



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# TIN HỌC ĐẠI CƯƠNG

## Phần 1: TIN HỌC CĂN BẢN

@it-hut.edu.vn

# Nội dung môn học

## Phần 1: Tin học căn bản

### 1. Chương 1: Thông tin và biểu diễn thông tin

- Các khái niệm cơ bản về thông tin và tin học
- Biểu diễn dữ liệu trong máy tính

### 2. Chương 2: Hệ thống máy tính

- Hệ thống máy tính
- Phần mềm máy tính
- Hệ điều hành
- Mạng máy tính

### 3. Chương 3 : Thuật toán

- Giải quyết bài toán bằng máy tính
- Khái niệm thuật toán
- Biểu diễn thuật toán

# Chương 3 Thuật toán

## 1. Giải quyết bài toán bằng máy tính

1. Khái niệm về bài toán
2. Giải quyết bài toán bằng máy tính
3. Phân loại bài toán

## 2. Khái niệm về thuật toán

1. Khái niệm và định nghĩa
2. Đặc trưng của thuật toán

## 3. Biểu diễn thuật toán

1. Tại sao phải biểu diễn thuật toán ?
2. Các phương pháp biểu diễn thuật toán

## 4. Một số thuật toán thông dụng

1. Thuật toán đệ quy
2. Thuật toán heuristic
3. Thuật toán thông dụng

# Giải quyết bài toán bằng máy tính

1. Khái niệm về bài toán

2. Giải quyết bài toán bằng máy tính

3. Phân loại bài toán

# Khái niệm về bài toán

## **Problem - Bài toán hay vấn đề?**

(Socrate 470-399 TCN )

- Vấn đề có 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, kinh tế

# Khái niệm về bài toán

## Hai loại vấn đề

(Pythagoras 582-507 TCN )

- **Theorema:** vấn đề cần khẳng định đúng/sai.
  - 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 đầu.
  - Bài toán dựng hình, tổng hợp chất hóa học, tìm đường đi ngắn nhất....

# Khái niệm về bài toán

## Biểu diễn bài toán

$$A \rightarrow B$$

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

# Khái niệm về bài toán

## Biểu diễn bài toán

$$A \rightarrow B$$

- 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 ra tập các thao tác cơ bản được dùng trong suy luận và hành động



# Khái niệm về bài toán

## Trong tin học

$$A \rightarrow B$$

- A: Input
- B: Output
- $\rightarrow$ : Chương trình cho phép biến đổi A thành B
  - Tạo thành từ các lệnh cơ bản của máy tính.

# Khái niệm về bài toán

- Chương trình
  - Cách mã hóa lại thuật toán/thuật giải để giải quyết vấn đề/bài toán đã cho
- Khó khăn
  - Tính không xác định của vấn đề/bài toán
    - A và B không đầy đủ, rõ ràng
- Giải quyết một vấn đề trên máy tính
  - Thiết kế thuật giải vẫn chủ yếu được thực hiện bởi con người
  - Tự động hóa xây dựng thuật toán/thuật giải ?

# Giải quyết bài toán bằng máy tính

1. Khái niệm về bài toán

2. Giải quyết bài toán bằng máy tính

3. Phân loại bài toán

# Giải quyết bài toán bằng máy tính

- 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 (chương trình). Máy tính sẽ thực hiện các chỉ dẫn đó.

# Giải quyết bài toán bằng máy tính

## Gồm nhiều giai đoạn

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

# Giải quyết bài toán bằng máy tính

## Xác định bài toán

- Làm rõ yêu cầu của người sử dụng,
- Đánh giá, nhận định tính khả thi của bài toán

## Ví dụ:

### Bài toán sắp xếp dãy số

- Yêu cầu: Nhập dãy, đưa ra dãy đã sắp xếp theo thứ tự

# Giải quyết bài toán bằng máy tính

## Lựa chọn phương pháp giải

- Tồn tại nhiều phương pháp
- Khác nhau về thời gian thực hiện, chi phí lưu trữ dữ liệu, độ chính xác...
- Cần tùy theo nhu cầu cụ thể và khả năng xử lý tự động mà ta sẽ sử dụng để lựa chọn phương pháp thích hợp
- **Ví dụ:** Bài toán sắp xếp dãy số
  - Nổi bọt, Vun đống, Sắp xếp nhanh

