

---

# LÝ THUYẾT TH

---

TS. Lê Nh<sup>t</sup> Duy

Blog: <https://Lnduy.wordpress.com>

Email: Ln.duy@mail.ru

# Nội dung chương trình

## ■ Mục tiêu môn học

Cung cấp cho sinh viên các khái niệm cơ bản của lý thuyết đồ thị, đồ Euler, Hamilton, cây và cây khung bênh tказа th, bài toán đường đi ngắn nhất và bài toán luồng có cát trong mạng  $\Rightarrow$  Giúp sinh viên có thể sử dụng mô hình lý thuyết để mô hình hóa vấn đề bài toán thông qua cách hiểu quan . Hỗn hợp này trang bị những kỹ năng cần thiết cho các ngành thu côn lanh và CNTT.

## ■ Thời lượng

- Lý thuyết : 45 tiết

# Nội dung chương trình

1. TH
2. CÂY
3. LOGIC MÃ NH

# Kiểm tra ánh giá

- Kiểm tra gián k
- Tיע luận/bài tập lén theo nhóm
- Thi kiểm thúc môn

# Giáo trình và TLTK

## ■ Giáo trình

**Kenneth H.Rosen**, *Toán rời rạc - ngดng trong tin h c*, NXB Khoa h c k thu t. Hà n i-1997. (Ph m V n Thi u và ng H u Th nh d ch).

## ■ Tài liệu tham khảo

Slides bài giảng của giảng viên.

# Rules

■ ...

# Lý thuyết đồ thị

Chương 1: Các khái niệm cơ bản

# Nội dung



**I. Định nghĩa đồ thị**



**II. Các loại đồ thị**



**III. Các thuật ngữ cơ bản trong đồ thị**



**IV. Đường đi, chu trình**



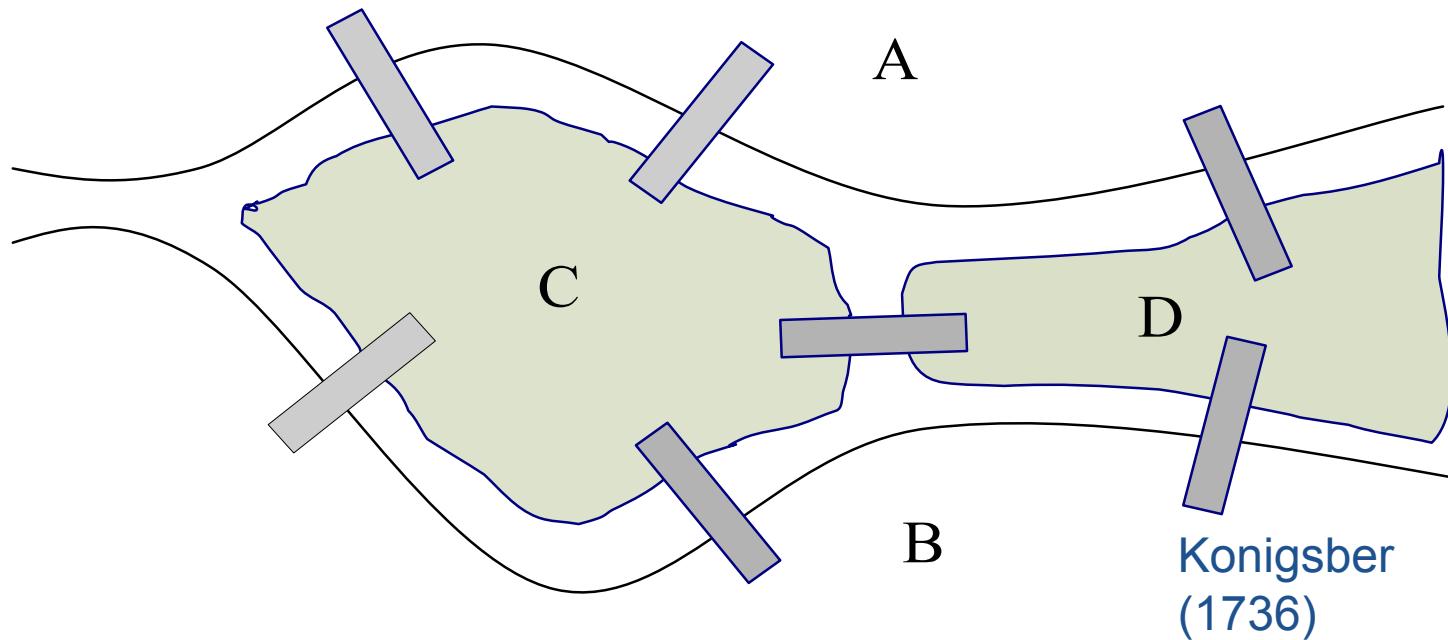
**V. Đồ thị liên thông**



**VI. Một số dạng đồ thị đặc biệt**

# I. Định nghĩa đồ thị

## ❖ Bài toán Euler

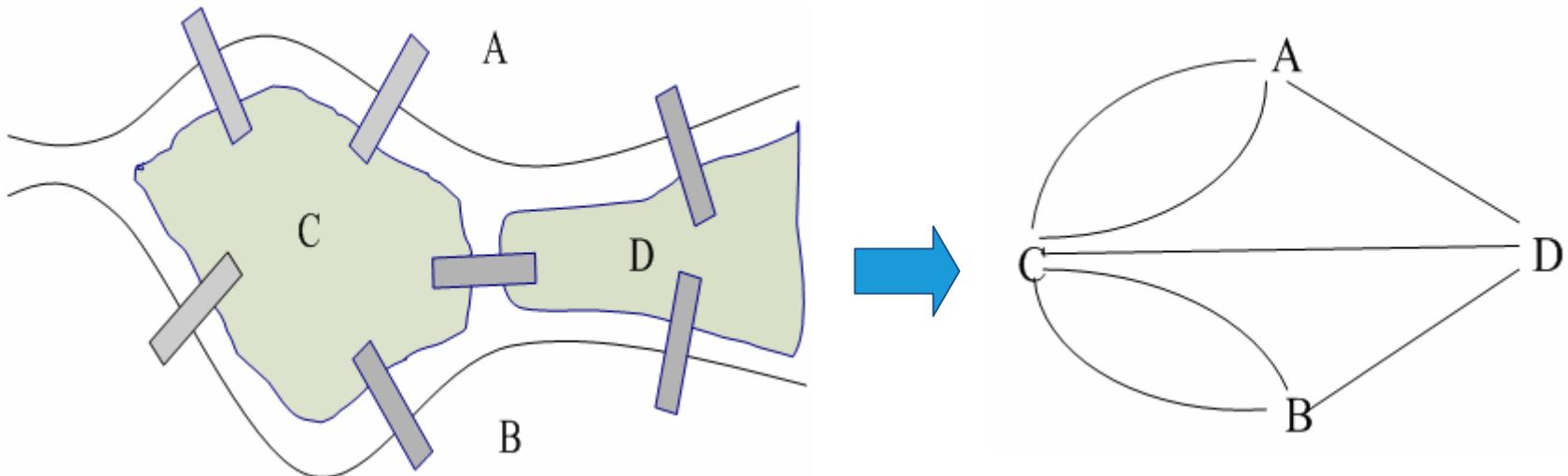


Có thể chỉ một lần đi qua tất cả 7 chiếc cầu này hay không?

# I. Định nghĩa đồ thị

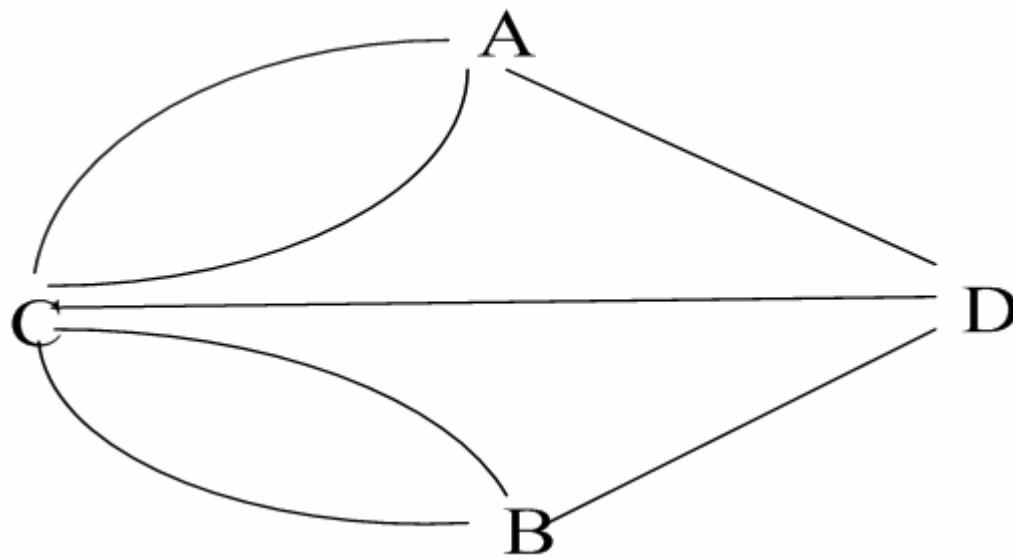
## ❖ Chuyển bài toán về dạng đồ thị

- Mỗi vùng là 1 đỉnh
- Mỗi chiếc cầu là 1 cạnh



# I. Định nghĩa đồ thị

- ❖ Đồ thị được xây dựng từ bài toán Euler
  - Có thể đi qua tất cả các cạnh của đồ thị, sao cho mỗi cạnh chỉ đi qua đúng một lần được không?



