Xác định đầu vào
- ☐ Xác định tiến trình thực hiện
  - ☐ Làm cách nào để thao tác với dữ liệu để sinh ra những kết quả có ý nghĩa?
- ☐ Xác định đầu ra
- ☐ Phát triển lược đồ HIPO
- ☐ Xác định các thành phần (module) liên quan



# Quá trình phát triển thuật toán

- ☐ Xác định đầu vào
- ☐ Xác định tiến trình thực hiện
- ☐ Xác định đầu ra
  - ☐ Dữ liệu nào cần được trả ra?
  - ☐ Định dạng dữ liệu trả ra?
- ☐ Phát triển lược đồ HIPO
- ☐ Xác định các thành phần (module) liên quan

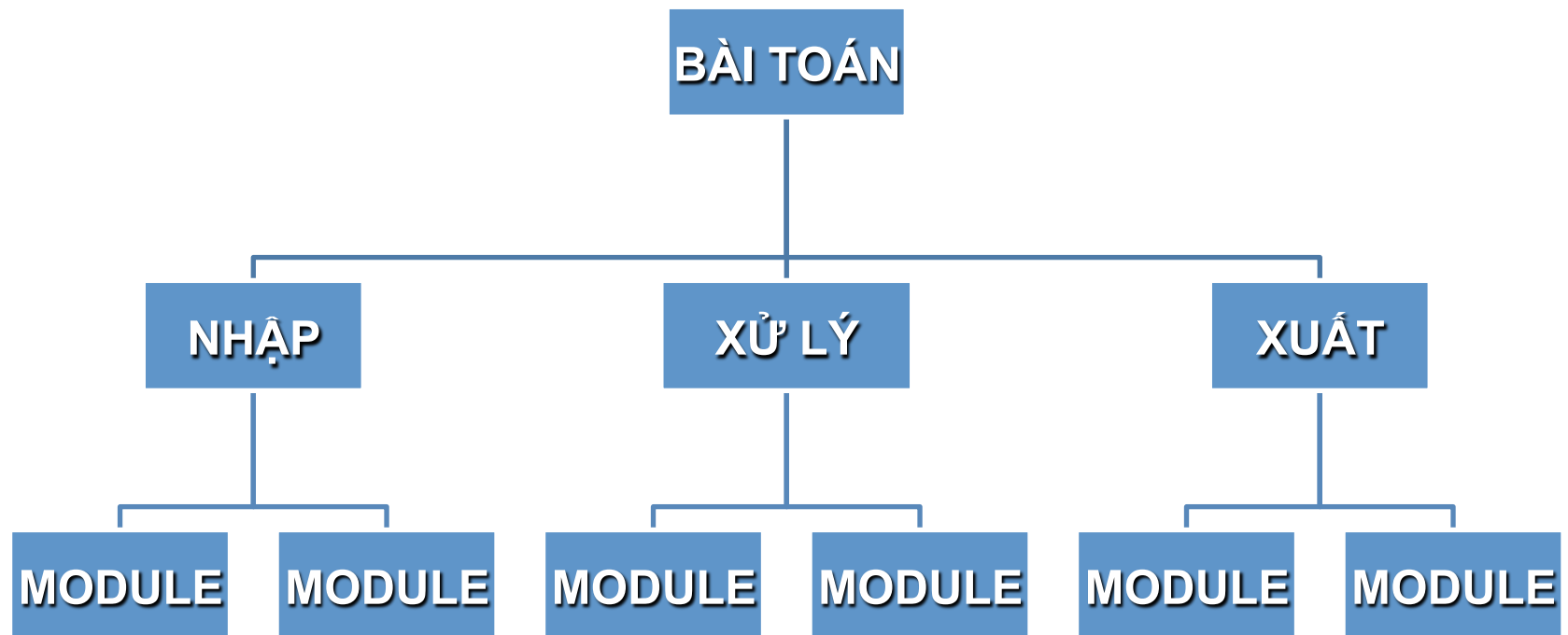


# Phát triển lược đồ HIPO

- HIPO (Hierarchy of Input-Processes-Outputs) là một kĩ thuật phục vụ cho việc lên kế hoạch và ghi tài liệu cho thuật toán
- HIPO là một biểu đồ phân tầng thể hiện cấu trúc điều khiển và một bộ nhập-xử lý-xuất để mô tả dữ liệu đến, dữ liệu xuất từ đâu và những xử lý được thực thi bởi các module trên lược đồ phân tầng này



# Phát triển lược đồ HIPO





# Xác định module liên quan

- ☐ Như thế nào để tách những bài toán lớn thành những mảnh nhỏ hơn và có thể quản lý được?
- ☐ Các module cần dữ liệu đầu vào nào?
- ☐ Những xử lý cần được thực hiện trong mỗi module?
- ☐ Dữ liệu kết xuất của từng module?