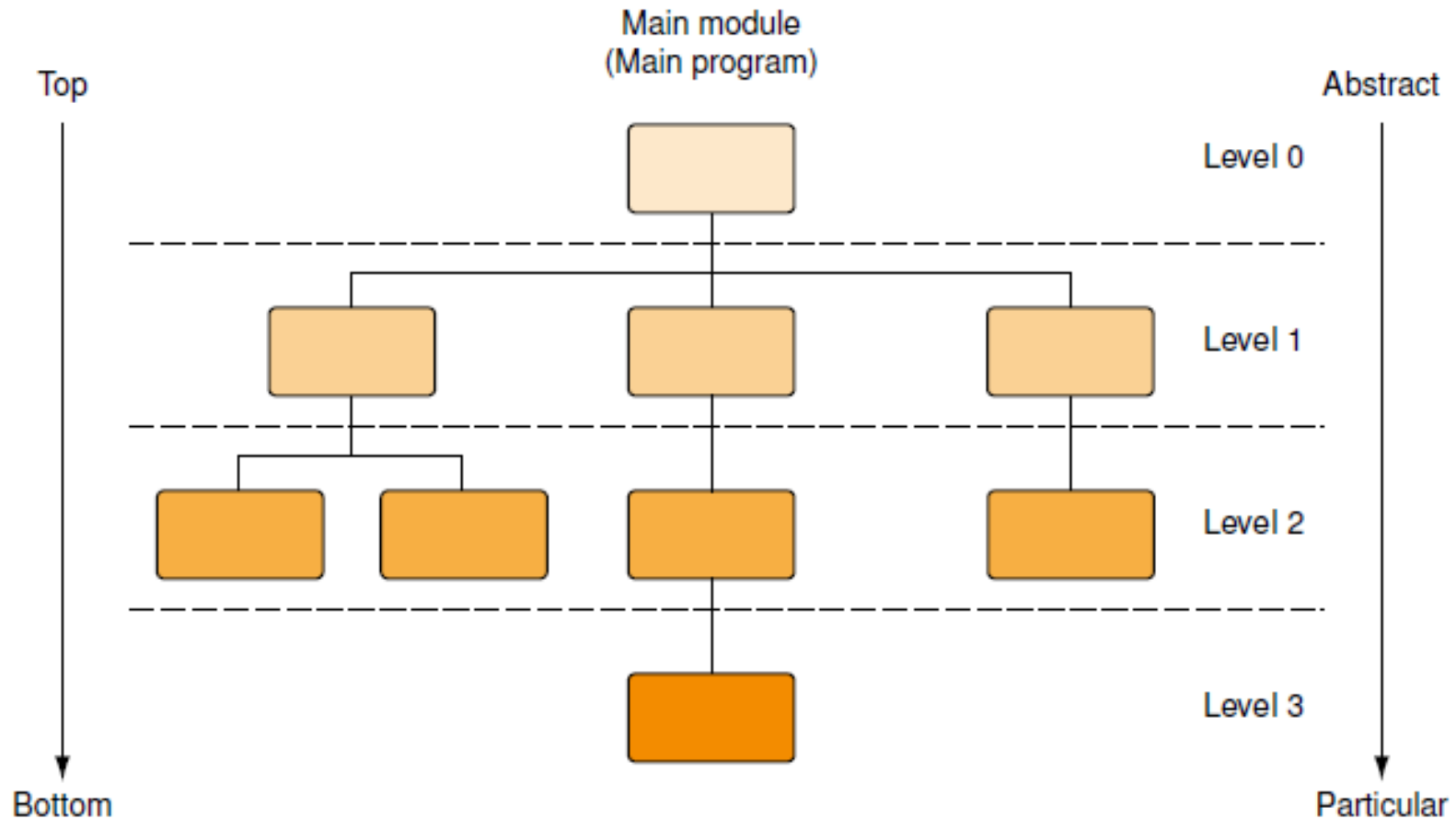
# Giải quyết bài toán bằng máy tính

## Xây dựng thuật toán hoặc thuật giải

- Xây dựng mô hình chặt chẽ, chính xác hơn và chi tiết hóa hơn phương pháp đã lựa chọn,
- Xác định rõ ràng dữ liệu vào, ra cho các bước thực hiện và trật tự thực hiện.
- Nên áp dụng phương pháp thiết kế có cấu trúc, từ thiết kế tổng thể tiến hành làm mịn từng bước.
  - **Ví dụ: phương pháp thiết kế top-down**