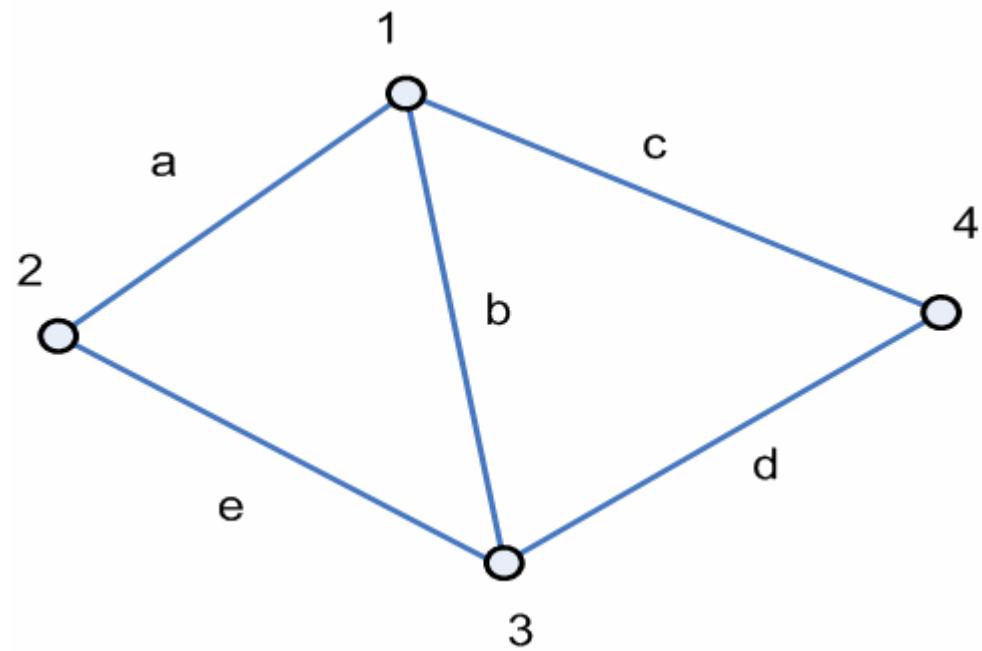
# I. Định nghĩa đồ thị

## ❖ Định nghĩa

- Đồ thị  $G$  là một tập hợp gồm các đỉnh và các cạnh. Ta thường ký hiệu:  $G = (V, E)$ , trong đó:

- +  $V$ : Là tập các đỉnh
- +  $E$ : Là tập các cạnh

$$\begin{aligned} V &= \{1, 2, 3, 4\} \\ E &= \{a, b, c, d, e\} \end{aligned}$$



# Nội dung



**I. Định nghĩa đồ thị**



**II. Các loại đồ thị**



**III. Các thuật ngữ cơ bản trong đồ thị**



**IV. Đường đi, chu trình**

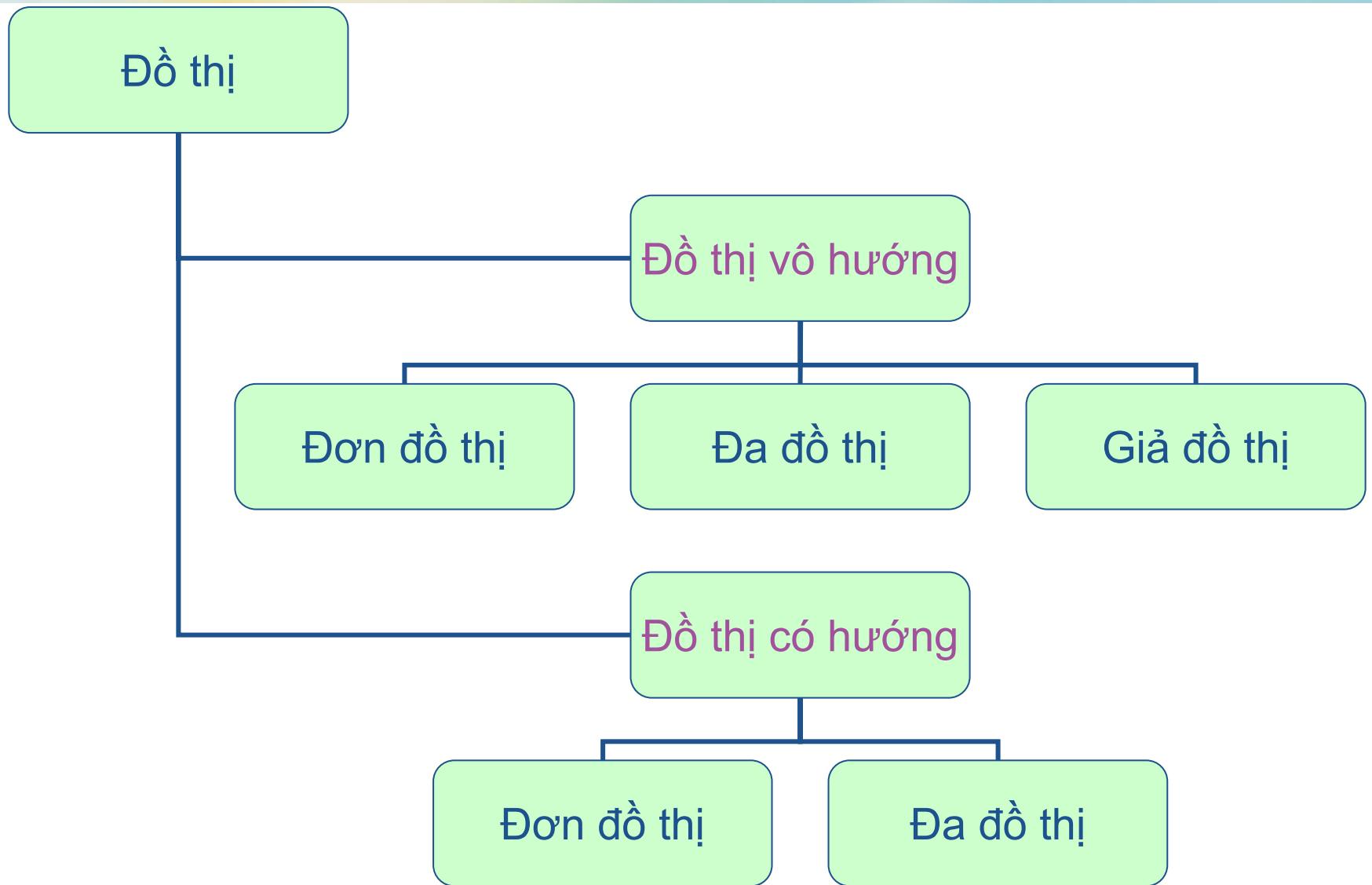


**V. Đồ thị liên thông**



**VI. Một số dạng đồ thị đặc biệt**

## II. Các loại đồ thị

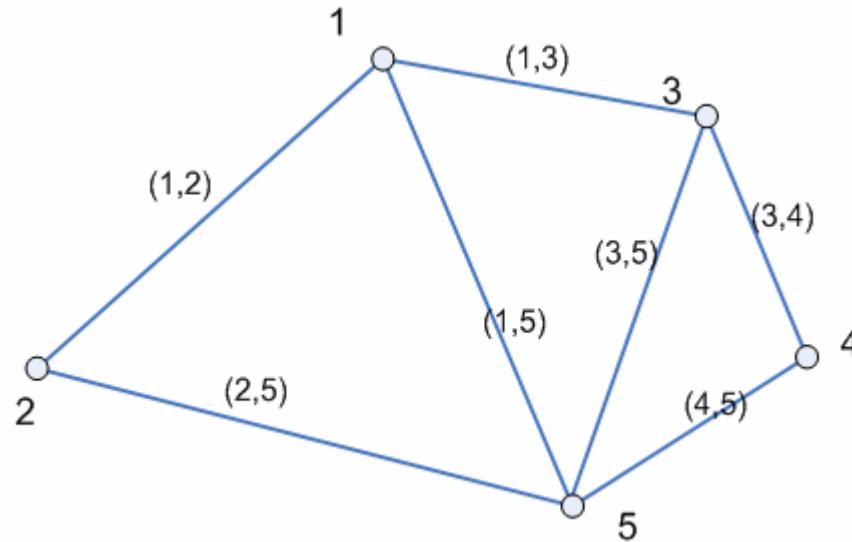


## II. Các loại đồ thị

### ❖ Đơn đồ thị vô hướng

Đồ thị  $G=(V, E)$  được gọi là đơn đồ thị vô hướng:

- $V$ : Là tập các đỉnh
- $E$ : là tập các cặp không có thứ tự gồm hai phần tử khác nhau của  $V$ .



$$V=\{1, 2, 3, 4, 5\}$$

$$E=\{(1, 2), (1, 3), (1, 5), (2, 5), (3, 4), (3, 5), (4, 5)\}$$

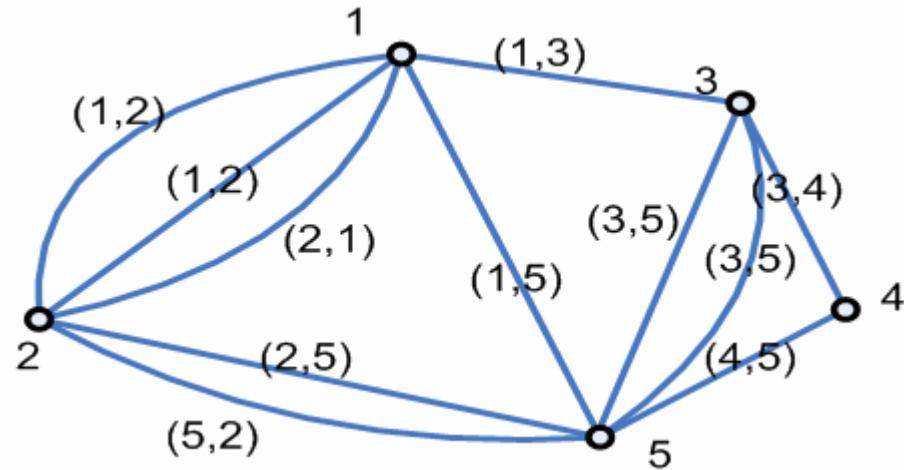
## II. Các loại đồ thị

### ❖ Đa đồ thị vô hướng

Đồ thị  $G=(V, E)$  được gọi là đa đồ thị vô hướng:

- $V$ : Là tập các đỉnh
- $E$ : Là họ các cặp không có thứ tự gồm hai phần tử khác nhau của  $V$ .

Hai cạnh  $e_1, e_2$  gọi là cạnh lặp nếu chúng cùng tương ứng với một cặp đỉnh



$$V=\{1, 2, 3, 4, 5\}$$

$$E=\{(1, 2), (1, 3), (1, 5), (2, 5), (3, 4), (3, 5), (4, 5), \textcolor{red}{(1, 2)}, \textcolor{red}{(2, 1)}, \textcolor{red}{(5, 2)}, \textcolor{red}{(3, 5)}\}$$

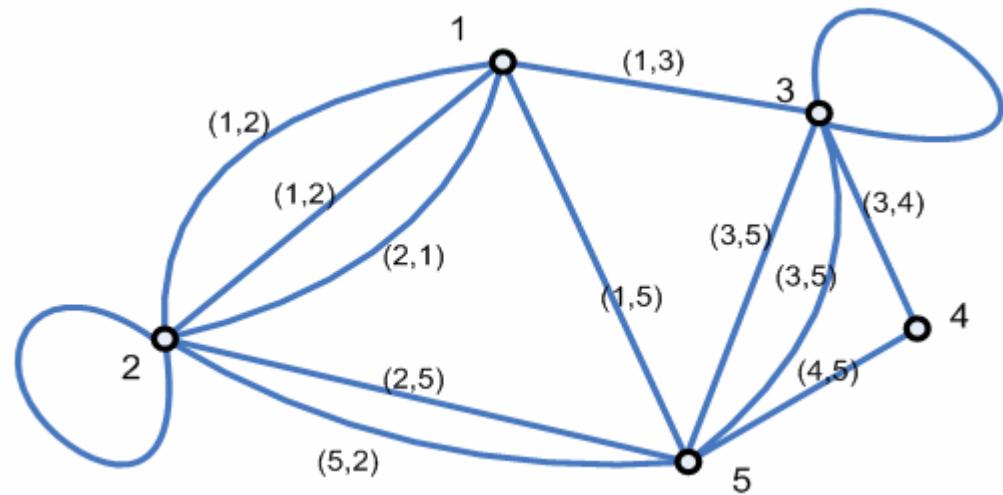
## II. Các loại đồ thị

### ❖ Giả đồ thị vô hướng

Đồ thị  $G=(V, E)$  được gọi là giả đồ thị vô hướng:

- $V$ : Là tập các đỉnh
- $E$ : Là họ các cặp không có thứ tự gồm hai phần tử không nhất thiết khác nhau của  $V$ .

Cạnh  $e$  được gọi là khuyên nếu nó có dạng:  $e=(u, u)$



$$V=\{1, 2, 3, 4, 5\}$$

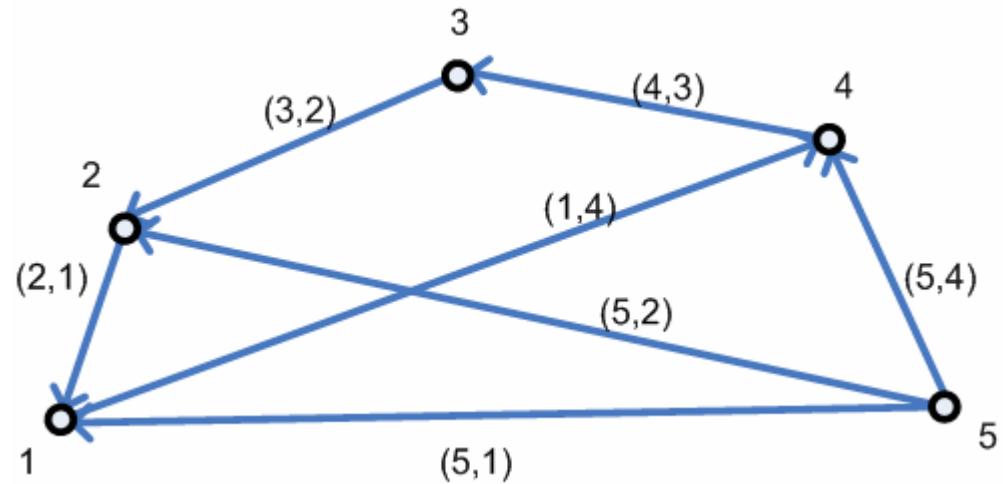
$$E=\{(1, 2), (1, 3), (1, 5), (2, 5), (3, 4), (3, 5), (4, 5), (1, 2), (2, 1), (5, 2), (3, 5), \textcolor{red}{(2, 2)}, \textcolor{red}{(3, 3)}\}$$

## II. Các loại đồ thị

### ❖ Đơn đồ thị có hướng

Đồ thị  $G=(V, E)$  được gọi là đơn đồ thị có hướng:

- $V$ : Là tập các đỉnh
- $E$ : Là tập các cặp có thứ tự gồm hai phần tử khác nhau của  $V$ .  
(tập các cung)



$$V=\{1, 2, 3, 4, 5\}$$

$$E=\{(2, 1), (1, 3), (5, 1), (4, 2), (3, 4), (3, 5), (5, 4)\}$$

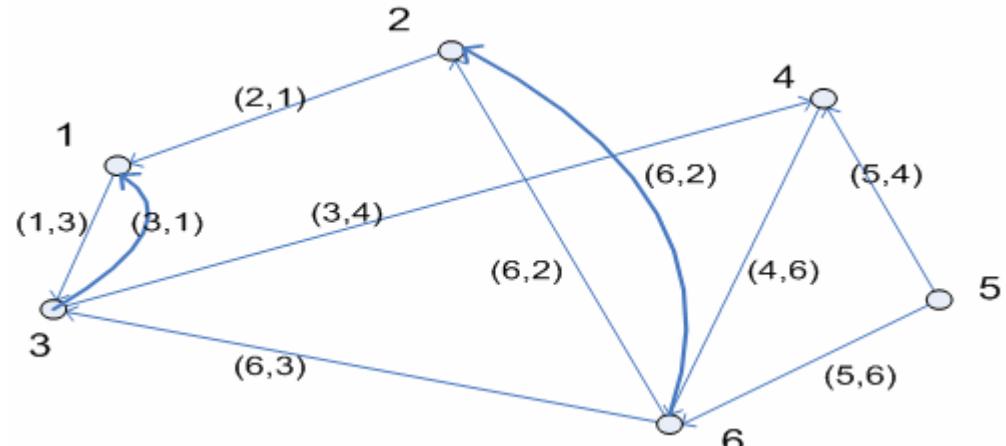
## II. Các loại đồ thị

### ❖ Đa đồ thị có hướng

Đồ thị  $G=(V, E)$  được gọi là đơn đồ thị có hướng:

- $V$ : Là tập các đỉnh
- $E$ : Là họ các cặp có thứ tự gồm hai phần tử khác nhau của  $V$ .  
(tập các cung)

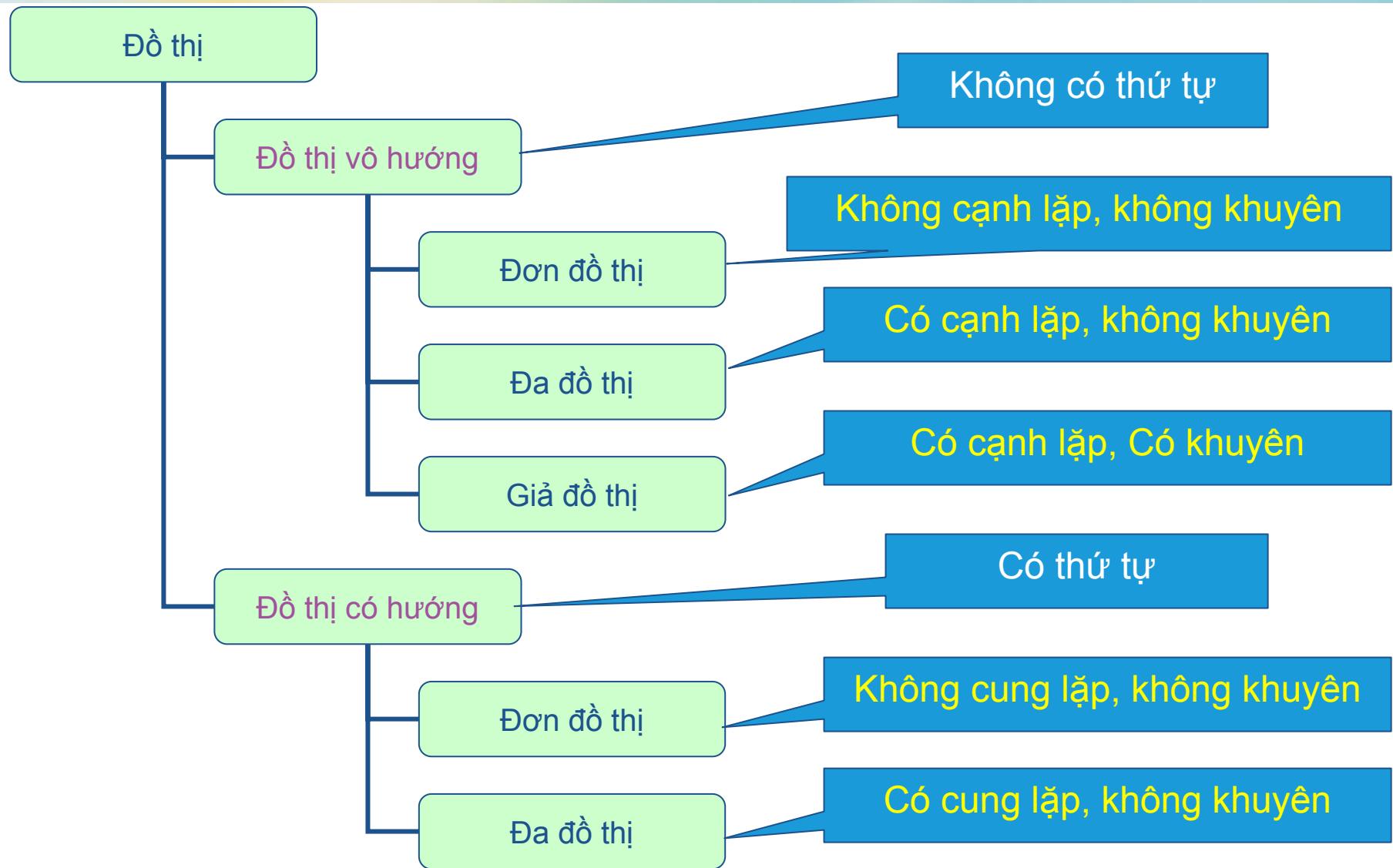
Hai cung  $e_1, e_2$  được gọi là cung lặp nếu chúng cùng tương ứng với một cặp đỉnh.



$$V=\{1, 2, 3, 4, 5\}$$

$$E=\{(2, 1), (1, 3), (6, 2), (3, 4), (6, 3), (4, 6), (5, 4), (5, 6), (3,1), (6,2)\}$$

## II. Các loại đồ thị



# Nội dung



I. Định nghĩa đồ thị



II. Các loại đồ thị



III. Các thuật ngữ cơ bản trong đồ thị



IV. Đường đi, chu trình



V. Đồ thị liên thông

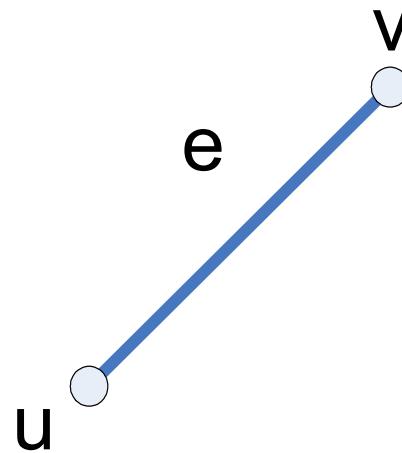


VI. Một số dạng đồ thị đặc biệt

### III. Các thuật ngữ cơ bản

#### ❖ Kề và liên thuộc

- Giả sử  $u$  và  $v$  là hai đỉnh của đồ thị vô hướng  $G$  và  $e=(u, v)$  là cạnh của đồ thị, khi đó ta nói:
  - +  $u$  và  $v$  kề nhau và  $e$  liên thuộc với  $u$  và  $v$ .
  - +  $u$  và  $v$  là các đỉnh đầu của cạnh  $e$

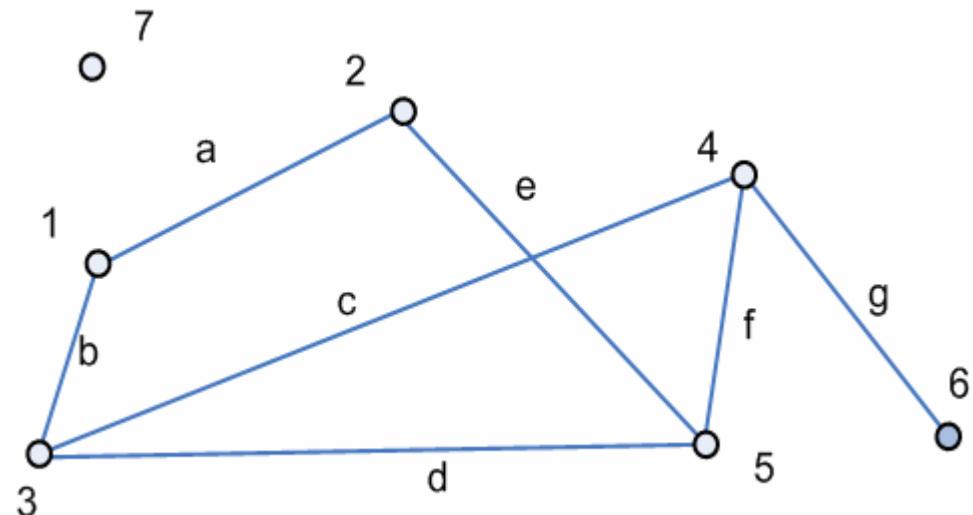


### III. Các thuật ngữ cơ bản

#### ❖ Bậc của đỉnh

- Bậc của đỉnh  $v$  trong đồ thị vô hướng là số cạnh liên thuộc với nó.
- Ký hiệu:  $\deg(v)$

$\deg(1)= 2$ ,  $\deg(2)= 2$ ,  
 $\deg(3)= 3$ ,  $\deg(4)= 3$ ,  
 $\deg(5)= 3$ ,  $\deg(6)= 1$ ,  
 $\deg(7)= 0$ .



