

# TOÁN RỜI RẠC

## CHƯƠNG I : KHÁI NIỆM CƠ BẢN Ma trận và giải thuật

Lecturer: PhD. Ngo Huu Phuc

Tel: 0438 326 077

Mob: 098 5696 580

Email: [ngohuuphuc76@gmail.com](mailto:ngohuuphuc76@gmail.com)

# NỘI DUNG

## I. Ma trận.

1. Khái niệm.
2. Các phép toán trên ma trận.

## II. *Thuật toán và biểu diễn thuật toán.*

1. Khái niệm.
2. Đặc tính cơ bản của thuật toán.
3. Biểu diễn thuật toán.

## III. Bài tập

# 1. Ma trận – Khái niệm

- Ma trận là một bảng số hình chữ nhật, có kích thước  $m \times n$ .

Cho ma trận

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ & & \dots & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

Hàng thứ  $i$  của ma trận là ma trận  $1 \times n$   
 $(a_{i1}, a_{i2}, \dots, a_{in})$

Cột thứ  $j$  của ma trận  $A$  là ma trận  $n \times 1$

$$\begin{pmatrix} a_{1j} \\ a_{2j} \\ \dots \\ a_{nj} \end{pmatrix}$$

Đơn giản, có thể viết ma trận như sau  **$A = [a_{ij}]$**

## 2. Ma trận - Các phép toán trên ma trận (1/3)

### a. Phép cộng

- Cho  $A = [a_{ij}]$  và  $B = [b_{ij}]$  là các ma trận  $m \times n$ .
- Tổng của  $A$  và  $B$  được ký hiệu là  $A + B$  là ma trận  $m \times n$  có phần tử thứ  $(i,j)$  là  $a_{ij} + b_{ij}$ .
- Nói cách khác  $A + B = [a_{ij} + b_{ij}]$ .

### b. Phép nhân

- Cho  $A = [a_{ij}]$  là ma trận  $m \times k$  và  $B = [b_{ij}]$  là ma trận  $k \times n$ .
- Tích của  $A$  và  $B$ , được ký hiệu là  $AB$ , là ma trận  $m \times n$  với phần tử  $(i, j)$  bằng tổng các tích của các phần tử tương ứng từ hàng thứ  $i$  của  $A$  và cột thứ  $j$  của  $B$ .
- Nói cách khác, nếu  $AB = [c_{ij}]$  thì

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{ik}b_{kj} = \sum_{t=1}^k a_{it}b_{tj}$$

## 2. Ma trận - Các phép toán trên ma trận (2/3)

### *c. Chuyển vị và lũy thừa các ma trận*

- Ma trận vuông  $n \times n$   $I_n = [\delta_{ij}]$  có các phần tử trên đường chéo chính  $\delta_{ij} = 1$  gọi là ma trận đơn vị.
- Cho ma trận  $A = [a_{ij}]$  có kích thước  $m \times n$ , chuyển vị của  $A$  ký hiệu là  $A^T$  là ma trận  $n \times m$  nhận được bằng cách trao đổi các hàng và cột của  $A$  cho nhau.
- Nói cách khác, nếu  $A^T = [b_{ij}]$ , thì  $b_{ij} = a_{ji}$ .

## 2. Ma trận - Các phép toán trên ma trận (3/3)

### **Một số ví dụ**

Ví dụ: Cho ma trận  $A = \begin{bmatrix} 1 & 2 & 3 \\ a & b & c \end{bmatrix}$  chuyển vị của  $A$  là  $A^T = \begin{bmatrix} 1 & a \\ 2 & b \\ 3 & c \end{bmatrix}$

Ma trận vuông  $A$  được gọi là đối xứng nếu  $A^T = A$ .

Ví dụ: Ma trận  $A = \begin{bmatrix} a & b & c & d \\ b & 1 & 2 & 3 \\ c & 2 & 2 & e \\ d & 2 & e & 3 \end{bmatrix}$  là ma trận đối xứng

### 3. Thuật toán và biểu diễn thuật toán (1/8)

#### a. Khái niệm

- Thuật toán là một phương pháp **giải quyết bài toán, vấn đề** bằng cách **mô tả từng bước** thực hiện để sau một số **hữu hạn** bước sẽ đi đến kết quả.
- Với thuật toán, có phương pháp chỉ dẫn cho người hoặc máy thực hiện việc giải quyết vấn đề cụ thể, theo đó không phải "tư duy" gì thêm vẫn đưa ra kết quả mong muốn.