# Giải quyết bài toán bằng máy tính



# Giải quyết bài toán bằng máy tính

## Cài đặt chương trình

- Lựa chọn ngôn ngữ lập trình
  - NNLT bậc thấp
  - NNLT bậc cao C, Pascal, Java,...
- Mô tả giải thuật bằng chương trình

# Giải quyết bài toán bằng máy tính

## Hiệu chỉnh chương trình

- Chạy thử để phát hiện và điều chỉnh các sai sót có thể có ở bước 4.
  - Lỗi cú pháp
  - Lỗi ngữ nghĩa.

# Giải quyết bài toán bằng máy tính

## 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.
    - Nếu không kiểm tra lại toàn bộ các bước.

# Giải quyết bài toán bằng máy tính

## Giải quyết bài toán

### 1. Giai đoạn quan niệm :

- Gồm các bước phân tích, lựa chọn mô hình, xây dựng thuật giải, cài đặt chương trình

### 2. Giai đoạn khai thác và bảo trì

- Đáp ứng nhu cầu về cải tiến, mở rộng chương trình do các yếu tố của bài toán ban đầu có thể thay đổi.

# Giải quyết bài toán bằng máy tính

## Các phương pháp

- Xác định trực tiếp lời giải
- Tìm kiếm lời giải

# Giải quyết bài toán bằng máy tính

## Hướng xác định trực tiếp lời giải

- Thường sử dụng trong quá trình học tập.
  - Tìm nghiệm phương trình bậc 2 theo định lý Viet.
- Xác định trực tiếp được lời giải qua
  - Các thủ tục tính toán (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.
- 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. Điều này hạn chế khi tính toán thủ công nhưng là thế mạnh của máy tính .

# Giải quyết bài toán bằng máy tính

## Hướng tìm kiếm lời giải

- Là cách tiếp cận chủ yếu, dựa theo nguyên lý “**thử và sai**”.
- Một loạt các phương pháp tìm kiếm lời giải theo nguyên lý “thử-sai” được ứng dụng rộng rãi để giải quyết hiệu quả nhiều vấn đề và bài toán phức tạp.
- 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 các khả năng. Khả năng thành công tùy theo chiến lược chọn ngẫu nhiên và một số điều kiện cụ thể của bài toán.
- Phương pháp quay lui:
  - Đánh dấu các thử nghiệm thất bại và thử khả năng mới (quay lui tìm đường khác).
- **Ví dụ:** bài toán 8 quân hậu...



# Giải quyết bài toán bằng máy tính

1. Khái niệm về bài toán

2. Giải quyết bài toán bằng máy tính

3. Phân loại bài toán

# Phân loại bài toán

**Độ phức tạp thuật toán:** Khối lượng thời gian và không gian cần thiết để thực hiện thuật toán

- Độ phức tạp không gian :
  - Phụ thuộc vào cấu trúc dữ liệu được sử dụng trong cài đặt thuật toán
- Độ phức tạp thời gian
  - Phụ thuộc nhiều yếu tố, khó khăn
  - Mang lại thông tin về độ phức tạp của thuật toán

# Phân loại bài toán

## Độ phức tạp thời gian

- Số lượng các thao tác cơ bản được sử dụng để giải quyết bài toán
  - So sánh 2 số nguyên,
  - Cộng, nhân, chia hai số nguyên
  - Thao tác cơ bản khác :gán,..
- Số lượng các thao tác cơ bản phụ thuộc vào kích thước dữ liệu vào

# Phân loại bài toán

Bài toán có dữ liệu nhập kích thước  $n$

- Thời gian thực hiện thuật toán nhỏ nhất
  - Thời gian thực hiện thuật toán trong trường hợp tốt nhất
- Thời gian thực hiện thuật toán lớn nhất
  - Thời gian thực hiện thuật toán trong trường hợp xấu nhất.
- Thời gian trung bình để thực hiện thuật toán

