

# TIN HỌC ĐẠI CƯƠNG

## Phần 2: GIẢI QUYẾT BÀI TOÁN

Nguyễn Thị Thanh Nga  
ngantt@soict.hust.edu.vn

1

### Phần 2: Giải quyết bài toán

#### Nội dung chính

##### 1. Chương 1: Giải quyết bài toán

- Khái niệm về bài toán
- Quá trình giải quyết bài toán bằng máy tính
- Phương pháp giải quyết bài toán bằng MT

##### 2. Chương 2: Thuật toán

- Khái niệm
- Biểu diễn thuật toán
- Thuật toán đệ quy
- Thuật giải heuristic
- Một số thuật toán thông dụng

2

2

### Chương 1: Giải quyết bài toán

#### Nội dung chính

##### 1. Khái niệm về bài toán

2. Quá trình giải quyết bài toán bằng máy tính
3. Phương pháp giải quyết bài toán bằng máy tính

3

3

### Chương 1: Giải quyết bài toán

#### 1. Khái niệm bài toán

#### Problem – Bài toán hay vấn đề?

- Theo Socrate (470-399 TCN): Vấn đề thường được dùng với ý nghĩa rộng hơn bài toán
- Bài toán là vấn đề mà để giải quyết phải liên quan ít nhiều đến tính toán
  - Bài toán trong vật lý, hóa học, xây dựng, kinh tế,...

4

4

## Phân loại vấn đề (Pytago)

- **Theorema:**
  - Vấn đề cần khẳng định đúng sai
    - Ví dụ: Chứng minh các định lý trong toán học
- **Problema:**
  - Vấn đề cần tìm giải pháp để đạt mục tiêu xác định từ những điều kiện ban đầu
    - Ví dụ: Bài toán dựng hình, tìm đường đi ngắn nhất, tổng hợp chất hóa học...

5

5

## Biểu diễn vấn đề (1/3)

$$A \rightarrow B$$

- A: Giả thiết, điều kiện ban đầu
- B: Kết luận, mục tiêu cần thực hiện
- $\rightarrow$ : Suy luận, giải pháp cần xác định

6

6

## Biểu diễn vấn đề (2/3)

- Cho vấn đề/bài toán:  
Cho A và B
- Giải quyết vấn đề/bài toán:  
Từ A dùng một số **hữu hạn các bước** suy luận có lý hoặc hành động thích hợp để đạt B.  
Cần xác định tập các thao tác cơ bản được dùng trong suy luận và hành động

7

7

## Biểu diễn vấn đề (3/3)

### Trong tin học

$$A \rightarrow B$$

- A: Input
- B: Output
- $\rightarrow$ : Chương trình cho phép biến đổi A thành B .

8

8

## Chương trình

- Chương trình
  - Cách mã hóa lại thuật toán/thuật giải để giải quyết vấn đề/bài toán đã cho
  - Tạo thành từ các lệnh cơ bản của máy tính
- Khó khăn:
  - Tồn tại các yếu tố không xác định
    - A và B không đầy đủ, rõ ràng
- Giải quyết bài toán trên máy tính?
  - Vấn đề tổ chức dữ liệu và thiết kế giải thuật

Cấu trúc dữ liệu + Giải thuật = Chương trình

9

## Thiết kế thuật giải

- Thực hiện bởi con người
  - Là cách thức chủ yếu, dựa trên
    - Những thông tin được phản ánh rõ ràng trong A, B hoặc →
    - Các tri thức của con người
- Tự động hóa xây dựng thuật giải
  - Lĩnh vực mới, đang được nghiên cứu
  - Cần phải biểu diễn nội dung và các tri thức liên quan dưới dạng tương minh và đầy đủ

10

10

## Nội dung chính

### 1. Khái niệm về bài toán

### 2. Quá trình giải quyết bài toán bằng máy tính

### 3. Phương pháp giải quyết bài toán bằng máy tính

11

11

## Máy tính & Lập trình viên

- Máy tính
  - Chỉ làm được những gì được bảo.
  - **Không thông minh:** không thể tự phân tích vấn đề và đưa ra giải pháp.
  - Không thể dùng giải quyết các vấn đề liên quan đến hành động vật lý hoặc biểu thị cảm xúc
- Lập trình viên
  - Phân tích vấn đề
  - Tạo ra các chỉ dẫn để giải quyết vấn đề (xây dựng chương trình).
    - Máy tính sẽ thực hiện các chỉ dẫn này.



12

12

1. Xác định bài toán
2. Lựa chọn phương pháp giải
3. Xây dựng thuật toán hoặc thuật giải
4. Cài đặt chương trình
5. Hiệu chỉnh chương trình
6. Thực hiện chương trình

13

13

- Mô tả bài toán cần giải quyết
  - **Dữ liệu vào:** Danh sách các dữ kiện vào
  - **Điều kiện vào:** Ràng buộc, quan hệ giữa chúng
  - **Dữ liệu ra:** Danh sách các dữ liệu ra
  - **Điều kiện ra:** Ràng buộc, quan hệ giữa chúng
- Đánh giá, nhận định tính khả thi của bài toán
  - Thời gian, kinh phí, nguồn lực,...

**Ví dụ:** Bài toán tìm ƯSCLN của 2 số nguyên dương

- **Nhập:** 2 số  $X, Y$
- **Điều kiện nhập:**  $X, Y$  nguyên dương
- **Dữ liệu ra:**  $Z$
- **Điều kiện ra:**  $Z$  là ƯSCLN( $X, Y$ )

14

14

- Tồn tại nhiều phương pháp khác nhau
  - Khác nhau về thời gian thực hiện, chi phí lưu trữ dữ liệu, độ chính xác...
- Tùy theo nhu cầu cụ thể và khả năng xử lý tự động được sử dụng để lựa chọn phương pháp thích hợp

**Ví dụ:** Bài toán sắp xếp dãy số

- Nổi bọt, Vun đống, Sắp xếp nhanh,...

15

15

- Xây dựng mô hình chặt chẽ, chính xác và chi tiết cho phương pháp đã lựa chọn
- Lập liên tiếp các bước sau để thuật toán ngày càng hoàn chỉnh hơn (*quá trình tinh chỉnh từng bước*)
  1. Xác định và chính xác hóa các thao tác
    - Để đạt được kết quả cần làm gì?
  2. Xác định các dữ liệu cần dùng và tính chất của chúng
    - Để thực hiện, thao tác cần gì và sẽ tạo ra gì?
  3. Xác định trình tự các thao tác
    - Thao tác nào cần làm trước
    - Thao tác thực hiện 1 hay nhiều lần, thực hiện trong điều kiện nào..?

16

16

### Bước 3: Xây dựng thuật giải (tiếp)

- *Quá trình tinh chỉnh từng bước* dừng lại khi
  - Yêu cầu cho biết 1 hay nhiều đại lượng
  - Tính một đại lượng theo công thức đã biết rõ
  - Thông báo một hay nhiều kết quả đã xử lý
- Sau khi tinh chỉnh cần phải diễn tả giải thuật dưới dạng chuẩn
  - Ngôn ngữ liệt kê các hành động
  - Sơ đồ khối...