- Đỉnh treo là đỉnh chỉ có duy nhất một cạnh liên thuộc với nó. → Đỉnh 6
- Đỉnh cô lập là đỉnh không có cạnh nào liên thuộc với nó. → Đỉnh 7

### III. Các thuật ngữ cơ bản

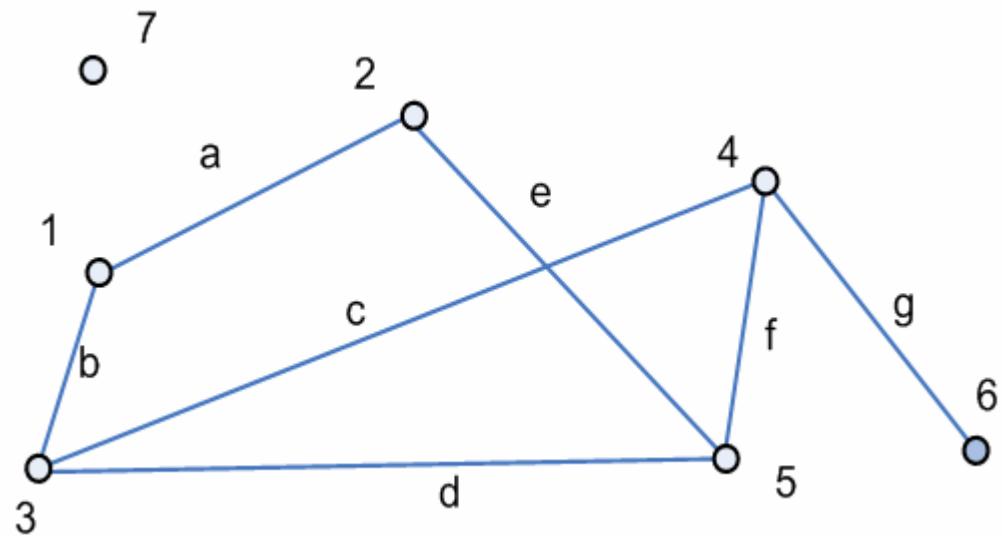
#### ❖ Định lý bắt tay

Giả sử  $G=(V,E)$  là đồ thị vô hướng với  $m$  cạnh. Khi đó tổng tất cả các bậc của đỉnh trong  $V$  bằng  $2m$ .

$$\sum_{v \in V} \deg(v) = 2m$$

$$m = 7$$

$$\sum_{v \in V} \deg(v) = 2m = 14$$



### III. Các thuật ngữ cơ bản

#### ❖ Định lý bắt tay

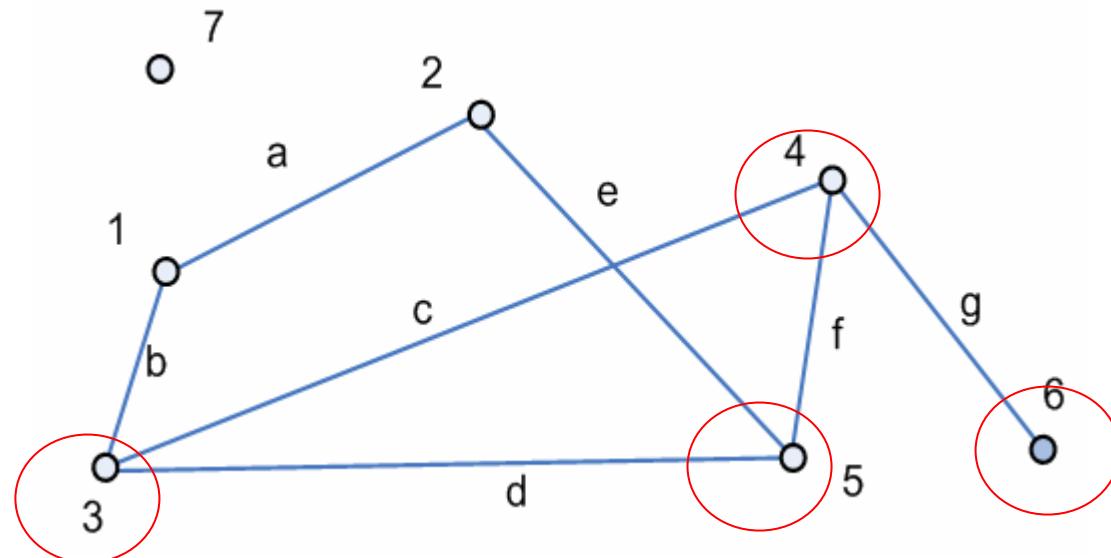
Chứng minh?

- ❖ Mỗi một cạnh nối với đúng hai đỉnh, vì thế mỗi cạnh đóng góp 2 đơn vị vào tổng các bậc của tất cả các đỉnh,  
➔ tổng các bậc của tất cả các đỉnh gấp đôi số cạnh của đồ thị

### III. Các thuật ngữ cơ bản

#### ❖ Hệ quả của định lý bắt tay

*Trong đồ thị vô hướng, số đỉnh bậc lẻ là một số chẵn.*



Các đỉnh bậc lẻ: 3, 5, 4, 6 → 4 đỉnh

### III. Các thuật ngữ cơ bản

#### ❖ Hệ quả của định lý bắt tay

*Trong đồ thị vô hướng, số đỉnh bậc lẻ là một số chẵn.*

Chứng minh:?

- ❖ Gọi L và C lần lượt là tập các đỉnh bậc lẻ và bậc chẵn của đồ thị vô hướng  $G = (V, E)$ . Ta có:

$$2m = \sum_{v \in V} \deg(v) = \sum_{v \in L} \deg(v) + \sum_{v \in C} \deg(v)$$

+ Tổng  $2m$  chẵn

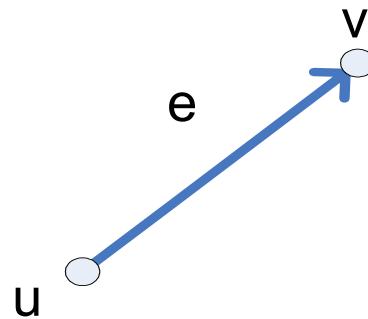
+ Tổng  $\sum_{v \in C} \deg(v)$  chẵn

➔ Tổng  $\sum_{v \in L} \deg(v)$  chẵn

### III. Các thuật ngữ cơ bản

#### ❖ Kề trong đồ thị có hướng

- Giả sử  $u$  và  $v$  là hai đỉnh của đồ thị có hướng  $G$  và  $e=(u, v)$  là một cung của đồ thị, khi đó ta nói:
  - +  $u$  và  $v$  kề nhau, cung  $e$  đi ra khỏi  $u$  và đi vào  $v$ .
  - +  $u$  là đỉnh đầu,  $v$  là đỉnh cuối của cạnh  $e$ .



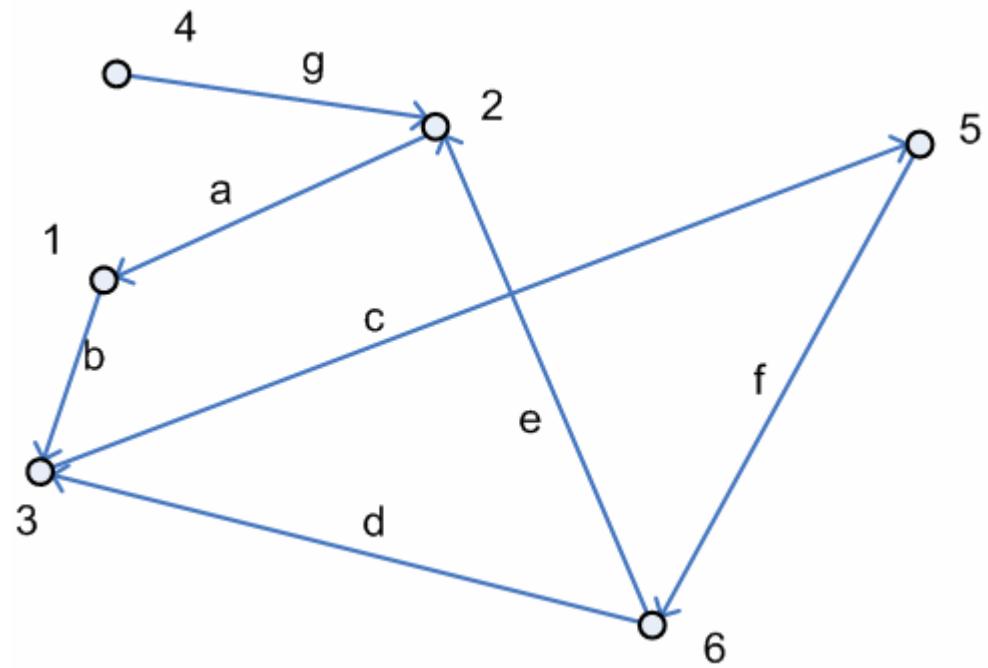
### III. Các thuật ngữ cơ bản

#### ❖ Bán bậc vào và bán bậc ra của đỉnh

- Bán bậc ra (bán bậc vào) của đỉnh  $v$  trong đồ thị có hướng là số cung ra khỏi nó (đi vào nó).
- Ký hiệu:  $\deg^+(v)$      $(\deg^-(v))$