# Phân loại bài toán

## Thuật ngữ thông dụng :

- Độ phức tạp hằng:  $O(1)$
- Độ phức tạp logarith:  $O(\log n)$
- Độ phức tạp tuyến tính:  $O(n)$
- Độ phức tạp  $n \log n$ :  $O(n \log(n))$
- Độ phức tạp đa thức:  $O(n^b)$
- Độ phức tạp lũy thừa:  $O(b^n)$ , trong đó  $b > 1$
- Độ phức tạp giai thừa:  $O(n!)$

# Phân loại bài toán

## Phân loại bài toán

1. Bài toán đa thức
2. Bài toán không đa thức
3. Những bài toán NP (NP problems)

# Phân loại bài toán

## Bài toán đa thức

- Gồm những giải thuật có độ phức tạp bị chặn bởi một đa thức. Những giải thuật ở lớp này chạy trong thời gian đa thức.
  - Ví dụ: thuật toán sắp xếp chèn, Quicksort
- Giải thuật không ở lớp bài toán đa thức có thời gian thực hiện cực lâu khi giá trị đưa vào càng tăng.
  - Ví dụ: một giải thuật có độ phức tạp  $2^n$  gần như không chạy được vì thời gian quá lâu khi dữ liệu vào lớn.
- Một bài toán có thể giải trên lý thuyết được hay không phụ thuộc vào việc có thuộc lớp đa thức hay không.

# Phân loại bài toán

## Bài toán không đa thức

Thời gian giải bài toán không phải đa thức

- Ví dụ Lập danh sách tất cả nhóm người có số lượng là một hay nhiều hơn có thể hình thành từ nhóm  $n$  người
  - Có  $2^n - 1$  nhóm như vậy; mọi giải thuật để giải bài toán phải có ít nhất  $2^{n-1}$  bước
  - Độ phức tạp ít nhất phải cỡ  $2^{n-1}$ .
- Ví dụ: bài toán tháp Hà nội



# Phân loại bài toán

## Những bài toán NP(NP problems)

- Thuật toán/giải thuật xác định
  - Tại mỗi bước là duy nhất, xác định rõ ràng
- Thuật toán/giải thuật không xác định
  - Tại mỗi bước là một lựa chọn từ nhiều cách
- Độ phức tạp thời gian của bài toán không xác định là thời gian cần thiết để thực hiện giải bài toán đó nếu **chọn đúng tại mỗi điểm của sự lựa chọn**
- Bài toán NP: Bài toán có thể giải được trong thời gian đa thức bằng một giải thuật không xác định
  - Phân loại bài toán NP hiện vẫn là một vấn đề mở trong ngành máy tính

# Chương 3 Thuật toán

## 1. Giải quyết bài toán bằng máy tính

1. Khái niệm về bài toán
2. Giải quyết bài toán bằng máy tính
3. Phân loại bài toán

## 2. Khái niệm về thuật toán

1. Khái niệm và định nghĩa
2. Đặc trưng của thuật toán

## 3. Biểu diễn thuật toán

1. Tại sao phải biểu diễn thuật toán ?
2. Các phương pháp biểu diễn thuật toán

## 4. Một số thuật toán thông dụng

1. Thuật toán đệ quy
2. Thuật toán heuristic
3. Thuật toán thông dụng

# Khái niệm về thuật toán

1. Khái niệm và định nghĩa

2. Các đặc trưng của thuật toán

# Khái niệm và định nghĩa

- Thuật toán:

Khái niệm cơ sở của Toán học và Tin học

- Nghiên cứu thuật toán:

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?**

# Khái niệm và định nghĩa

- 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, dùng để hướng dẫn thực hiện hành động nhằm đạt được mục tiêu đề ra
- Về mặt phương pháp: thuật toán là sự thể hiện của một phương pháp để giải quyết một vấn đề

# Khái niệm và định nghĩa

*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*

# Khái niệm và định nghĩa

**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 nào 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.

# Khái niệm về thuật toán

1. Khái niệm và định nghĩa

2. Các đặc trưng của thuật toán



# Các đặc trưng của thuật toán

## Các đặc trưng của thuật toán

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

# Các đặc trưng của thuật toán

- **Nhập (input):**

Các thuật toán thường có các giá trị “*đầu vào*” (input values) từ một tập hợp nhất định nào đó.