17

17

### Bước 4: Cài đặt chương trình

Mã hóa giải thuật bằng một ngôn ngữ lập trình

- Thay thế các thao tác bằng các lệnh tương ứng của ngôn ngữ sử dụng
  - **Thao tác:** In ra một thông báo
  - **Câu lệnh:** `printf(".....")/ write(".....");`
- Lựa chọn ngôn ngữ lập trình, tùy theo bài toán giải quyết
  - NNLT bậc thấp: Hợp ngữ
  - NNLT bậc cao: C, Pascal, Java,...

18

18

### Bước 5: Hiệu chỉnh chương trình

Chạy thử để phát hiện và điều chỉnh các sai sót có thể có ở bước 4.

- Lỗi cú pháp:
  - Viết sai cú pháp của ngôn ngữ lập trình lựa chọn
- Lỗi ngữ nghĩa
  - Mã hóa sai giải thuật
  - Giải thuật sai

19

19

### Bước 6: Thực hiện chương trình

- Cho máy tính thực hiện chương trình.
- Tiến hành phân tích kết quả thu được
  - Kết quả đó có phù hợp hay không.
  - Không phù hợp kiểm tra lại toàn bộ các bước.

20

20

### Các giai đoạn giải quyết bài toán

#### 1. Giai đoạn quan niệm :

- Gồm các bước xác định bài toán, lựa chọn mô hình, xây dựng thuật giải, cài đặt chương trình

#### 2. Giai đoạn khai thác và bảo trì

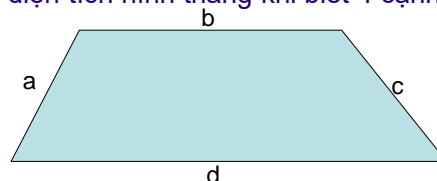
- Gồm các bước hiệu chỉnh và thực hiện chương trình
- Nhằm đáp ứng nhu cầu về cải tiến, mở rộng chương trình
  - Do các yếu tố ban đầu của bài toán có thể thay đổi.

21

21

### Ví dụ

Tính diện tích hình thang khi biết 4 cạnh



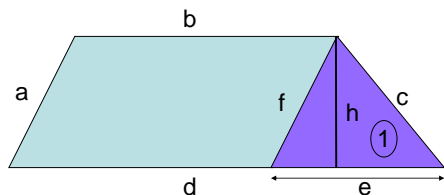
#### Mô tả bài toán

- Nhập: 4 cạnh a, b, c, d
- Điều kiện nhập: a, b, c, d > 0 và d > b
- Xuất: Một giá trị số
- Điều kiện xuất: Diện tích hình thang

22

22

### Ví dụ→Xây dựng thuật toán



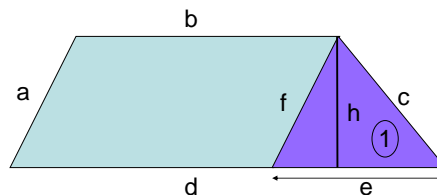
1. Để tính diện tích hình thang, cần tính đường cao (công thức  $S = h(b+d)/2$ )
2. Tính đường cao h, cần phải biết 3 cạnh của tam giác (1)  $h_c = \frac{2\sqrt{p(p-a)(p-b)(p-c)}}{c}$
3. Cần tính cạnh tam giác (1) trước khi tính đường cao h

3/20/2019

23

23

### Ví dụ→Xây dựng thuật toán



#### Lặp lại các bước

- Để tính cạnh của tam giác (1) cần biết các cạnh của hình thang
- Các cạnh của hình thang là dữ kiện cho biết của đề bài (điều kiện dừng)

3/20/2019

24

24

## Ví dụ → Chuẩn hóa thuật toán

1. Nhập các số a, b, c, d
2. Tính các cạnh của tam giác (1)
  - $f \leftarrow a$
  - $e \leftarrow d - c$
  - $p \leftarrow (f + e + c) / 2$

3. Tính chiều cao của tam giác (1)

$$h = \frac{2\sqrt{p(p-e)(p-f)(p-c)}}{e}$$

4. Tính diện tích hình thang  $S = h(d+b)/2$
5. In kết quả S
6. Kết thúc

3/20/2019

25

25

## Nội dung chính

1. Khái niệm về bài toán
2. Quá trình giải quyết bài toán bằng máy tính

### 3. Phương pháp giải quyết bài toán bằng máy tính

26

26

## Các phương pháp

1. Xác định trực tiếp lời giải
2. Tìm kiếm lời giải

27

27

## Hướng xác định trực tiếp lời giải

- Thường sử dụng trong quá trình học tập.
  - Ví dụ: Tìm nghiệm phương trình bậc 2 theo định lý Viet.
- Xác định trực tiếp được lời giải qua
  - Các thủ tục tính toán theo công thức, hệ thức, định luật...
  - Các thủ tục bao gồm một số hữu hạn các thao tác sơ cấp, có thể chuyển thành các thuật toán và chương trình chạy trên máy tính.

28

28

### Hướng xác định trực tiếp lời giải

- Trường hợp dùng các công thức lập để tính gần đúng nghiệm của bài toán.
  - Lời giải xác định bởi các công thức lập có thể xấp xỉ lời giải thật sự của bài toán với độ chính xác tăng theo quá trình lập.
    - Ví dụ: giải phương trình  $f(x)=0$  bằng PP chia đôi
  - Đây là hạn chế khi tính toán thủ công nhưng là thế mạnh của máy tính.
  - Được xem là cách xác định trực tiếp lời giải

29

29

### Hướng tìm kiếm lời giải

- Cách tiếp cận dựa theo nguyên lý “**thử - sai**”.
- Ứng dụng hiệu quả cho một số bài toán - vấn đề phức tạp
- Nhiều phương pháp đề xuất

30

30

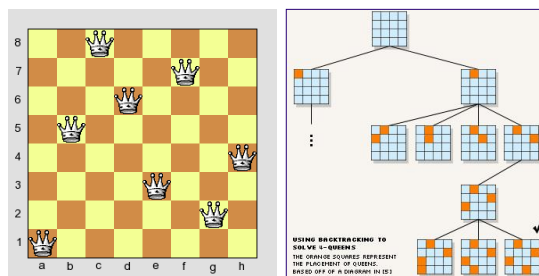
### Hướng tìm kiếm lời giải→Một số phương pháp

- Phương pháp liệt kê hay vét cạn:
  - Xác định tập các khả năng chứa các lời giải và cách thức liệt kê của từng khả năng để thử, không bỏ sót một khả năng nào.
- Phương pháp thử ngẫu nhiên:
  - Thử một số khả năng được chọn ngẫu nhiên trong tập (*rất lớn*) các khả năng.
  - Khả năng thành công tùy theo chiến lược chọn ngẫu nhiên và một số điều kiện cụ thể của bài toán.
- Chia bài toán thành bài toán con:
  - Chia cho tới khi bài toán ban đầu được quy thành bài toán con có lời giải
- Phương pháp quay lui:
  - Đánh dấu các thử nghiệm thất bại và thử khả năng mới (quay lui tìm đường khác).