$$\deg^+(2) = 1, \deg^-(2) = 2$$

$$\deg^+(6) = 2, \deg^-(6) = 1$$



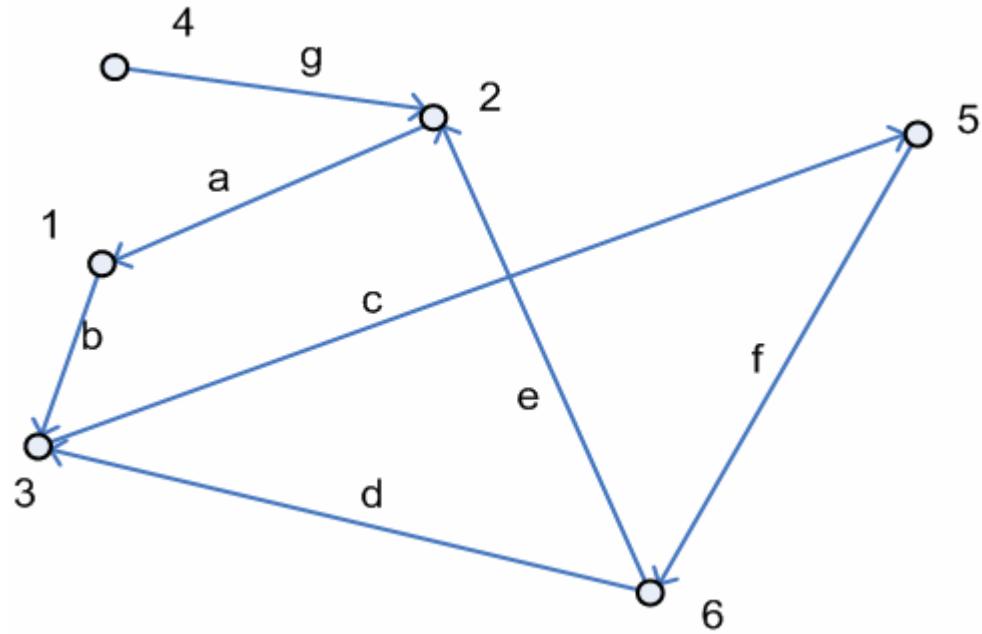
### III. Các thuật ngữ cơ bản

#### ❖ Định lý

Giả sử  $G=(V,E)$  là đồ thị có hướng với  $m$  cung, khi đó tổng tất cả các bán bậc ra bằng tổng tất cả các bán bậc vào và bằng  $m$ .

$$\sum_{v \in V} \deg^+(v) = \sum_{v \in V} \deg^-(v) = m$$

$$\sum_{v \in V} \deg^+(v) = \sum_{v \in V} \deg^-(v) = 7$$



### III. Các thuật ngữ cơ bản

#### ❖ Bài tập

1. Có bao nhiêu cạnh trong đồ thị có 10 đỉnh, mỗi đỉnh có bậc bằng 6
  - a) 20
  - b) 30
  - c) 40
  - d) 50
2. Cho biết các đỉnh của đồ thị có bậc lần lượt là: 4, 3, 3, 2, 2. Số cạnh của đồ thị này là:
  - a) 5
  - b) 6
  - c) 7
  - d) 8
3. Cho danh sách bậc các đỉnh của các đồ thị sau, đồ thị nào không tồn tại?
  - a) 3, 3, 3, 3, 2
  - b) 1, 2, 3, 4, 5
  - c) 0, 1, 2, 2, 3
  - d) 1, 1, 1, 1

### III. Các thuật ngữ cơ bản

#### ❖ Bài tập

4. Có thể tồn tại đồ thị đơn 15 đỉnh, mỗi đỉnh có bậc bằng 5 hay không?
5. Trong một giải thi đấu có  $n$  đội tham dự và đã có  $n+1$  trận đấu được tiến hành. CMR có 1 đội đã thi đấu ít nhất 3 trận.

# Nội dung



**I. Định nghĩa đồ thị**



**II. Các loại đồ thị**



**III. Các thuật ngữ cơ bản trong đồ thị**



**IV. Đường đi, chu trình**



**V. Đồ thị liên thông**



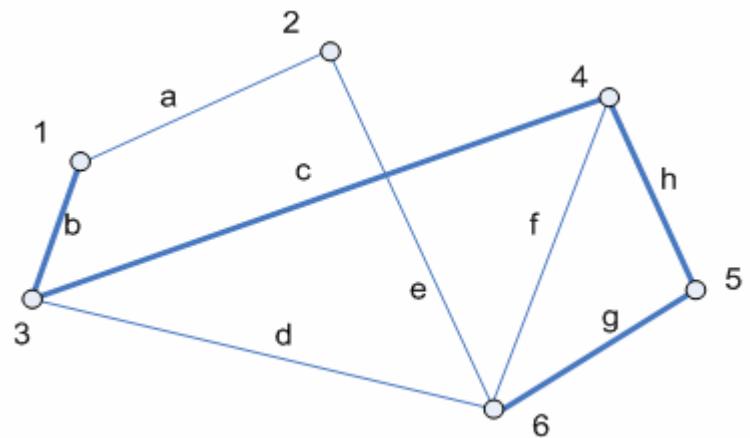
**VI. Một số dạng đồ thị đặc biệt**

## IV. Đường đi, chu trình

- ❖ Đường đi độ dài n từ đỉnh u đến đỉnh v trên đồ thị vô hướng  $G=(V,E)$  là dãy(theo đỉnh):  $x_0, x_1, \dots, x_{n-1}, x_n$ . Trong đó:
  - +  $u = x_0$
  - +  $v = x_n$
  - +  $(x_i, x_{i+1}) \in E$
- ❖ Hay theo cạnh:  $(x_0, x_1), (x_1, x_2), \dots, (x_{n-1}, x_n)$ .
- ❖ Khi đó: u gọi là đỉnh đầu, v gọi là đỉnh cuối của đường đi.

Theo đỉnh: (1, 3, 4, 5, 6)

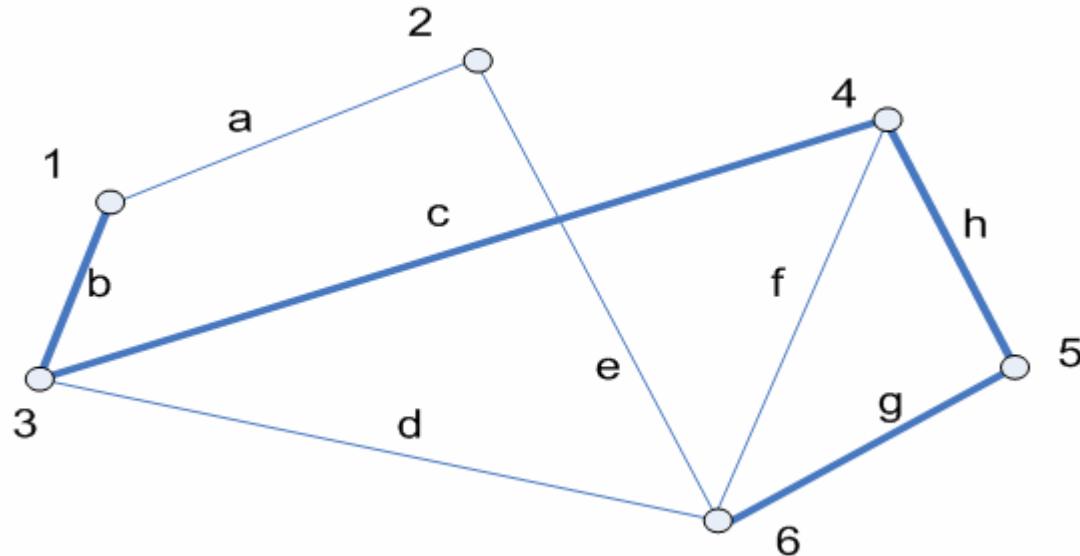
Theo cạnh: (b, c, h, g)



## IV. Đường đi, chu trình

❖ Đường đi có đỉnh đầu và đỉnh cuối trùng nhau gọi là chu trình.

- Đường đi (hay chu trình) được gọi là đơn nếu nó không đi qua một cạnh nào quá một lần.