- **Xuất (output):**

Từ mỗi tập hợp các giá trị nhập, thuật toán thường tạo ra những giá trị xuất (output values) thuộc một tập hợp nhất định nào đó thể hiện lời giải cho bài toán (hay vấn đề)

# Các đặc trưng của thuật toán

- Tính xác định (definiteness):  
Các bước trong thuật toán phải chính xác rõ ràng.
- Tính hữu hạn (finiteness):  
Thuật giải phải cho ra lời giải (hay kết quả) sau một số hữu hạn các bước

# Các đặc trưng của thuật toán

- Tính hiệu quả:

Được đánh giá dựa trên một số tiêu chuẩn như khối lượng tính toán, không gian sử dụng và thời gian thực hiện (khi thực hiện thuật toán trên máy tính).

- Tính tổng quát:

Áp dụng được cho tất cả các bài toán có dạng như mong muốn chứ không phải chỉ áp dụng được cho một số trường hợp riêng lẻ nào đó

# Chương 3 Thuật toán

## 1. Giải quyết bài toán bằng máy tính

1. Khái niệm về bài toán
2. Giải quyết bài toán bằng máy tính
3. Phân loại bài toán

## 2. Khái niệm về thuật toán

1. Khái niệm và định nghĩa
2. Đặc trưng của thuật toán

## 3. Biểu diễn thuật toán

1. Tại sao phải biểu diễn thuật toán ?
2. Các phương pháp biểu diễn thuật toán

## 4. Một số thuật toán thông dụng

1. Thuật toán đệ quy
2. Thuật toán heuristic
3. Thuật toán thông dụng

# Biểu diễn thuật toán

1. Tại sao phải biểu diễn thuật toán ?

2. Các phương pháp biểu diễn thuật toán

# Tại sao phải biểu diễn thuật toán

- Tại sao:
  - Truyền đạt thuật toán cho người khác
  - “*Truyền đạt*” thuật toán cho máy tính

# Biểu diễn thuật toán

1. Tại sao phải biểu diễn thuật toán ?

2. Phương pháp biểu diễn thuật toán



# Phương pháp biểu diễn thuật toán

- Phương pháp:
  - Ngôn ngữ tự nhiên
  - Ngôn ngữ lưu đồ(sơ đồ khối)
  - Ngôn ngữ tựa ngôn ngữ lập trình (mã giả)
  - Ngôn ngữ lập trình

# Phương pháp biểu diễn thuật toán

## Ngôn ngữ tự nhiên:

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

# Phương pháp biểu diễn thuật toán

Ví dụ: Tìm giá trị lớn nhất của một dãy số nguyên gồm N số

- Đầu vào:
  - Số nguyên dương N
  - N số nguyên  $a_1, a_2, \dots, a_N$
- Đầu ra:

Số nguyên lớn nhất của dãy
- Ý tưởng
  - Khởi tạo giá trị  $\text{Max} = a_1$
  - Lần lượt so sánh Max với  $a_i$  ( $i=2, 3, \dots, N$ )
    - Nếu  $a_i > \text{Max}$ ; gán giá trị mới cho Max là  $a_i$ .

# Phương pháp biểu diễn thuật toán

## ***Thuật toán***

- B1: Nhập  $N$  và dãy số  $a_i$ .
- B2: Gán giá trị  $a_1$  cho  $Max$ ,  $i \leftarrow 2$ .
- B3: Nếu  $i > N$ , thực hiện bước 7
- B4: Nếu  $a_i > Max$ , gán giá trị  $a_i$  cho  $Max$ .
- B5: Tăng  $i$  lên 1 đơn vị.
- B6: Quay lên B3.
- B7: Thông báo:  $Max$  là giá trị lớn nhất của dãy
- B8: Kết thúc.

