

## Nhập môn Công nghệ Thông tin

### Chapter 7 - Problem Solving and Algorithms (Giải quyết Vấn đề và Giải thuật)

Khoa Công nghệ Thông tin  
Đại học Ngoại ngữ - Tin học TP HCM  
2023

1

## Giải quyết vấn đề

### Problem solving (Giải quyết vấn đề)

The act of finding a solution to a perplexing, distressing, vexing, or unsettled question  
(Hành động để tìm ra giải pháp cho một câu hỏi rắc rối, chưa được giải)

*Bạn định nghĩa thế nào là giải quyết vấn đề?*

2

2

## Giải quyết vấn đề

*Bạn giải quyết vấn đề như thế nào?*

Understand the problem (**Hiểu vấn đề**)

Devise a plan (**Lập kế hoạch**)

Carry out the plan (**Thực hiện kế hoạch**)

Look back (**Xem xét lại**)

3

3

## Chiến lược

### Tự đặt ra các câu hỏi!

- Tôi biết gì về vấn đề này?
- Thông tin mà tôi phải xử lý để tìm ra giải pháp?
- Giải pháp như thế nào?
- Tồn tại những trường hợp đặc biệt nào?
- Làm thế nào để tôi nhận ra rằng tôi đã tìm ra giải pháp?

4

4

## Chiến lược

### Ask questions! Never reinvent the wheel!

Các **vấn đề tương tự** xuất hiện lặp đi lặp lại dưới nhiều hình thức khác nhau

Một lập trình viên giỏi nhận ra một nhiệm vụ hoặc nhiệm vụ phụ đã được giải quyết trước đó và đưa ra giải pháp

*Bạn có thể nghĩ về hai vấn đề giống nhau không?*

5

5

## Chiến lược

### Divide and Conquer! (Chia để trị)

Chia một vấn đề lớn thành các đơn vị nhỏ hơn và giải quyết từng vấn đề nhỏ hơn

- Áp dụng khái niệm trừu tượng
- Phương pháp chia để trị có thể được áp dụng lặp đi lặp lại cho đến khi quản lý được từng nhiệm vụ con

6

6

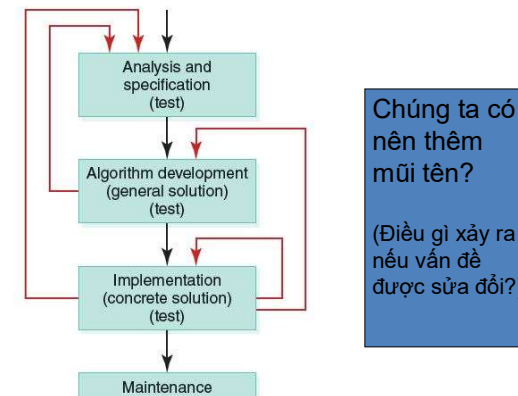
## Giải quyết vấn đề máy tính

- **Giai đoạn Phân tích và Đặc tả**  
Phân Tích (Analyze)  
Đặc tả (Specification)
- **Giai đoạn Phát triển giải thuật**  
Phát triển thuật toán (Develop algorithm)  
Thử nghiệm thuật toán (Test algorithm)
- **Giai đoạn Hiện thực**  
Code thuật toán (Code algorithm)  
Kiểm tra thuật toán (Test algorithm)
- **Giai đoạn Bảo trì**  
Sử dụng (Use)  
Bảo trì (Maintain)

7

7

## Tương tác giai đoạn



Tương tác của 4 giai đoạn trong giải quyết vấn đề CNTT

8

8

## Algorithms

### Algorithm (Giải thuật)

Một tập hợp các hướng dẫn rõ ràng để giải quyết một vấn đề hoặc vấn đề con trong một khoảng thời gian hữu hạn bằng cách sử dụng một lượng dữ liệu hữu hạn

### Abstract Step (Bước trừu tượng)

Một bước thuật toán chứa các chi tiết không xác định

### Concrete Step (Bước cụ thể)

Một bước thuật toán trong đó tất cả các chi tiết được định rõ

9

9

## Phát triển giải thuật

Hai phương pháp luận được sử dụng để phát triển các giải pháp máy tính cho một vấn đề:

- **Top-down design (Thiết kế từ trên xuống)** Tập trung vào các công việc cần thực hiện
- **Object-oriented design (Thiết kế hướng đối tượng)** tập trung vào dữ liệu liên quan đến giải pháp (Chúng ta sẽ thảo luận về thiết kế này trong Chương 9)

10

10

## Tóm tắt phương pháp luận

### Analyze the Problem (Phân tích vấn đề)

Hiểu vấn đề!!