### 3. Thuật toán và biểu diễn thuật toán (2/8)

#### **b. Đặc tính cơ bản của thuật toán**

- **Tính đúng đắn.**

- Thuật toán được xây dựng cho một bài toán, một vấn đề nào đó phải bảo đảm sau một số hữu hạn bước thực hiện phải đi đến kết quả đúng.

- **Tính tuần tự.**

- Thuật toán được xây dựng trong đó phải mô tả tuần tự thứ tự thực hiện các bước cụ thể, bảo đảm khi thực hiện không đi vào ngõ cụt, không gặp trở ngại nào.

- **Tính phổ biến.**

- Thuật toán được xây dựng thường nhằm giải quyết một lớp bài toán hoặc vấn đề nào đó.

- **Tính tối ưu.**

- Khi xây dựng thuật toán cần phải lưu ý bảo đảm điều kiện tốt nhất cho việc thực hiện, điều này có nghĩa là trong từng bước hoặc tổng thể cần lựa chọn trong các phương án tốt nhất có thể được.



### 3. Thuật toán và biểu diễn thuật toán (3/8)

#### ***c. Biểu diễn thuật toán.***

- *Biểu diễn bằng ngôn ngữ tự nhiên.*
- *Biểu diễn sơ đồ, lưu đồ khối.*
- *Biểu diễn giả lệnh ngôn ngữ lập trình.*

### 3. Thuật toán và biểu diễn thuật toán (4/8)

#### ***c.1. Biểu diễn bằng ngôn ngữ tự nhiên.***

- Phương pháp này dùng ngôn ngữ tự nhiên để diễn tả các bước cần thực hiện của thuật toán.
- Phương pháp biểu diễn ngôn ngữ có ưu, nhược điểm:
  - gần gũi dễ hiểu đối người thực hiện,
  - trong nhiều trường hợp không chặt chẽ và đa nghĩa vì bản chất của ngôn ngữ tự nhiên là đa nghĩa.
  - không có tính thống nhất giữa các ngôn ngữ khác nhau.

### 3. Thuật toán và biểu diễn thuật toán (5/8)

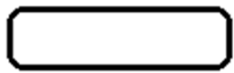
#### *c.1. Biểu diễn bằng ngôn ngữ tự nhiên.*

- Ví dụ: Biểu diễn thuật toán giải phương trình bậc 2:  $ax^2 + bx + c = 0$  với  $a, b, c$  là các số thực và  $a \neq 0$ .
  - Bước 1. Đưa vào (Input)  $a, b, c$
  - Bước 2. Tính biệt thức  $\Delta = b^2 - 4ac$
  - Bước 3. Xét dấu  $\Delta$ :
    - Nếu  $\Delta \geq 0$  chuyển sang bước 4
    - Ngược lại chuyển sang bước 4
  - Bước 4. Tính nghiệm
$$x_1 = \frac{-b + \sqrt{\Delta}}{2a} \quad ; \quad x_2 = \frac{-b - \sqrt{\Delta}}{2a}$$
đưa ra thông báo nghiệm của phương trình là  $x_1$  và  $x_2$ . Sang bước 6.
  - Bước 5. (Output) Đưa ra thông báo phương trình vô nghiệm.
  - Bước 6. Kết thúc.

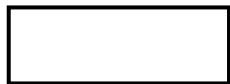
### 3. Thuật toán và biểu diễn thuật toán (5/8)

#### *c.2. Biểu diễn sơ đồ, lưu đồ khối.*

- Thuật toán có thể biểu diễn bằng sơ đồ khối.



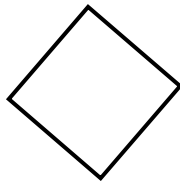
Chỉ sự bắt đầu hoặc kết thúc của thuật toán



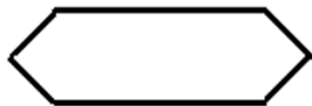
Mô tả một phép toán, thao tác cần thực hiện



Mô tả dữ liệu vào (Input), ra (Output)