# BIỂU DIỄN THUẬT TOÁN



# Biểu diễn thuật toán

- Cần 1 ***ngôn ngữ*** để biểu diễn thuật toán
  - Được định nghĩa chính xác
  - Cung cấp đầy đủ thông tin
  
- Xem xét
  - Ngôn ngữ tự nhiên: hiểu nhầm
  - Ngôn ngữ hình ảnh: mức độ chi tiết



# Biểu diễn thuật toán

## □ Giải pháp

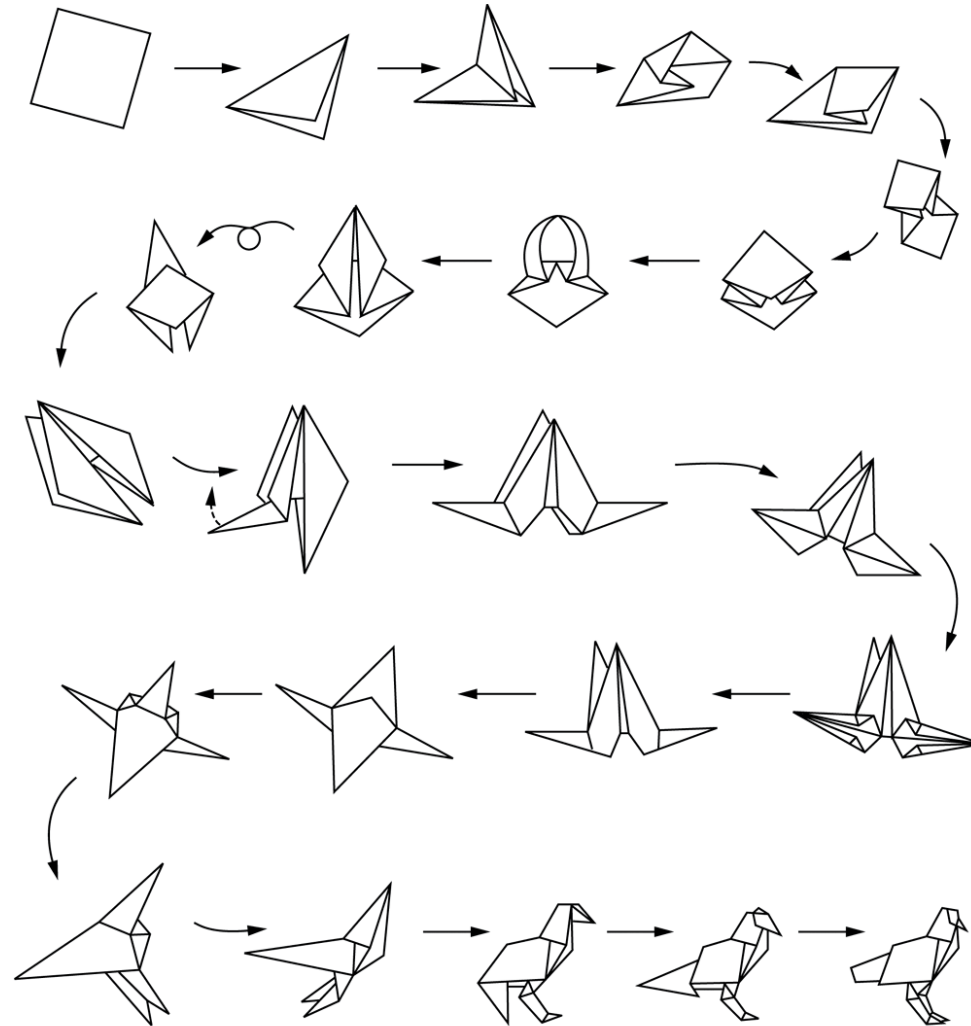
### □ Nguyên thủy (primitives)

- Cú pháp (syntax) - Ký hiệu
- Ngữ nghĩa (semantic) - Ý nghĩa

### □ Mã giả (pseudocode)

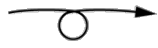
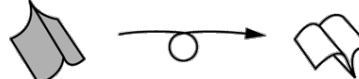
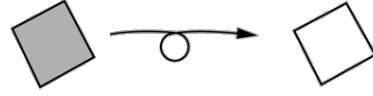












- Hệ thống ký hiệu trực giác biểu diễn các ý tưởng thuật toán

# Ví dụ



Source: Computer Science - An Overview, 12e

# Ví dụ

Syntax	Semantics
	Turn paper over as in 
Shade one side of paper	Distinguishes between different sides of paper as in 
	Represents a valley fold so that  represents 
	Represents a mountain fold so that  represents 
	Fold over so that  produces 
	Push in so that  produces 