Xây dựng kế hoạch tấn công

### List the Main Tasks (becomes Main Module) (liệt kê công việc chính)

Phân tích trình bày lại vấn đề dưới dạng danh sách các nhiệm vụ(modules)

Đặt tên cho mỗi nhiệm vụ

### Write the Remaining Modules (Viết các module còn lại)

Trình bày từng mô-đun trừu tượng dưới dạng danh sách các nhiệm vụ

Đặt tên cho mỗi nhiệm vụ

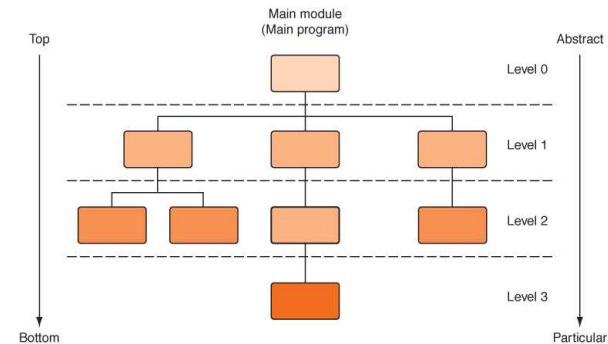
### Re-sequence and Revise as Necessary (Sắp xếp lại và xem xét sự cần thiết)

Quy trình kết thúc khi tất cả các bước (mô-đun) đã cụ thể

11

11

## Thiết kế từ trên xuống(Top-Down)



Quy trình tiếp tục ở nhiều cấp độ cần thiết để làm cho mọi bước trở nên cụ thể  
Tên của vấn đề (phụ) ở một cấp độ trở thành một mô-đun ở cấp độ thấp hơn tiếp theo

12

12

## Cấu trúc điều khiển

### Control structure (cấu trúc điều khiển)

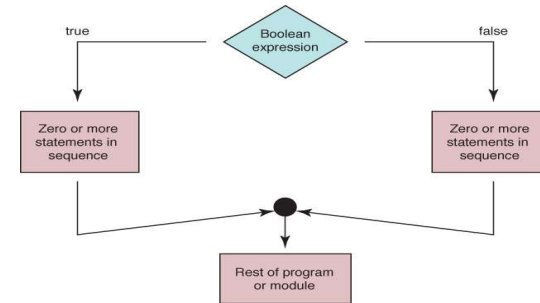
Lệnh xác định thứ tự thực hiện các lệnh khác trong chương trình

*Bạn có thể kể tên những cái mà chúng ta đã xác định trong chức năng của mã giả không?*

13

13

## Cấu trúc chọn lựa



Luồng kiểm soát câu lệnh if

14

14

## Thuật toán chọn lựa

### Determine Dress

IF (temperature > 90)

Write "Texas weather: wear shorts"

ELSE IF (temperature > 70)

Write "Ideal weather: short sleeves are fine"

ELSE IF (temperature > 50)

Write "A little chilly: wear a light jacket"

ELSE IF (temperature > 32)

Write "Philadelphia weather: wear a heavy coat"

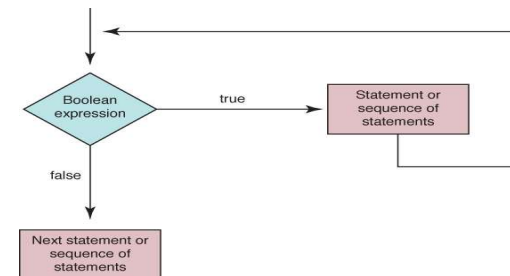
ELSE

Write "Stay inside"

15

15

## Vòng lặp



Luồng kiểm soát vòng lặp while

16

16

## Mảng

[0]	1066	← numbers[0]
[1]	1492	
[2]	1668	
[3]	1945	
[4]	1972	← numbers[4]
[5]	1510	
[6]	999	
[7]	1001	
[8]	21	
[9]	2001	

FIGURE 7.5 An array of ten numbers

17

17

## Mảng

Khi dữ liệu được đọc vào một mảng, bộ đếm sẽ được cập nhật để chúng ta luôn biết có bao nhiêu mục dữ liệu được lưu trữ

Nếu mảng được gọi là danh sách, chúng ta đang làm việc với

*list[0] to list[length-1] or  
list[0]..list[length-1]*

18

18

## Mảng không theo thứ tự

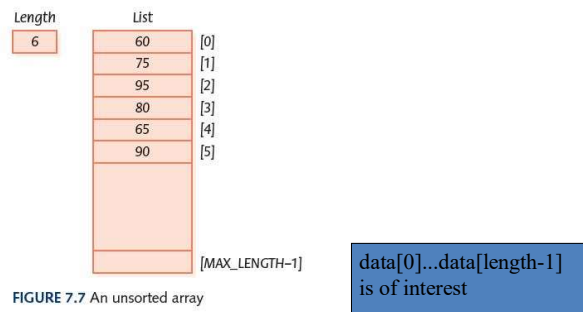


FIGURE 7.7 An unsorted array

19

19

## Thuật toán tìm kiếm tuần tự (Sequential Search)

Set Position to 0

Set found to FALSE

WHILE (position < length AND NOT found )