Mô tả điều kiện hoặc một biểu thức logic cần kiểm tra



Mô tả lựa chọn một trong các khả năng xảy ra

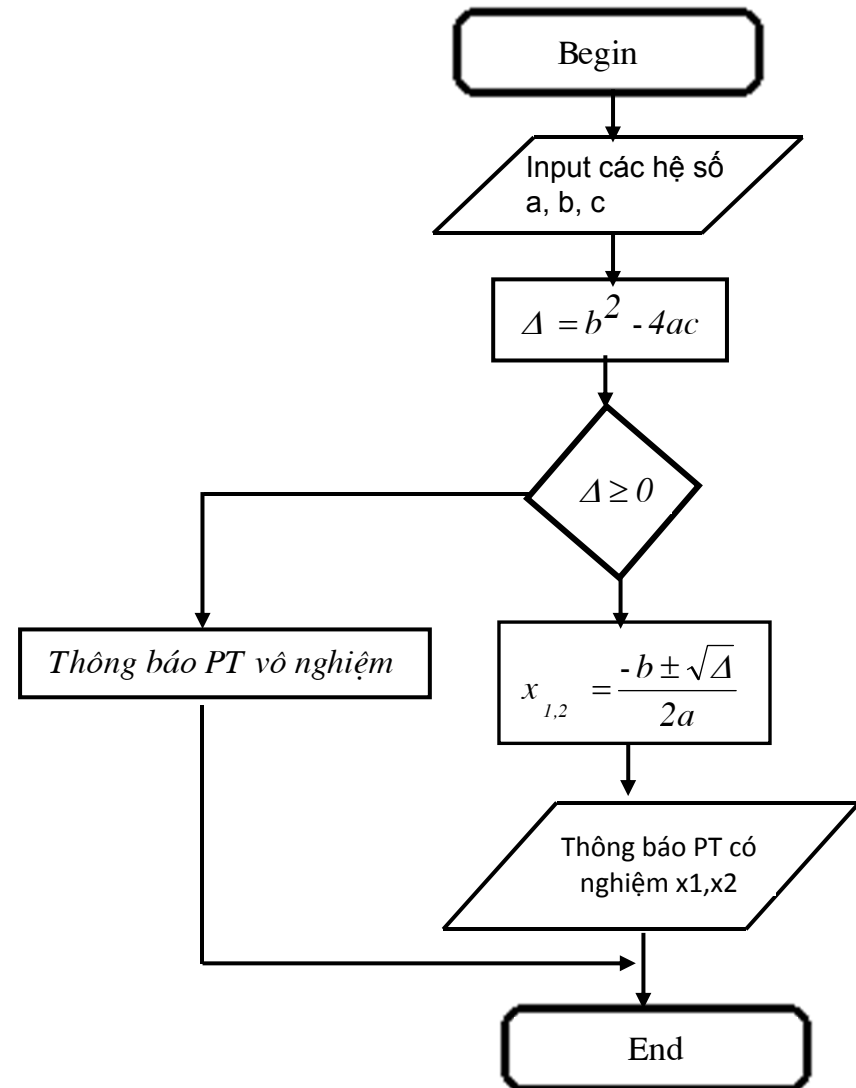


Chỉ chiều đi của thuật toán

### 3. Thuật toán và biểu diễn thuật toán (6/8)

#### **c.2. Biểu diễn sơ đồ, lưu đồ khối.**

- Ví dụ về sử dụng sơ đồ khối:
- Một số ưu, nhược điểm:
  - Có tính tổng quát cao,
  - thống nhất,
  - khắc phục được tính đa nghĩa và hàng rào ngôn ngữ,
  - khó hiểu với đại đa số những người khác chuyên ngành,
  - khó biểu diễn với giải thuật lớn.



### 3. Thuật toán và biểu diễn thuật toán (7/8)

#### ***c.3. Biểu diễn giả lệnh ngôn ngữ lập trình.***

- Có thể sử dụng giả mã lệnh để biểu diễn giải thuật.
- Với giả mã lệnh, có thể hiểu thuật toán mà không phụ thuộc vào ngôn ngữ lập trình.
- Phương pháp này rất thông dụng và dễ dàng sử dụng.
- Tuy nhiên, khó chuyển đổi đối với trường hợp quá tổng quát.

### 3. Thuật toán và biểu diễn thuật toán (8/8)

**c.3. Biểu diễn giả lệnh ngôn ngữ lập trình.**

**Ví dụ về biểu diễn bằng giả mã lệnh:**

**Begin**

**Input  $a, b, c$ ;**

**$\Delta = b^2 - 4ac$ ;**

**if  $\Delta \geq 0$  then**

**Begin**

**$x1 = (-b + \text{Sqrt}(\Delta)) / (2a)$ ;**

**$x2 = (-b - \text{Sqrt}(\Delta)) / (2a)$ ;**

**Output 'Phương trình có 2 nghiệm  $x1 = ', x1,$  và  $x2 = ', x2$ ;**

**End;**

**Else**

**Output 'Phương trình vô nghiệm';**