31

31

### Hướng tìm kiếm lời giải→Ví dụ bài toán 8 hậu



32

32



## Phần 2: Giải quyết bài toán

### Nội dung chính

#### 1. Chương 1: Giải quyết bài toán

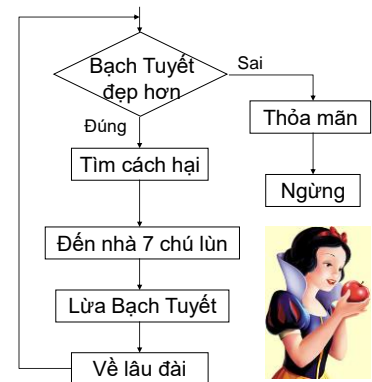
- Khái niệm về bài toán
- Quá trình giải quyết bài toán bằng máy tính
- Phương pháp giải quyết bài toán bằng MT

#### 2. Chương 2: Thuật toán

- Khái niệm
- Biểu diễn thuật toán
- Thuật toán đệ quy
- Thuật giải heuristic
- Một số thuật toán thông dụng

33

33



34

34

## Chương 2: Thuật toán

### Nội dung chính

#### 1. Khái niệm

#### 2. Biểu diễn thuật toán

#### 3. Thuật toán đệ quy

#### 4. Thuật giải heuristic

#### 5. Một số thuật toán thông dụng

35

35

## Chương 2: Thuật toán

### 1. Khái niệm

#### Khái niệm

- Thuật toán (*algorithm*) là khái niệm cơ sở của Toán học và Tin học
- Nghiên cứu thuật toán đóng vai trò quan trọng trong khoa học máy tính
  - Máy tính chỉ có khả năng thực hiện công việc theo một thuật toán.
  - Thuật toán chỉ đạo máy tính từng bước phải làm gì.

Thuật toán là gì?

36

36

## Khái niệm

- Một tập các lệnh hay chỉ thị nhằm hướng dẫn việc thực hiện một công việc nào đó
- Bao gồm một dãy hữu hạn các chỉ thị rõ ràng và có thể thi hành được, được bố trí theo một trình tự nhất định, cần thực hiện trên những dữ liệu vào sao cho sau một số hữu hạn bước ta thu được kết quả của bài toán cho trước
- Thuật toán là sự thể hiện của một phương pháp để giải quyết một vấn đề

37

37

## Ví dụ

Tìm phần tử lớn nhất trong một dãy hữu hạn các số nguyên

1. Đặt giá trị lớn nhất tạm thời (Max) bằng số nguyên đầu tiên của dãy  
Max là giá trị lớn nhất ở mỗi giai đoạn thực hiện
2. Nếu tất cả số nguyên trong dãy đã được xét, thực hiện bước 5
3. So sánh số nguyên kế tiếp trong dãy với Max
  - Nếu lớn hơn Max thì thay Max bằng số nguyên này.
4. Lặp lại bước 2
5. Thông báo: Max là giá trị lớn nhất trong dãy số.

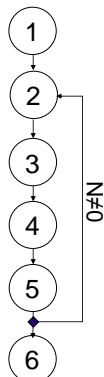
38

38

## Ví dụ

Đổi số thập phân sang dạng nhị phân

1. Cho biết N
2. Chia N cho 2
3. Ghép phần dư vào bên trái kết quả
4. Lấy phần thương làm N mới
5. Nếu N khác 0, lặp lại Bước 2
6. Xong



39

39

## Định nghĩa (KHMT)

*Thuật toán để giải một bài toán là một dãy hữu hạn các thao tác và trình tự thực hiện các thao tác đó sao cho sau khi thực hiện dãy thao tác này theo trình tự đã chỉ ra, với đầu vào (input) ta thu được kết quả đầu ra (output) mong muốn*

40

40

## Thao tác/lệnh

- Là hành động cần được thực hiện bởi cơ chế của thuật toán
- Các thao tác (*lệnh*) sẽ biến đổi bài toán từ trạng thái trước tới trạng thái sau
- Dãy các thao tác cần thiết sẽ biến đổi bài toán từ trạng thái ban đầu đến kết quả
- Các thao tác có thể phân tích thành thao tác khác nhỏ hơn
- Thứ tự thao tác là quan trọng
  - Cùng tập thao tác, thứ tự khác nhau dẫn đến kết quả khác nhau
- Cơ cấu thể hiện trình tự thực hiện các thao tác gọi là **Cấu trúc điều khiển**
  - Có 3 loại cơ bản: Tuần tự, Lặp, Rẽ nhánh

41

41

## Các đặc trưng của thuật toán

Khi mô tả thuật toán, cần chú ý các đặc trưng

- Nhập (input)
- Xuất (output)
- Tính xác định (definiteness)
- Tính hữu hạn (finiteness)
- Tính hiệu quả
- Tính tổng quát

42

42

## Nhập/Xuất

- **Nhập (input):**
  - Các giá trị “đầu vào” (input values) từ một tập hợp nhất định nào đó.
- **Xuất (output):**
  - Những giá trị trả về (output values) thuộc một tập hợp nhất định nào đó thể hiện lời giải cho bài toán/vấn đề
  - Tương ứng với tập hợp các giá trị nhập

43

43

## Tính xác định (definiteness)

- Các bước trong thuật toán phải chính xác rõ ràng, không gây sự nhập nhằng nhầm lẫn
- Cùng một điều kiện nhập, cùng một giải thuật thì 2 bộ VXL (người, máy) phải cho ra cùng một kết quả

44

44

## Tính hữu hạn (finiteness)

- Trong mọi trường hợp của dữ liệu vào, thuật toán phải cho ra kết quả sau một thời gian hữu hạn
  - Thời gian có thể phụ thuộc vào từng bài toán cụ thể, dữ liệu cụ thể các thuật toán khác nhau cho một bài toán
  - Ví dụ bài toán sắp xếp dãy số
    - Bubble sort, Quick sort,...