IF (numbers [position] equals searchitem)

Set Found to TRUE

ELSE

Set position to position + 1

20

20



## Mảng được sắp xếp

Set found to TRUE if searchItem is there

Set index to 0

Set found to FALSE

WHILE (index < length AND NOT found)

    IF (data[index] equals searchItem)

        Set found to TRUE

    ELSE IF (data[index] > searchItem)

        Set index to length

    ELSE

        Set index to index + 1

25

25

## Tìm kiếm nhị phân (Binary Search)

### Sequential search (Tìm kiếm tuần tự)

Tìm kiếm bắt đầu ở đầu danh sách và tiếp tục cho đến khi tìm thấy đối tượng đó hoặc toàn bộ danh sách đã được tìm kiếm

### Binary search (list must be sorted)

Tìm kiếm bắt đầu ở giữa và tìm thấy đối tượng hoặc loại bỏ một nửa số mục chưa được kiểm tra; quá trình được lặp lại ở một nửa nơi đối tượng có thể có

26

26

## Tìm kiếm nhị phân

Set first to 0

Set last to length-1

Set found to FALSE

WHILE (first <= last AND NOT found)

    Set middle to (first + last) / 2

    IF (item equals data[middle])

        Set found to TRUE

    ELSE

        IF (item < data[middle])

            Set last to middle - 1

    ELSE

        Set first to middle + 1

RETURN found

27

27

## Tìm kiếm nhị phân

Length	Items
11	ant [0]
	cat [1]
	chicken [2]
	cow [3]
	deer [4]
	dog [5]
	fish [6]
	goat [7]
	horse [8]
	rat [9]
	snake [10]
	.
	.

FIGURE 7.9 Binary search example

Searching for cat			
First	Last	Middle	Comparison
0	10	5	cat < dog
0	4	2	cat < chicken
0	1	0	cat > ant
1	1	1	cat = cat
Return: true			

Searching for fish			
First	Last	Middle	Comparison
0	10	5	fish > dog
6	10	8	fish < horse
6	7	6	fish = fish
Return: true			

Searching for zebra			
First	Last	Middle	Comparison
0	10	5	zebra > dog
6	10	8	zebra > horse
9	10	9	zebra > rat
10	10	10	zebra > snake
11	10		first > last
Return: false			

FIGURE 7.10 Trace of the binary search

28

28

## Tìm kiếm nhị phân

TABLE		
7.1 Average Number of Comparisons		
Length	Sequential Search	Binary Search
10	5.5	2.9
100	50.5	5.8
1000	500.5	9.0
10000	5000.5	12.0

Is a binary search  
always better?

29

29

## Sắp xếp

### Sắp xếp

Sắp xếp các đối tượng trong bộ sao cho có thứ tự trên một (hoặc nhiều) trường

### Khóa sắp xếp

Trường (hoặc các trường) dựa trên thứ tự

### Các thuật toán sắp xếp

Thuật toán sắp xếp các mục trong bộ sưu tập dựa trên khóa sắp xếp

Why is sorting important?

30

30

## Sắp xếp lựa chọn Selection Sort

### Ý tưởng chính của thuật toán:

Chọn phần tử nhỏ nhất (hoặc lớn nhất, tùy thuộc vào kiểu sắp xếp) trong danh sách và đổi chỗ nó với phần tử ở vị trí đầu tiên. Sau đó, chọn phần tử nhỏ nhất trong danh sách còn lại và đổi chỗ nó với phần tử ở vị trí thứ hai, và tiếp tục thực hiện như vậy cho đến khi toàn bộ danh sách được sắp xếp.