# Phương pháp biểu diễn thuật toán

Ngôn ngữ lưu đồ (sơ đồ khối) :

- Công cụ trực quan để diễn đạt các thuật toán
- Đưa ra một cái nhìn tổng quan về toàn cảnh của 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

# Phương pháp biểu diễn thuật toán

## **Nút giới hạn** (nút đầu và nút cuối )

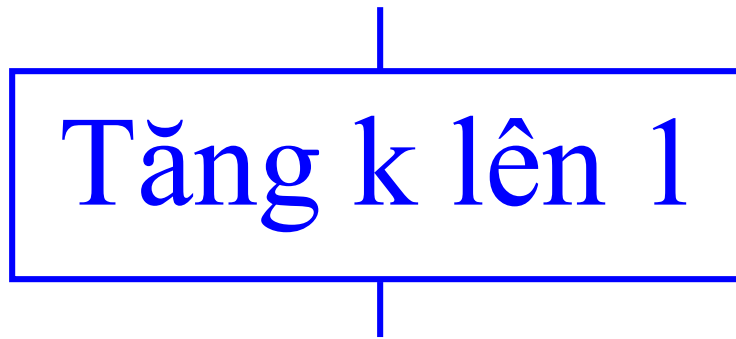
- Được biểu diễn bởi hình ôvan có ghi chữ bên trong



# Phương pháp biểu diễn thuật toán

## Nút thao tác

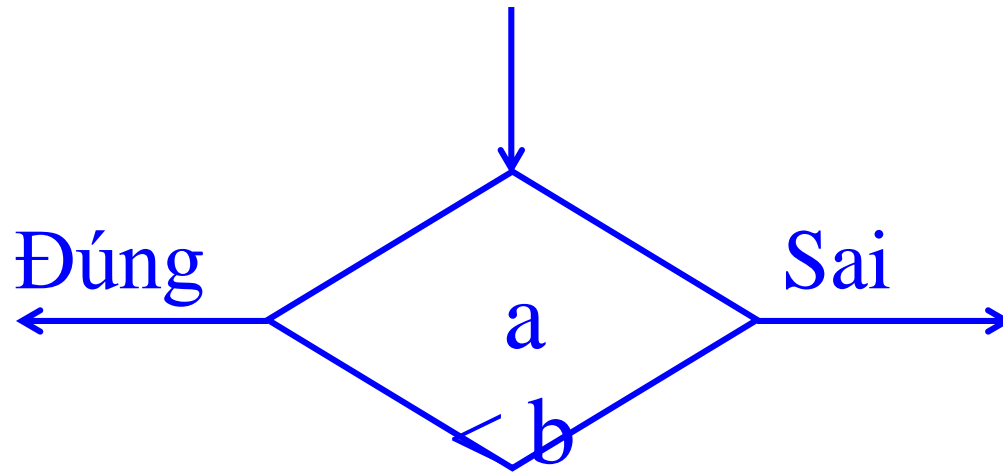
- Là một hình chữ nhật có ghi các lệnh cần thực hiện



# Phương pháp biểu diễn thuật toán

## Nút điều kiện

- Là một hình thoi có ghi điều kiện cần kiểm tra.



- Các cung nối với nút này:
  - 2 cung ra chỉ hướng đi theo 2 trường hợp: điều kiện đúng và điều kiện sai



# Phương pháp biểu diễn thuật toán

## Cung

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

# Phương pháp biểu diễn thuật toán

## Hoạt động

- Bắt đầu từ nút đầu tiên.
- Sau khi thực hiện các thao tác hoặc kiểm tra điều kiện ở mỗi nút, bộ xử lý sẽ theo một cung đến một nút khác.
- Thuật toán sẽ dừng khi gặp nút kết thúc hay nút cuối.
- **Ví dụ:**  
Tìm giá trị lớn nhất của một dãy số nguyên gồm  $N$  số