45

45

## Tính hiệu quả

- Thực hiện thuật toán cần
  - Thời gian
  - Các công cụ hỗ trợ (giấy, bộ nhớ,...)
    - Để ghi kết quả trung gian
- Độ phức tạp thuật toán: Thời gian và các công cụ hỗ trợ
  - Thuật toán càng hiệu quả độ phức tạp càng bé
  - Trong máy tính, thường quan tâm tới
    - Thời gian thực hiện
      - Số thao tác cơ bản cần thực hiện
    - Độ lớn của bộ nhớ mà thuật toán sử dụng

46

46

## Tính tổng quát

Thuật toán có tính tổng quát cao nếu có thể giải bất kỳ bài toán nào trong một lớp lớn các bài toán

### Ví dụ

Thuật toán giải phương trình  $ax^2+bx+c=0$  phổ dụng hơn thuật toán giải phương trình  $x^2+5x+6=0$

47

47

## Nội dung chính

1. Khái niệm
2. Biểu diễn thuật toán
3. Thuật toán đệ quy
4. Thuật giải heuristic
5. Một số thuật toán thông dụng

48

48

- Tại sao:
  - Truyền đạt thuật toán cho người khác
  - “*Truyền đạt*” thuật toán cho máy tính
    - Chuyển thành chương trình điều khiển
- Phương pháp:
  1. Ngôn ngữ tự nhiên
  2. Ngôn ngữ lưu đồ (sơ đồ khối)
  3. Ngôn ngữ tựa ngôn ngữ lập trình (mã giả)
  4. Ngôn ngữ lập trình

49

49

- Nguyên tắc:
  - Sử dụng ngôn ngữ tự nhiên để liệt kê các bước của thuật toán
- Đặc điểm
  - Không yêu cầu phải có một số kiến thức đặc biệt
  - Dài dòng,
  - Không làm nổi bật cấu trúc của thuật toán.

50

50

Giải phương trình  $ax + b = 0$

- B1: Nhập a
- B2: Nhập b.
- B3: Nếu a = 0 thực hiện B6
- B4: **Thông báo:** Nghiệm  $-b/a$
- B5: Thực hiện B10
- B6: Nếu b = 0, thực hiện B9
- B7: **Thông báo:** Phương trình vô nghiệm.
- B8: Thực hiện B10
- B9: **Thông báo:** Phương trình vô số nghiệm.
- B10: Kết thúc

51

51

Tìm giá trị lớn nhất của một dãy N số nguyên

- B1: Nhập N
- B2: Nhập dãy số  $a_i$  gồm N số.
- B3: Gán giá trị  $a_1$  cho Max, 2 cho biến i ( $i \leftarrow 2$ )
- B4: Nếu  $i > N$ , thực hiện bước 8
- B5: Nếu  $a_i > \text{Max}$ , gán giá trị  $a_i$  cho Max.
- B6: Tăng i lên 1 đơn vị.
- B7: Quay lên B4.
- B8: Thông báo: Max là giá trị lớn nhất dãy
- B9: Kết thúc.

52

52

Chương 2: Thuật toán  
2.2 Biểu diễn thuật toán  
2.2 Ngôn ngữ lưu đồ (sơ đồ khối)

Công cụ diễn đạt các thuật toán trực quan

- Đưa ra một cái nhìn tổng quan về quá trình xử lý theo thuật toán
- Gồm hệ thống các nút có hình dạng khác nhau, thể hiện các chức năng khác nhau, được nối với nhau bởi các cung

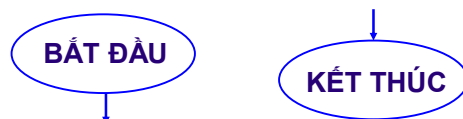
Thành phần chủ yếu của thuật toán

53

53

Chương 2: Thuật toán  
2.2 Biểu diễn thuật toán  
2.2 Sơ đồ khối → Nút /khối giới hạn

- 2 loại nút giới hạn: nút đầu và nút cuối
- Ghi rõ điểm bắt đầu và kết thúc (dừng) của thuật toán
- Được biểu diễn bởi hình ôvan có ghi chữ bên trong

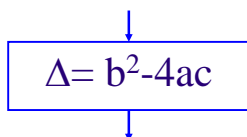


54

54

Chương 2: Thuật toán  
2.2 Biểu diễn thuật toán  
2.2 Sơ đồ khối → Nút/Khối thao tác

Là một hình chữ nhật chứa dãy các lệnh cần thực hiện như gán, tính toán...

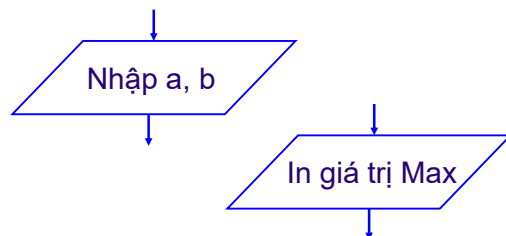


55

55

Chương 2: Thuật toán  
2.2 Biểu diễn thuật toán  
2.2 Sơ đồ khối → Nút/khối vào/ra dữ liệu

Là một hình bình hành chứa đựng một thao tác nhập/ xuất dữ liệu

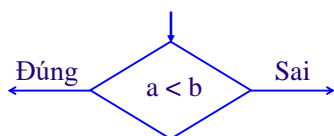


56

56

## 2.2 Sơ đồ khối → Nút/khối điều kiện

Là một hình thoi chứa một điều kiện/biểu thức logic cần kiểm tra.



Nút điều kiện có 2 cung ra chỉ hướng ứng với 2 trường hợp: điều kiện đúng và điều kiện sai

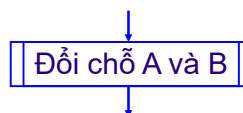
57

57

## 2.2 Sơ đồ khối → Nút/khối gọi chương trình con

Là một hình chữ nhật, cạnh kép chứa tên một chương trình con cần thực hiện

- Chương trình con: Thuật toán đã biết
- Nhằm cho sơ đồ đỡ rắc rối

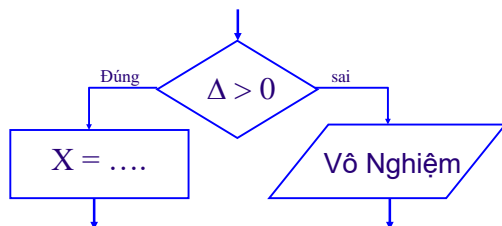


58

58

## 2.2 Sơ đồ khối → Cung

Là các đường nối từ nút này đến nút khác của lưu đồ



59

59

## 2.2 Sơ đồ khối → Hoạt động

- Bắt đầu từ nút giới hạn đầu tiên (nút bắt đầu).
- Thực hiện các thao tác được ghi trong nút
- Theo một cung đi tới nút khác
- Nếu là nút điều kiện: đi theo cung tương ứng với trạng thái của điều kiện được kiểm tra.
- Thuật toán sẽ dừng khi gặp nút kết thúc