31

31

## Sắp xếp lựa chọn Selection Sort

Cho một danh sách các tên, hãy xếp chúng theo thứ tự bảng chữ cái

- Tìm tên đứng đầu tiên trong bảng chữ cái và viết tên đó vào tờ giấy thứ hai
- Gạch bỏ tên khỏi danh sách ban đầu
- Tiếp tục chu trình này cho đến khi tất cả các tên trong danh sách ban đầu được gạch bỏ và ghi vào danh sách thứ hai, lúc này danh sách thứ hai chứa các mục giống nhau nhưng được sắp xếp theo thứ tự

32

32



## Sắp xếp lựa chọn Selection Sort

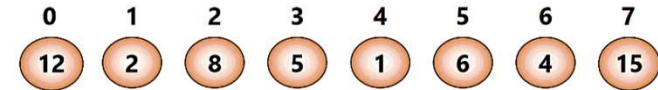
Một chút điều chỉnh đối với cách tiếp cận thủ công này sẽ không cần phải nhân đôi không gian

- Khi bạn gạch tên khỏi danh sách ban đầu, một không gian trống sẽ mở ra.
- Thay vì viết giá trị được tìm thấy trên danh sách thứ hai, hãy trao đổi nó với giá trị hiện tại ở vị trí mà mục bị gạch chéo tiếp theo

33

33

## Sắp xếp lựa chọn Selection Sort



34

34

## Selection Sort

Names	Names	Names	Names	Names
[0] Sue	[0] Ann	[0] Ann	[0] Ann	[0] Ann
[1] Cora	[1] Cora	[1] Beth	[1] Beth	[1] Beth
[2] Beth	[2] Beth	[2] Cora	[2] Cora	[2] Cora
[3] Ann	[3] Sue	[3] Sue	[3] Sue	[3] June
[4] June	[4] June	[4] June	[4] June	[4] Sue
(a)	(b)	(c)	(d)	(e)

FIGURE 7.11 Examples of selection sort (sorted elements are shaded)

35

35

## Bubble Sort Sắp xếp nổi bọt

### Ý tưởng chính của thuật toán :

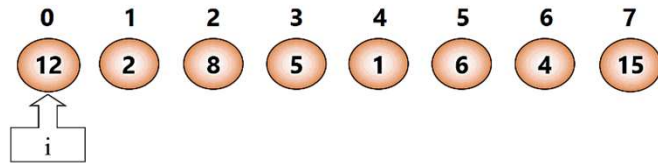
So sánh lần lượt các cặp phần tử liền kề trong danh sách và đổi chỗ chúng nếu chúng không đúng thứ tự mong muốn. Quá trình này lặp lại cho đến khi không còn cặp phần tử nào cần phải đổi chỗ, tức là danh sách đã được sắp xếp

36

36

## Bubble Sort

### Sắp xếp nổi bọt



Nếu  $a[j] < a[j-1]$  thì đổi chỗ  $a[j]$ ,  $a[j-1]$

37

37

## Bubble Sort

Names	Names	Names	Names	Names
[0] Phil	[0] Phil	[0] Phil	[0] Phil	[0] Al
[1] Al	[1] Al	[1] Al	[1] Al	[1] Phil
[2] John	[2] John	[2] Bob	[2] Bob	[2] Bob
[3] Jim	[3] Bob	[3] John	[3] John	[3] John
[4] Bob	[4] Jim	[4] Jim	[4] Jim	[4] Jim

(a) First iteration (sorted elements are shaded)

Names	Names	Names	Names
[0] Al	[0] Al	[0] Al	[0] Al
[1] Phil	[1] Bob	[1] Bob	[1] Bob
[2] Bob	[2] Phil	[2] Jim	[2] Jim
[3] John	[3] Jim	[3] Phil	[3] John
[4] Jim	[4] John	[4] John	[4] Phil

(b) Remaining iterations (sorted elements are shaded)

FIGURE 7.12 Examples of a bubble sort

38

38

## Insertion Sort

### Sắp xếp chèn

#### Ý tưởng chính của thuật toán :

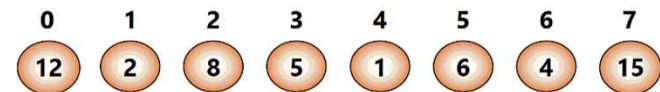
Chia danh sách thành hai phần: một phần đã được sắp xếp và một phần chưa được sắp xếp. Thuật toán này lặp qua từng phần tử trong phần chưa được sắp xếp và chèn nó vào đúng vị trí trong phần đã được sắp xếp.

39

39

## Insertion Sort

### Sắp xếp chèn



40

40

## Insertion Sort

The item being added to the sorted portion can be bubbled up as in the **bubble sort**

Names		Names		Names		Names		Names	
[0]	Phil	[0]	John	[0]	Al	[0]	Al	[0]	Al
[1]	John	[1]	Phil	[1]	John	[1]	Jim	[1]	Bob
[2]	Al	[2]	Al	[2]	Phil	[2]	John	[2]	Jim
[3]	Jim	[3]	Jim	[3]	Jim	[3]	Phil	[3]	John
[4]	Bob	[4]	Bob	[4]	Bob	[4]	Bob	[4]	Phil

FIGURE 7.13 Insertion sort