Chu trình đơn: (1, 2, 6, 3, 1)

Chu trình không phải chu trình đơn: (2, 6, 4, 3, 6, 2)

## IV. Đường đi, chu trình

### ❖ Đường đi và chu trình trong đồ thị có hướng

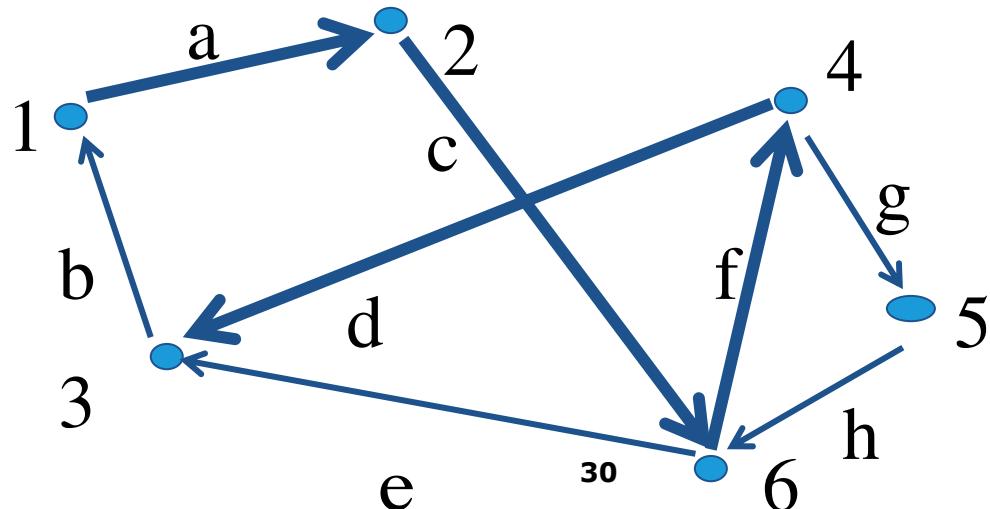
**Đường đi** độ dài n ( $n \in \mathbb{N}^+$ ) từ đỉnh u đến đỉnh v trên đồ thị có hướng

$G = (V, E)$  là dãy:

$x_0, x_1, \dots, x_{n-1}, x_n$ .

Trong đó  $u = x_0, v = x_n, (x_i, x_{i+1}) \in E$

Hay theo các cung:  $(x_0, x_1), (x_1, x_2), \dots, (x_{n-1}, x_n)$ .



(1, 2, 6, 4, 3)  
(a, c, f, d)  
(1, 3, 4, 5, 6)

# Nội dung



**I. Định nghĩa đồ thị**



**II. Các loại đồ thị**



**III. Các thuật ngữ cơ bản trong đồ thị**



**IV. Đường đi, chu trình**



**V. Đồ thị liên thông**

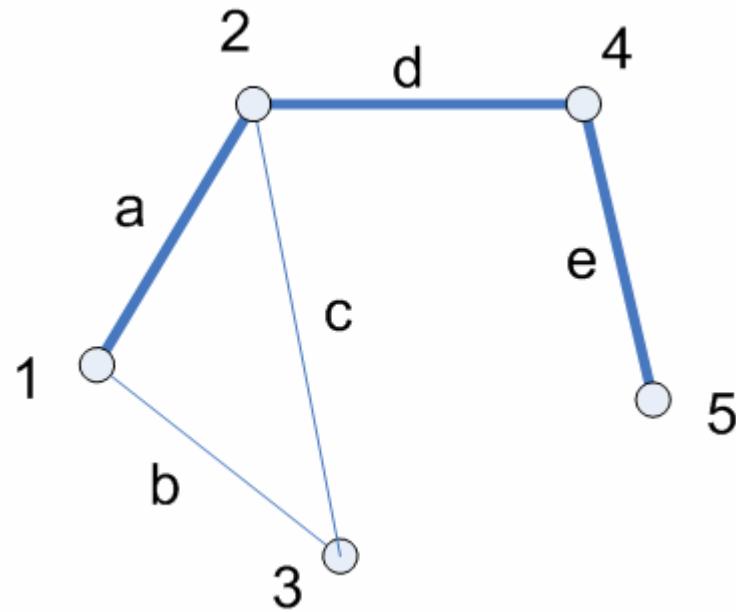


**VI. Một số dạng đồ thị đặc biệt**

## V.Đồ thị liên thông

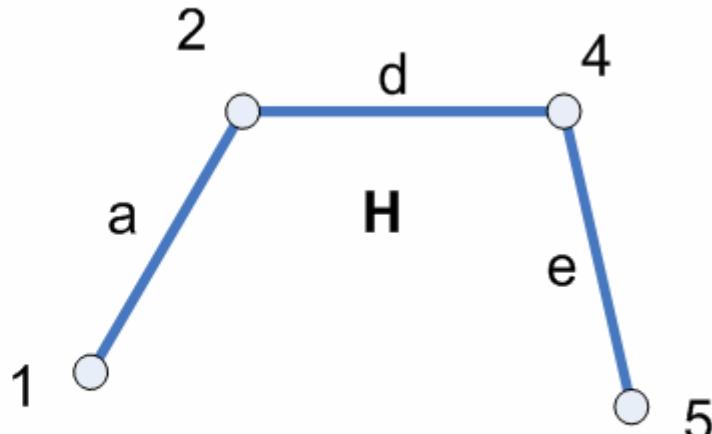
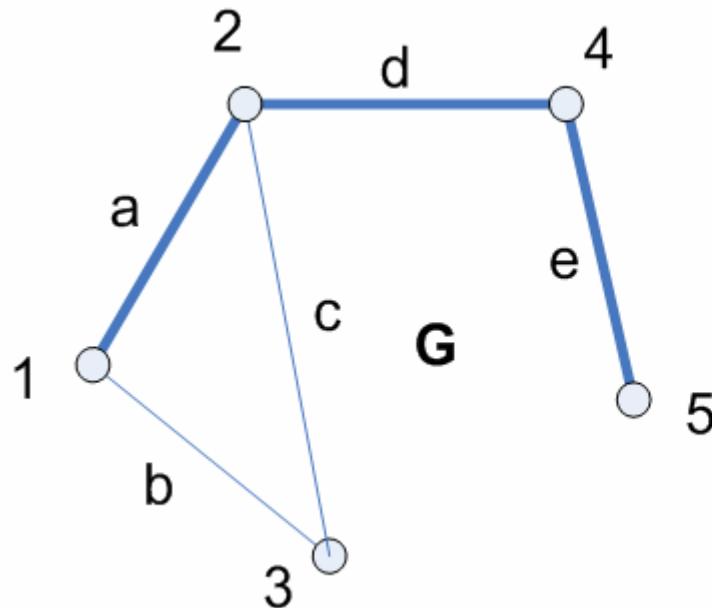
- ❖ Đồ thị vô hướng  $G=(V,E)$  được gọi là liên thông nếu luôn tìm được đường đi giữa 2 đỉnh bất kỳ của nó.

Đường đi: 1, 3, 2, 4, 5



# V.Đồ thị liên thông

- ❖ Đồ thị  $H=(W,F)$  được gọi là **đồ thị con** của đồ thị  $G=(V,E)$  nếu :  $W \subseteq V$  và  $F \subseteq E$