60

60

## 2.2 Sơ đồ khối → Ví dụ 1 → Biểu diễn bằng lời

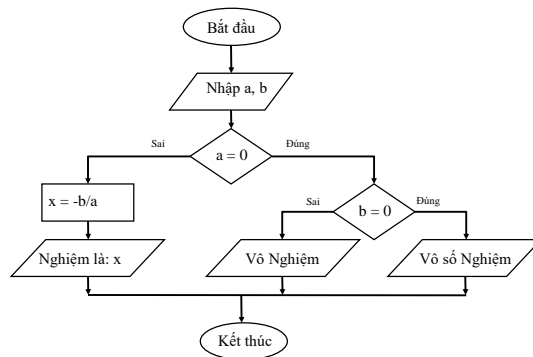
Giải phương trình  $ax + b = 0$

- B1: Nhập a
- B2: Nhập b.
- B3: Nếu  $a = 0$  thực hiện B6
- B4: **Thông báo:** Nghiệm  $-b/a$
- B5: Thực hiện B10
- B6: Nếu  $b = 0$ , thực hiện B9
- B7: **Thông báo:** Phương trình vô nghiệm.
- B8: Thực hiện B10
- B9: **Thông báo:** Phương trình vô số nghiệm.
- B10: Kết thúc

61

61

## 2.2 Sơ đồ khối → Ví dụ 1 → Biểu diễn sơ đồ khối



62

62

## 2.2 Sơ đồ khối → Ví dụ 2 → Biểu diễn bằng lời

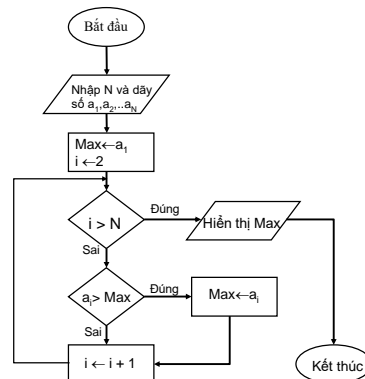
Tìm giá trị lớn nhất của một dãy N số nguyên

- B1: Nhập N
- B2: Nhập dãy số  $a_i$  gồm N số.
- B3: Gán giá trị  $a_1$  cho Max,  $i \leftarrow 2$ .
- B4: Nếu  $i > N$ , thực hiện bước 8
- B5: Nếu  $a_i > \text{Max}$ , gán giá trị  $a_i$  cho Max.
- B6: Tăng i lên 1 đơn vị.
- B7: Quay lên B4.
- B8: Thông báo: Max là giá trị lớn nhất dãy
- B9: Kết thúc.

63

63

## 2.2 Sơ đồ khối → Ví dụ 2 → Biểu diễn sơ đồ khối



64

64



### 2.3 Ngôn ngữ tựa ngôn ngữ lập trình (mã giả)

- Mô tả thuật toán theo ngôn ngữ tựa ngôn ngữ lập trình
  - Sử dụng các mệnh đề có cấu trúc chuẩn hóa
  - Vẫn dùng ngôn ngữ tự nhiên.
    - Có thể sử dụng các ký hiệu toán học
    - Có thể sử dụng cấu trúc kiểu thủ tục để trình bày thuật toán đệ quy/ thuật toán phức tạp..
- Đặc điểm:
  - Tiện lợi, đơn giản, và dễ hiểu.
- Các cấu trúc thường gặp:
  - Gán, lựa chọn, lặp, nhảy, gọi hàm

65

65

### 2.3 Mã giả → Phát biểu gán

- Mục đích:
  - Đặt giá trị cho một biến
- Biểu diễn:

$\text{Max} := a_1$

$\text{Max} \leftarrow a_1$

$n \leftarrow n + 1$

66

66

### 2.3 Mã giả → Lựa chọn/rẽ nhánh

**if** <điều kiện> **then** <hành động>  
**endif**

*Hoặc là*

**if** <điều kiện> **then** <hành động>  
**else** <hành động>  
**endif**

67

67

### 2.3 Mã giả → Cấu trúc lặp

**while** <điều kiện> **do**  
    <hành động>

**end while**

**repeat**  
    <hành động>

**until** <điều kiện>

**for** biến ← giá trị đầu **to** giá trị cuối **do** <hành động>

**end for**

**for** biến ← giá trị đầu **downto** giá trị cuối **do** <h/động>  
**end for**

68

68

Chương 2: Thuật toán  
2.3 Mã giả → Lệnh nhảy không điều kiện

Nhảy đến vị trí có nhãn là L  
goto L

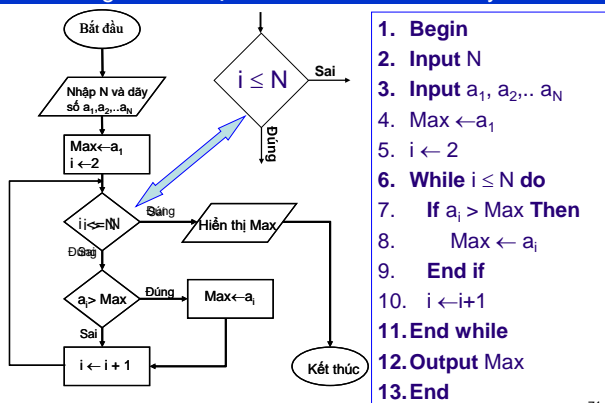
69

Chương 2: Thuật toán  
2.3 Mã giả → Hàm/Thủ tục

- Khai báo hàm  
**Function** <Tên hàm>(<Các tham số>)  
Hành động với các tham số  
**return** <Giá trị>  
**End Function**
- Gọi hàm  
**[Call]** <Tên hàm>(<Các tham số>)

70

Chương 2: Thuật toán  
2.3 Mã giả → Ví dụ: tìm số lớn nhất của dãy



71

Chương 2: Thuật toán  
2.4 Ngôn ngữ lập trình

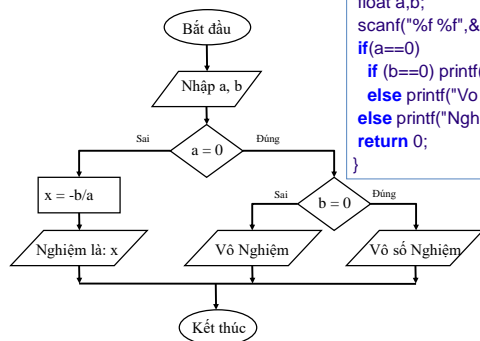
- Tuân theo cú pháp của ngôn ngữ lập trình
  - Cấu trúc tuần tự
  - Cấu trúc rẽ nhánh
  - Cấu trúc lặp
  - Chương trình con
- Tồn tại nhiều loại ngôn ngữ lập trình
  - Ngôn ngữ máy /Hợp ngữ
  - Ngôn ngữ bậc cao:
    - Fortran, Pascal, C/C++/C#,Java

72

72

Chương 2: Thuật toán  
2.4 Ngôn ngữ lập trình → Ví dụ

Giải phương trình  $ax+b=0$



```

#include <stdio.h>
int main(){
    float a,b;
    scanf("%f %f",&a,&b);
    if(a==0)
        if (b==0) printf("Vo so nghiem");
        else printf("Vo nghiem");
    else printf("Nghiem %f",-b/a);
    return 0;
}
  