Source: Computer Science - An Overview, 12e

# Mã giả

- Phép gán

*name = expression*

- Ví dụ

**RemainingFunds** = CheckingBalance +  
SavingsBalance

# Mã giả

□ Phép chọn có điều kiện

*if (condition):  
activity*

□ Ví dụ

*if (sales have decreased):  
lower the price by 5%*





# Mã giả

□ Phép chọn có điều kiện

**if** (*condition*):

*activity*

**else:**

*activity*

□ Ví dụ

**if** (year is leap year):

**daily total** = total / 366

**else:**

**daily total** = total / 365

# Mã giả

- Thực hiện vòng lặp

**while** (*condition*):  
*body*

- Ví dụ

**while** (tickets remain to be sold):  
sell a ticket



# Mã giả

□ Các điều kiện lồng nhau

```
if (not raining):
```

```
    if (temperature == hot):
```

```
        go swimming
```

```
    else:
```

```
        play golf
```

```
else:
```

```
    watch television
```



# Mã giả

- Định nghĩa hàm

```
def name():
```

- Ví dụ

```
def ProcessLoan():
```

- Gọi hàm

```
if (...):
```

```
    ProcessLoan()
```

```
else:
```

```
    RejectApplication()
```

# Mã giả

## □ Thủ tục

```
def Greetings():  
    Count = 3  
    while (Count > 0):  
        print('Hello')  
        Count = Count - 1
```