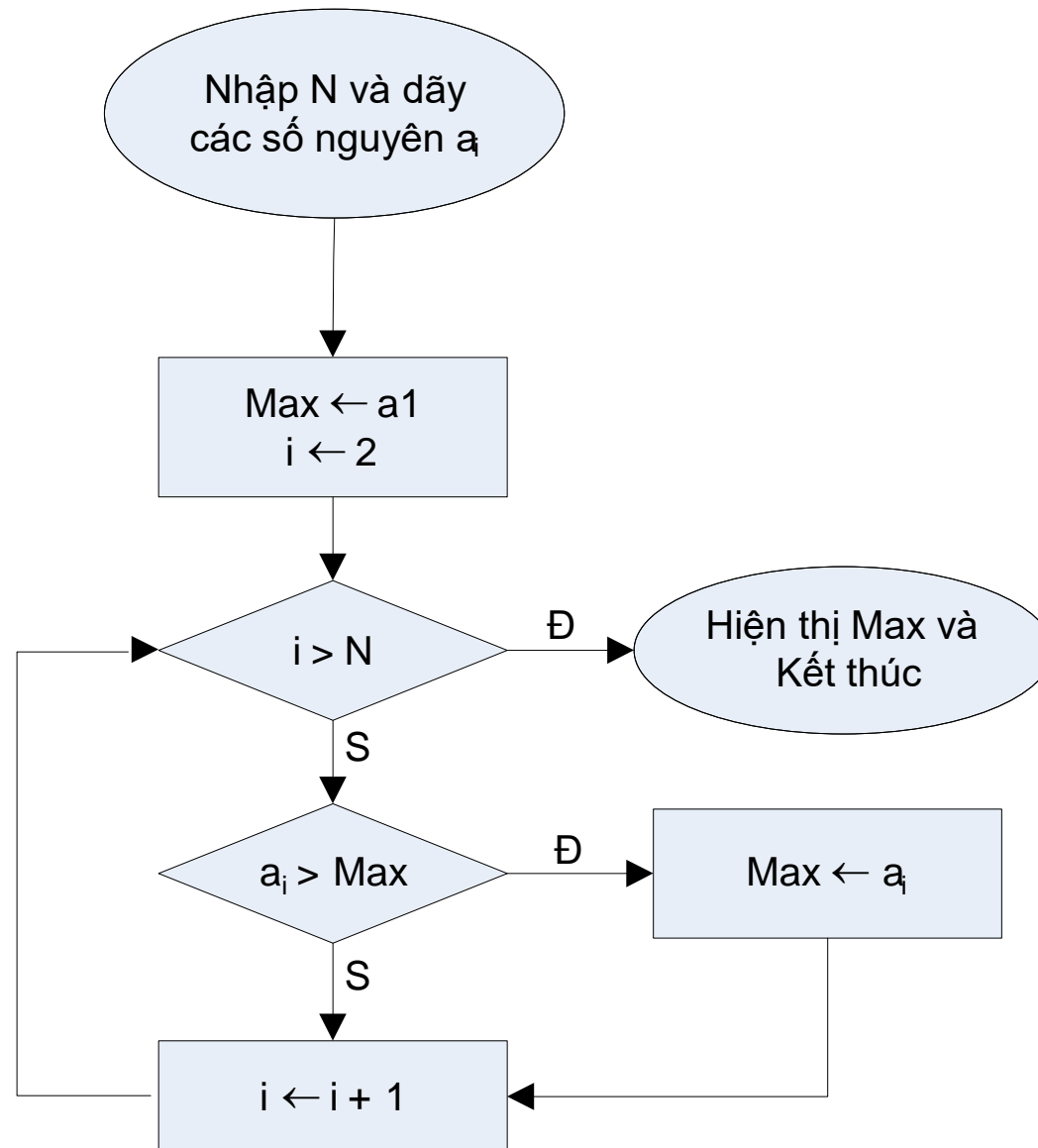
# Phương pháp biểu diễn thuật toán

## ***Biểu diễn bằng lời***

- B1: Nhập  $N$  và dãy số  $a_i$ .
- B2: Gán giá trị  $a_1$  cho  $Max$ ,  $i \leftarrow 2$ .
- B3: Nếu  $i > N$ , thực hiện bước 7
- B4: Nếu  $a_i > Max$ , gán giá trị  $a_i$  cho  $Max$ .
- B5: Tăng  $i$  lên 1 đơn vị.
- B6: Quay lên B3.
- B7: Thông báo:  $Max$  là giá trị lớn nhất của dãy
- B8: Kết thúc.