```

73

73

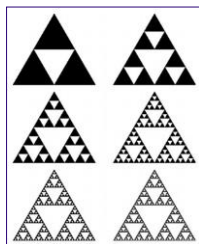
Chương 2: Thuật toán  
Nội dung chính

1. Khái niệm
2. Biểu diễn thuật toán
3. Thuật toán đệ quy
4. Thuật giải heuristic
5. Một số thuật toán thông dụng

74

74

Chương 2: Thuật toán  
3. Thuật toán đệ quy  
Ví dụ đệ quy



75

75

Chương 2: Thuật toán  
3. Thuật toán đệ quy  
Khái niệm thuật toán đệ quy

- Bài toán có thể được phân tích và đưa tới việc giải một bài toán cùng loại nhưng cấp độ thấp hơn,
    - Độ lớn dữ liệu vào nhỏ hơn
    - Giá trị cần tính toán nhỏ hơn
  - **Ví dụ:** Định nghĩa giai thừa  
 Giai thừa của một số tự nhiên  $n$ , ký hiệu là  $n!$ , được định nghĩa bằng cách quy nạp như sau:
    - $0!=1$ ,
    - $n!=(n-1)!*n$ , với mọi  $n>0$
  - Giải một bài toán có thể dựa trên chính nó
    - Mỗi bước của thuật toán hiện lại chính thuật toán đó
    - Dữ liệu vào có độ lớn thấp hơn
- Thuật toán đệ qui.**

76

76

### Ví dụ 1: Tính giai thừa của một số tự nhiên

- Input: số tự nhiên  $n$
- Output:  $F(n)=n!$

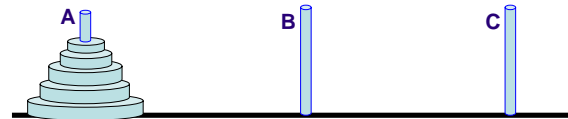
#### Thuật Toán:

```
Function F(n);
  If n=0 then
    Return 1
  Else
    Return n * F(n-1)
  End If
End Function
```

77

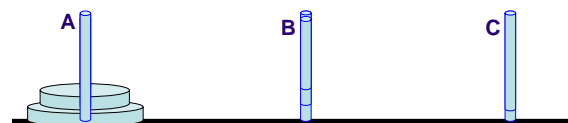
77

### Ví dụ 2: Bài toán tháp Hà nội



Yêu cầu: Dịch chuyển  $N$  đĩa từ cọc A sang B với trung gian là cọc C

Trường hợp với  $N=2$ :

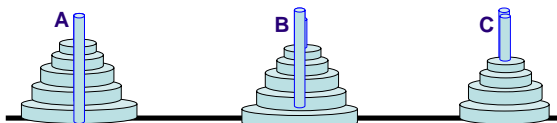


78

78

### Ví dụ 2: Bài toán tháp Hà nội

Trường hợp tổng quát với  $N > 2$ :



```
PROCEDURE ThapHanoi(N,A,B,C)
  IF N= 2 Then Print(A→C, A→B, C→B)
  ELSE
    ThapHanoi(N-1,A,C,B)
    Print(A →B)
    ThapHaNoi(N-1,C,B,A)
  ENDIF
END
```

79

79

### Lưu ý

- Thuật toán đệ quy gồm 2 phần
  - Phần cơ sở:** không cần thực hiện lại thuật toán
    - Không có yêu cầu gọi đệ qui.
    - Là điều kiện dừng cử giả thuật đệ quy
  - Phần đệ qui:** có yêu cầu gọi đệ qui
    - Yêu cầu thực hiện lại thuật toán
    - Đặt trong một điều kiện kiểm tra việc gọi đệ qui.
- Đệ qui dễ gây tình trạng tràn bộ nhớ (stack).
- Nếu có thể, nên viết dưới dạng lặp.

```
P←1
For k ←1 To N Do P ←P*k
Print P
```

80

80

## Chương 2: Thuật toán

### Nội dung chính

1. Khái niệm
2. Biểu diễn thuật toán
3. Thuật toán đệ quy
4. Thuật giải heuristic
5. Một số thuật toán thông dụng

81

81

## Chương 2: Thuật toán

### 4. Thuật giải Heuristic

#### Vấn đề mở rộng khái niệm thuật toán

- Có những bài toán đến nay vẫn chưa có một cách giải theo kiểu thuật toán được tìm ra và cũng không biết có tồn tại thuật toán hay không.
- Có những bài toán đã có thuật toán để giải nhưng không chấp nhận được vì thời gian giải theo thuật toán đó quá dài hoặc các điều kiện cho thuật toán khó đáp ứng
- Có những bài toán được giải theo cách giải vi phạm thuật toán nhưng vẫn được chấp nhận.

#### Cần phải đổi mới cho khái niệm thuật toán

- Mở rộng những tiêu chuẩn của thuật toán

82

82

## Chương 2: Thuật toán

### 4. Thuật giải Heuristic

#### Mở rộng tiêu chuẩn của thuật toán

- Tính xác định (*tính đơn trị của mỗi bước*)
  - Các giải thuật đệ quy: bước tiếp gọi đến chính nó
  - Các giải thuật ngẫu nhiên: bước tiếp không xác định rõ
- Tính đúng đắn (*được hiểu cho kết quả đúng*)
  - Không còn bắt buộc với một số cách giải cho các bài toán nhất là các cách giải gần đúng.
  - Trong thực tế có nhiều trường hợp, chấp nhận các cách giải cho kết quả gần đúng nhưng ít phức tạp và hiệu quả
- Ví dụ: trong trí tuệ nhân tạo
  - Cách giải theo kiểu heuristic. Đơn giản, tự nhiên nhưng cho kết quả đúng hoặc gần đúng trong phạm vi cho phép

83

83

## Chương 2: Thuật toán

### 4. Thuật giải Heuristic

#### Thuật giải Heuristic

- **Khái niệm thuật giải:**
  - Các cách giải chấp nhận được nhưng không hoàn toàn đáp ứng đầy đủ các tiêu chuẩn của thuật toán
- **Thuật giải heuristic**
  - Thể hiện cách giải bài toán với các đặc tính sau:
    - Tìm được lời giải tốt (không chắc là tốt nhất)
    - Dễ dàng và nhanh chóng hơn so với giải thuật tối ưu
    - Thể hiện một cách hành động khá tự nhiên, gần gũi với cách suy nghĩ và hành động của con người.

84

84

## Nguyên lý thiết kế thuật giải heuristic

- **Nguyên lý vết cận thông minh:**
  - Trong một bài toán tìm kiếm, khi không gian tìm kiếm lớn, thường tìm cách để giới hạn lại không gian hoặc thực hiện một kiểu dò tìm đặc biệt dựa vào đặc thù của bài toán để nhanh chóng tìm ra mục tiêu.
- **Nguyên lý tham lam:**
  - Lấy tiêu chuẩn tối ưu (*trên phạm vi toàn bộ*) của bài toán để làm tiêu chuẩn chọn lựa hành động cho phạm vi cục bộ của từng bước (*hay từng giai đoạn*) trong quá trình tìm kiếm lời giải.
- **Nguyên lý thứ tự:**
  - Thực hiện hành động dựa trên một cấu trúc thứ tự hợp lý của không gian khảo sát nhằm nhanh chóng đạt được một lời giải tốt..