# Mã giả

□ Tham số

```
def Sort(List):
```

□ Ví dụ

```
Sort(the membership list)
```

```
Sort(the wedding guest list)
```



# Biểu diễn thuật toán

## ☐ Giải pháp

- ☐ Lược đồ
- ☐ Bảng điều khiển
- ☐ Ngôn ngữ lập trình

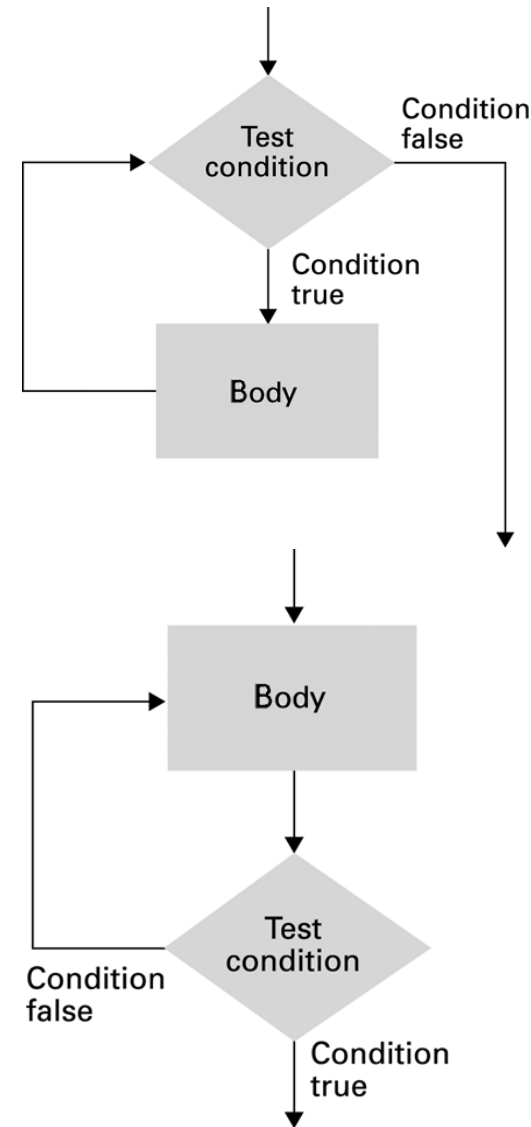


# Lược đồ

□ Vòng lặp

**while** (*condition*):  
*body*

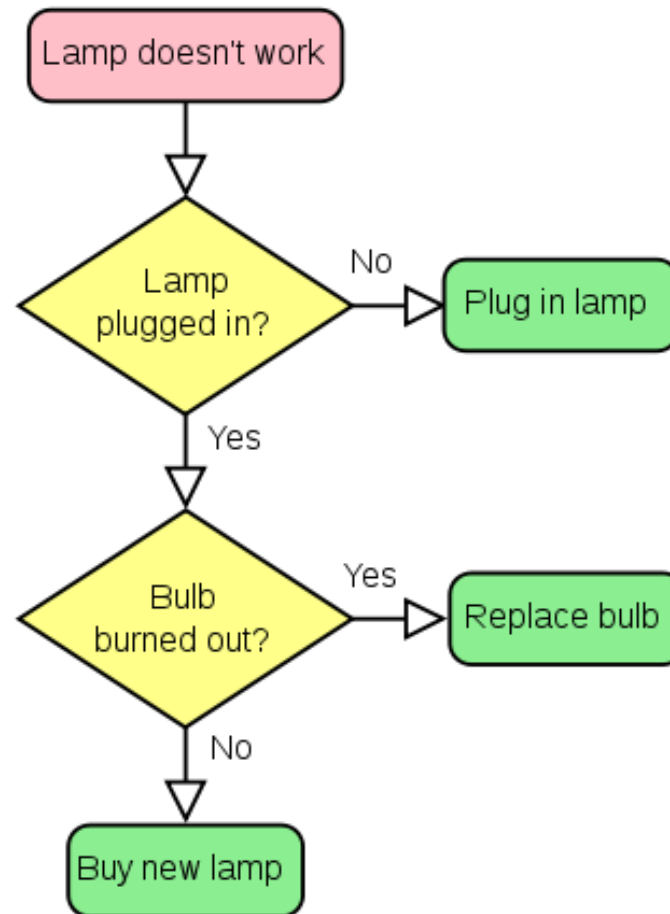
**repeat**:  
*body*  
**until**(*condition*)



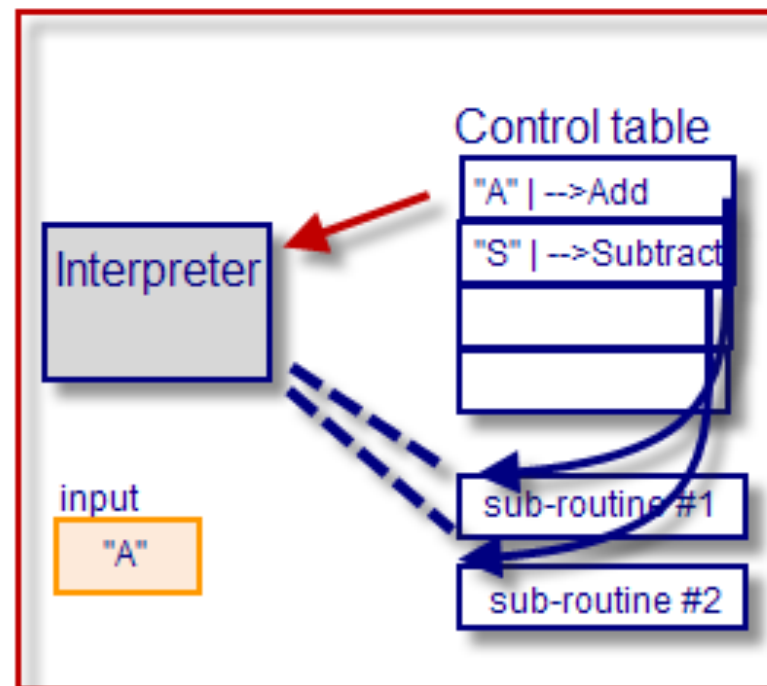
Source: Computer Science -  
An Overview, 12e



# Lược đồ



# Bảng điều khiển



# ĐÁNH GIÁ THUẬT TOÁN



# Đánh giá thuật toán

- Thuật toán được đánh giá dựa trên khối lượng tài nguyên (thời gian và bộ nhớ) cần để thực thi nó
  - Độ phức tạp về mặt không gian
  - Độ phức tạp về mặt thời gian



# Tại sao cần đánh giá

- Việc sử dụng vô ý một thuật toán không hiệu quả có thể ảnh hưởng đến hiệu năng hệ thống
- Trong các ứng dụng thời gian thực, một thuật toán chạy quá lâu có thể làm cho kết quả của nó đã lỗi thời hoặc vô dụng
- Một thuật toán không hiệu quả cũng có thể tiêu tốn một khối lượng tính toán hay vùng nhớ một cách không kinh tế để chạy



# Phân loại thuật toán

- Tiêu chí phân loại
  - Theo cách thực thi
    - Tuần tự, song song, ...
  - Theo phương pháp thiết kế
    - Vết cặn, chia để trị, ...
  - Theo lĩnh vực nghiên cứu
    - Tìm kiếm, sắp xếp, ...
  - Theo độ phức tạp
    - Khối lượng thời gian cần để hoàn thành so với kích thước dữ liệu nhập



# TÓM TẮT



# Bài giảng hôm nay

☐ Thuật toán

☐ Biểu diễn

☐ Mã giả

☐ Lược đồ





# Bài giảng tuần sau

- ☐ Nghề nghiệp
- ☐ Quy định về đạo đức nghề nghiệp