# Phương pháp biểu diễn thuật toán

## Bảng lưu đồ



# Phương pháp biểu diễn thuật toán

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

- 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
  - Cấu trúc kiểu thủ tục
    - Thuật toán phức tạp cần được trình bày thành nhiều cấp độ
- Tiện lợi, đơn giản, và dễ hiểu.

# Phương pháp biểu diễn thuật toán

## Cấu trúc thường gặp

- Phát biểu gán
- Lựa chọn
- Lặp

# Phương pháp biểu diễn thuật toán

- Phát biểu gán
    - Đặt giá trị cho một biến
- Max := a1
- Max ← a1

# Phương pháp biểu diễn thuật toán

Lựa chọn

**if**(điều kiện) **then** (hành động)

Hoặc là

**if**(điều kiện) **then** (hành động)  
**else** (hành động)



# Phương pháp biểu diễn thuật toán

## Cấu trúc lặp

**while** (điều kiện) **do** (hành động)

hoặc

**repeat**

(hành động)

**until** (điều kiện)

hoặc

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

hoặc

**for** biến ← giá trị đầu **downto** giá trị cuối **do** hành động

# Phương pháp biểu diễn thuật toán

## Ngôn ngữ lập trình :

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

# Chương 3 Thuật toán

## 1. Giải quyết bài toán bằng máy tính

1. Khái niệm về bài toán
2. Giải quyết bài toán bằng máy tính
3. Phân loại bài toán

## 2. Khái niệm về thuật toán

1. Khái niệm và định nghĩa
2. Đặc trưng của thuật toán

## 3. Biểu diễn thuật toán

1. Tại sao phải biểu diễn thuật toán ?
2. Các phương pháp biểu diễn thuật toán

## 4. Một số thuật toán thông dụng

1. Thuật toán đệ quy
2. Thuật toán heuristic
3. Một số thuật toán thông dụng

# Một số thuật toán thông dụng

1. Thuật toán đệ quy

2. Thuật toán heuristic

3. Một số thuật toán thông dụng

# 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, Trong một bước của thuật toán có thể hiện lại thuật toán đó (với dữ liệu vào có độ lớn thấp hơn)  $\Rightarrow$  “thuật toán đệ qui”.

# Thuật toán đệ quy

## Tính giai thừa của một số tự nhiên:

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

## Thuật giải:

- $F:=1;$
- If  $n>0$  then  
     $F:=F(n-1)*n;$
- Output  $F$

# Thuật toán đệ quy

## Lưu ý

- Thuật toán đệ qui gồm 2 phần
  - **Phần cơ sở:** không cần thực hiện lại thuật toán (hay không có yêu cầu gọi đệ qui).
  - **Phần đệ qui:** có yêu cầu gọi đệ qui (yêu cầu thực hiện lại thuật toán); Yêu cầu gọi đệ qui được đặt trong một điều kiện kiểm tra việc gọi đệ qui.
- Đệ qui dễ gây ra tình trạng tràn STACK.
  - Nếu có thể, nên viết lại thuật toán đệ qui dưới dạng lặp.
    - $P \leftarrow 1$
    - For  $k \leftarrow 1$  To  $N$  Do  $P \leftarrow P * k$
    - Print  $P$

# Một số thuật toán thông dụng

1. Thuật toán đệ quy

2. Thuật toán heuristic

3. Một số thuật toán thông dụng



# 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 có những đổi mới cho khái niệm thuật toán**

# Thuật giải heuristic

## Tiêu chuẩn của thuật toán được mở rộng:

- Tính xác định
  - Các thuật giải đệ qui, các giải thuật ngẫu nhiên
- Tính đúng đắn
  - Không còn bắt buộc với một số cách giải cho các bài toán nhất là các cách giải gần đúng.
  - Trong thực tiễn có nhiều trường hợp, chấp nhận các cách giải chỉ 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

# 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.

# Thuật giải heuristic

## Nguyên lý

- Vết cạn 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.
- 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.
- 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..

# Một số thuật toán thông dụng

1. Thuật toán đệ quy

2. Thuật toán heuristic

3. Một số thuật toán thông dụng

# Một số thuật toán thông dụng

## 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
- Tổng các phần tử
- Max, Min
- Sắp xếp
- Tìm phần tử; Đếm phần tử
- Chèn phần tử
- Xóa phần tử