85

85

## Nội dung chính

1. Khái niệm
2. Biểu diễn thuật toán
3. Thuật toán đệ quy
4. Thuật giải heuristic
5. Một số thuật toán thông dụng

86

86

## Các bài toán

1. Thuật toán số học
  - Hoán đổi giá trị
  - Số nguyên tố, phân tích ra thừa số nguyên tố...
  - Tìm ước số chung, phân số tối giản
  - Số hoàn hảo
2. Thuật toán về dãy
  - Vào/ra dãy
  - Tìm Max, Min
  - Sắp xếp
  - Tìm phần tử: Đếm phần tử
  - Tính toán trên các phần tử..
    - Trung bình cộng, tính tổng,...
  - Chèn phần tử/Xóa phần tử (*liên quan tới kiểu mảng*)

87

87

## Hoán đổi giá trị 2 biến X, Y

Nguyên tắc:

Dùng một biến trung gian T

**Begin**

1.  $T \leftarrow X$

2.  $X \leftarrow Y$

3.  $Y \leftarrow T$

**End**



Function *swap*(X,Y)

$T \leftarrow X$

$X \leftarrow Y$

$Y \leftarrow T$

End Function

88

88

## Kiểm tra số nguyên tố

Begin

1. Input P
2. Flag  $\leftarrow$  FALSE
3. If P=1 Then Goto Bước 6
4. flag  $\leftarrow$  TRUE (gán cho cờ hiệu "flag" giá trị true)
5. For k:=2 to p-1 do
  - If (k là ước số của P) Then
  - flag  $\leftarrow$  FALSE;
  - Goto B6
- Endif
- End for
6. If flag=TRUE Then Output: P là số nguyên tố
- Else Output: P không là số nguyên tố
- Endif

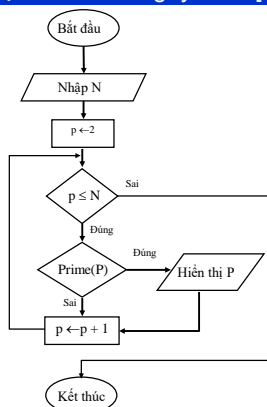
End

Function Prime(P):Bool  
Flag  $\leftarrow$  FALSE  
...  
return Flag  
End Function

89

89

## Liệt kê các số nguyên tố [2..N]



Begin

- Input N
- For p  $\leftarrow$  2 to N do
- If Prime(P) Then
- Output: P
- Endif
- End For
- End

90

90

## Tìm ƯSCLN của 2 số nguyên dương (1/3)

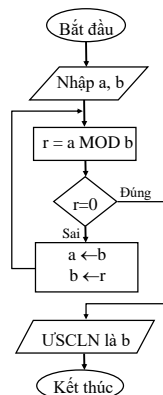
1. Nhập số a
2. Nhập số b
3. Chia a cho b với số dư là r
4. Nếu r = 0 thực hiện 6
5. Nếu r  $\neq$  0 gán giá trị b cho a, giá trị r cho b và quay lại bước 3
6. Thông báo kết quả ƯSCLN là b
7. Kết thúc



91

91

## Tìm ƯSCLN của 2 số nguyên dương (2/3)



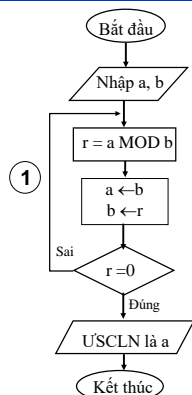
Begin

1. Input a, b
  2. Repeat
  3. r  $\leftarrow$  a MOD b
  4. if r=0 then goto 8
  5. a  $\leftarrow$  b
  6. b  $\leftarrow$  r
  7. Until r=0
  8. Output USCLN là: b
- End

92

92

Chương 2: Thuật toán  
 5. Một số thuật toán thông dụng  
 Tìm ƯSCLN của 2 số nguyên dương (3/3)



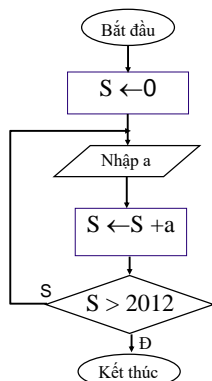
**Begin**  
 1. Input a, b  
 2. Repeat  
 3.  $r \leftarrow a \text{ MOD } b$   
 4.  $a \leftarrow b$   
 5.  $b \leftarrow r$   
 6. Until  $r=0$   
 7. Output *USCLN* là: a  
**End**

93

Chương 2: Thuật toán  
 5. Một số thuật toán thông dụng  
 Giải phương trình  $ax^2+bx+c = 0$

94

Chương 2: Thuật toán  
 5. Một số thuật toán thông dụng  
 Nhập dãy số cho tới khi tổng lớn hơn 1012



**Begin**  
 1.  $S \leftarrow 0$   
 2. Repeat  
 Input a  
 $S \leftarrow S + a$   
 3. Until  $S > 1012$   
**End**

95

Chương 2: Thuật toán  
 5. Một số thuật toán thông dụng  
 Bài tập tại lớp

Mô tả các thuật toán (sơ đồ khối/mã giả) sau

1. Nhập dãy số nguyên cho tới khi gặp một số âm và đưa ra tổng của các số chia hết cho 5
2. Nhập dãy số nguyên cho tới khi gặp một số chia hết cho 5 và đưa ra trung bình cộng của các số dương trong dãy đã nhập
3. Nhập dãy số cho tới khi gặp số âm và đưa ra số lượng các số nằm trong đoạn [11..22]
4. Nhập dãy số cho tới khi tổng của dãy lớn hơn 1001 hoặc số phần tử đã nhập là 100

96

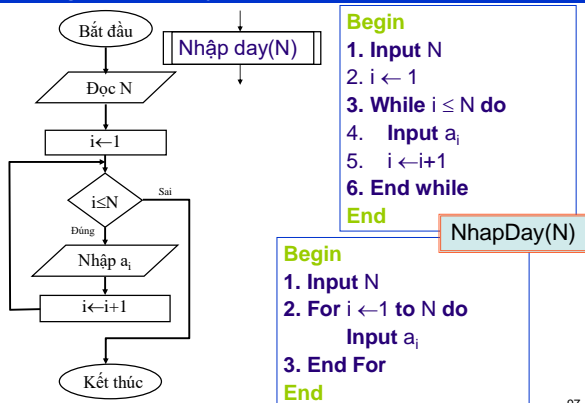
95

96



Chương 2: Thuật toán  
5. Một số thuật toán thông dụng

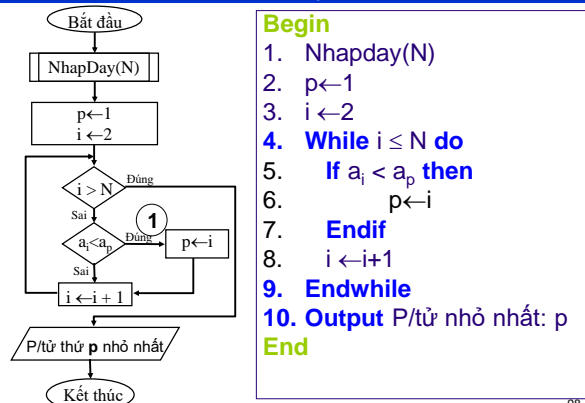
### Nhập giá trị cho dãy N số



97

Chương 2: Thuật toán  
5. Một số thuật toán thông dụng

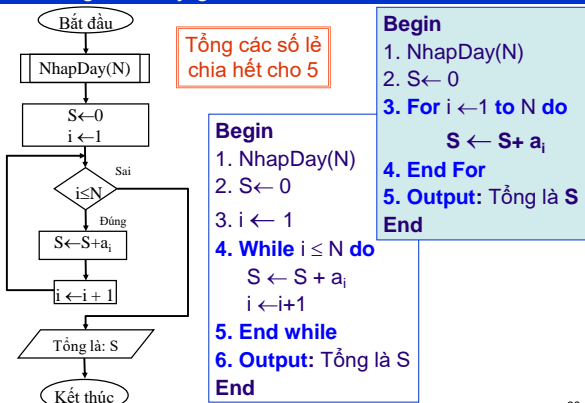
### Tìm phần tử nhỏ nhất cho dãy N số



98

Chương 2: Thuật toán  
5. Một số thuật toán thông dụng

### Tìm tổng của dãy gồm N số



99

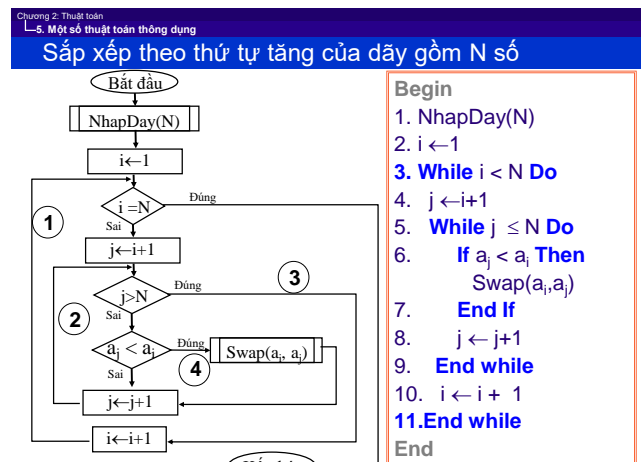
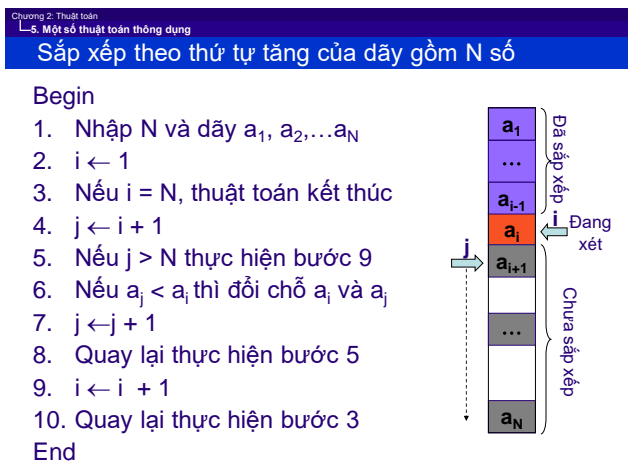
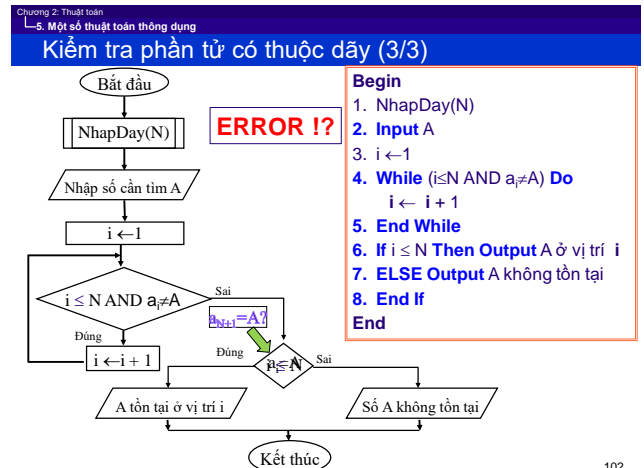
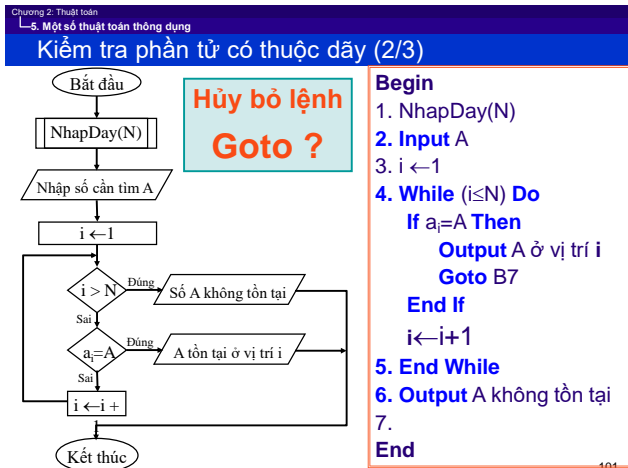
Chương 2: Thuật toán  
5. Một số thuật toán thông dụng

### Kiểm tra phần tử có thuộc dãy (1/3)

#### Begin

1. Nhập N, dãy số  $a_1, a_2, \dots, a_N$  và số cần tìm A
  2.  $i \leftarrow 1$
  3. Nếu  $i > N$  sang bước 7
  4. Nếu  $A = a_i$  sang bước 8
  5.  $i \leftarrow i + 1$
  6. Quay về bước 3
  7. Thông báo: A không thuộc dãy và kết thúc
  8. Thông báo A thuộc dãy và kết thúc
- End**

100



### Bài tập về nhà

Mô tả các thuật toán (sơ đồ khối/mã giả) sau

- Đếm số chẵn của một dãy N số nguyên
- Đưa ra trung bình cộng các số dương của một dãy gồm N số
- Sắp xếp lại dãy N số thực theo nguyên tắc:
  - Các số 0 ở đầu dãy, sau đó là các số âm rồi đến các số dương
- Đếm số hạng đạt giá trị bằng giá trị lớn nhất
- ....