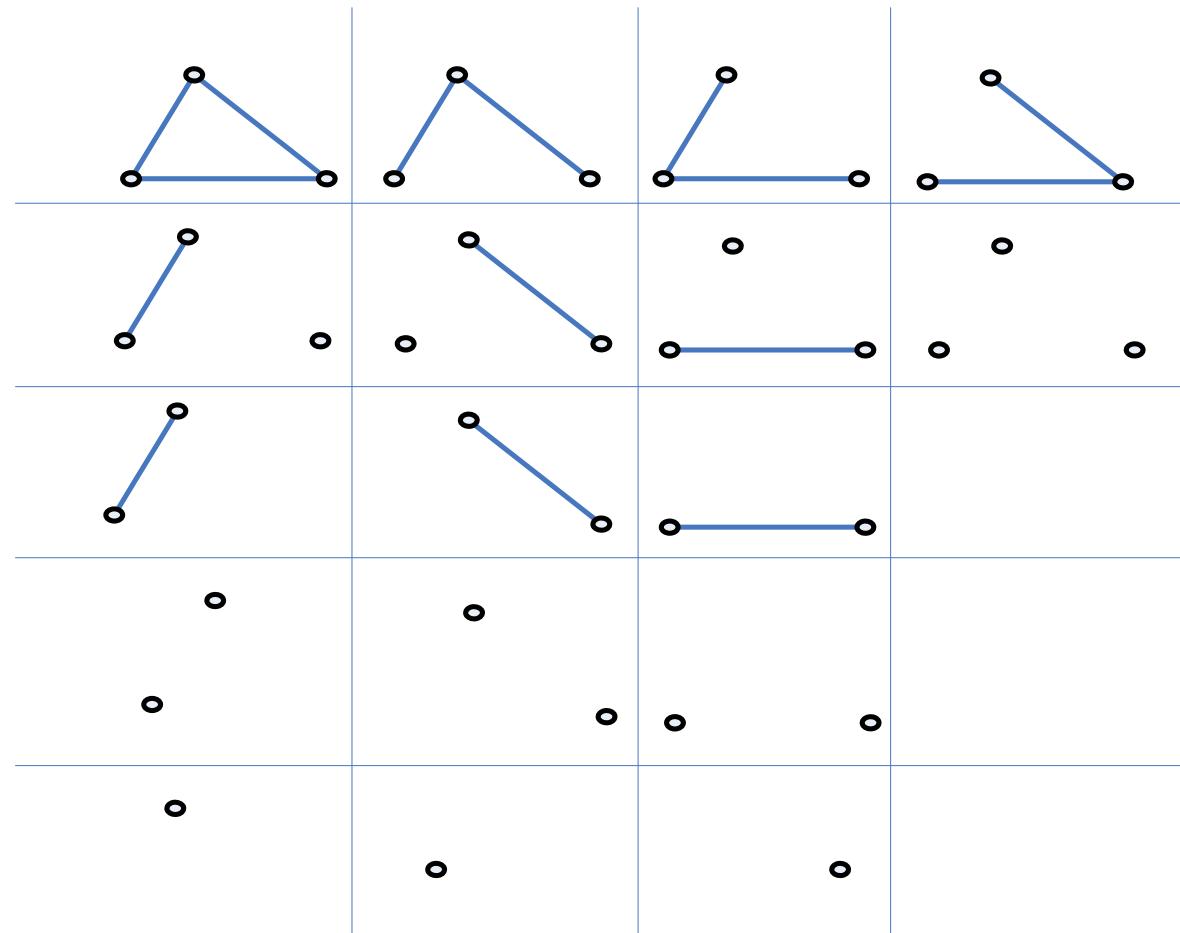


$$\begin{array}{ll} V=\{1, 2, 3, 4, 5\} & W=\{1, 2, 4, 5\} \\ E=\{a, b, c, d, e\} & F=\{a, d, e\} \end{array}$$

# VI. Một số dạng đồ thị đặc biệt

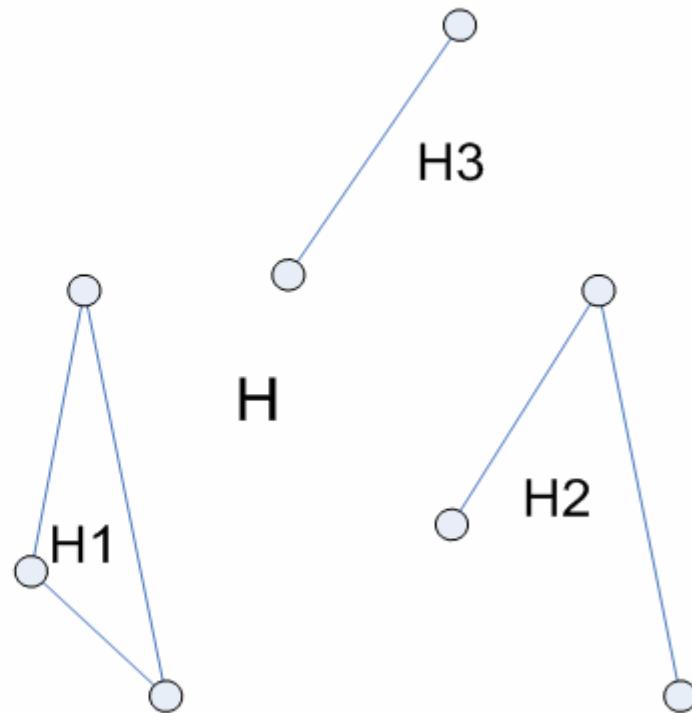
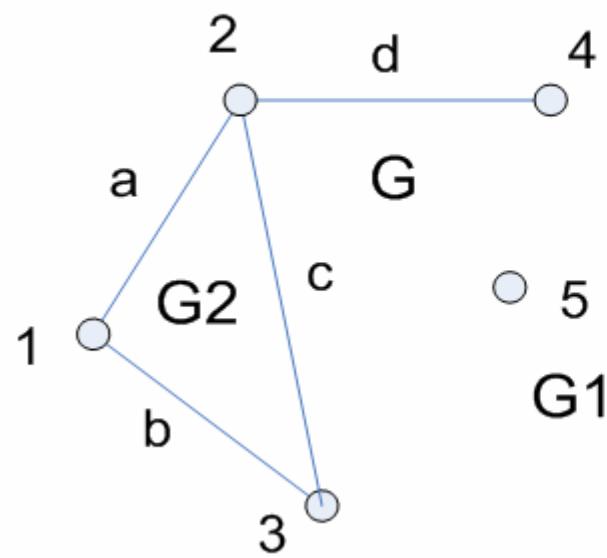
## ❖ Bài tập

1. Đồ thị  $K_3$  có bao nhiêu đồ thị con có ít nhất một đỉnh ?



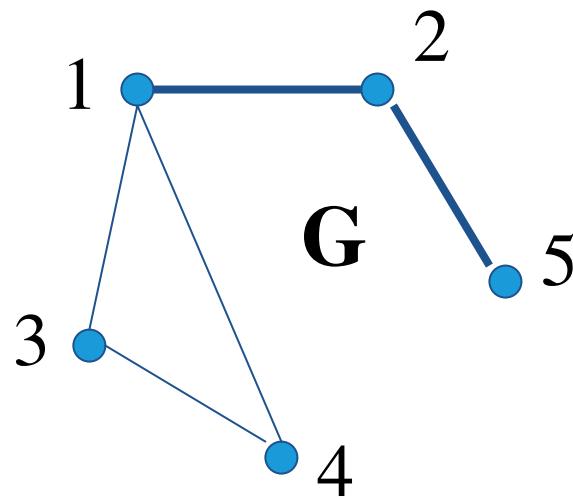
## V.Đồ thị liên thông

- ❖ Một đồ thị không liên thông sẽ được phân rã thành các thành phần liên thông, và mỗi thành phần liên thông này là một đồ thị con của đồ thị ban đầu.



## V. Đồ thị liên thông

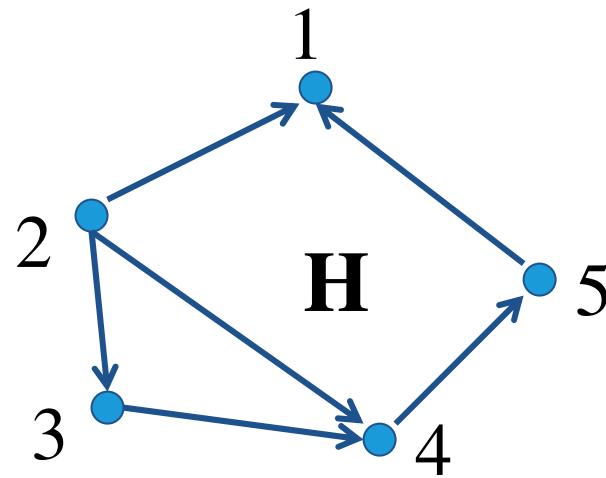
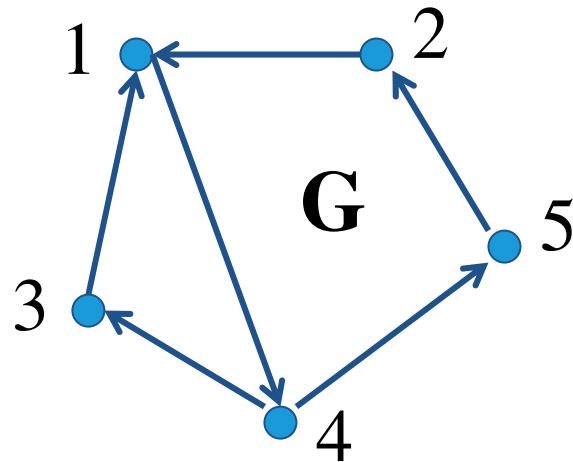
- Đỉnh v được gọi là đỉnh rẽ nhánh nếu việc loại bỏ v cùng các cạnh liên thuộc với nó sẽ làm tăng số thành phần liên thông của đồ thị
- Cạnh e được gọi là cầu nếu việc loại bỏ nó sẽ làm tăng số thành phần liên thông của đồ thị



Các đỉnh rẽ nhánh?  
Các cạnh là cầu ?

# V.Đồ thị liên thông

- Đồ thị có hướng  $G=(V,E)$  được gọi là liên thông mạnh nếu luôn tìm được đường đi từ 1 đỉnh bất kỳ đến một đỉnh bất kỳ khác của nó.
- Đồ thị có hướng  $G=(V,E)$  được gọi là liên thông yếu nếu đồ thị vô hướng tương ứng với nó là đồ thị vô hướng liên thông.



## V.Đồ thị liên thông

### ❖ Bài tập

1. Trong 1 đồ thị G có chứa đúng 2 đỉnh bậc lẻ (các đỉnh còn lại nếu có đều bậc chẵn). CM có 1 đường đi nối 2 đỉnh bậc lẻ đó với nhau.

# Nội dung



**I. Định nghĩa đồ thị**



**II. Các loại đồ thị**



**III. Các thuật ngữ cơ bản trong đồ thị**



**IV. Đường đi, chu trình**



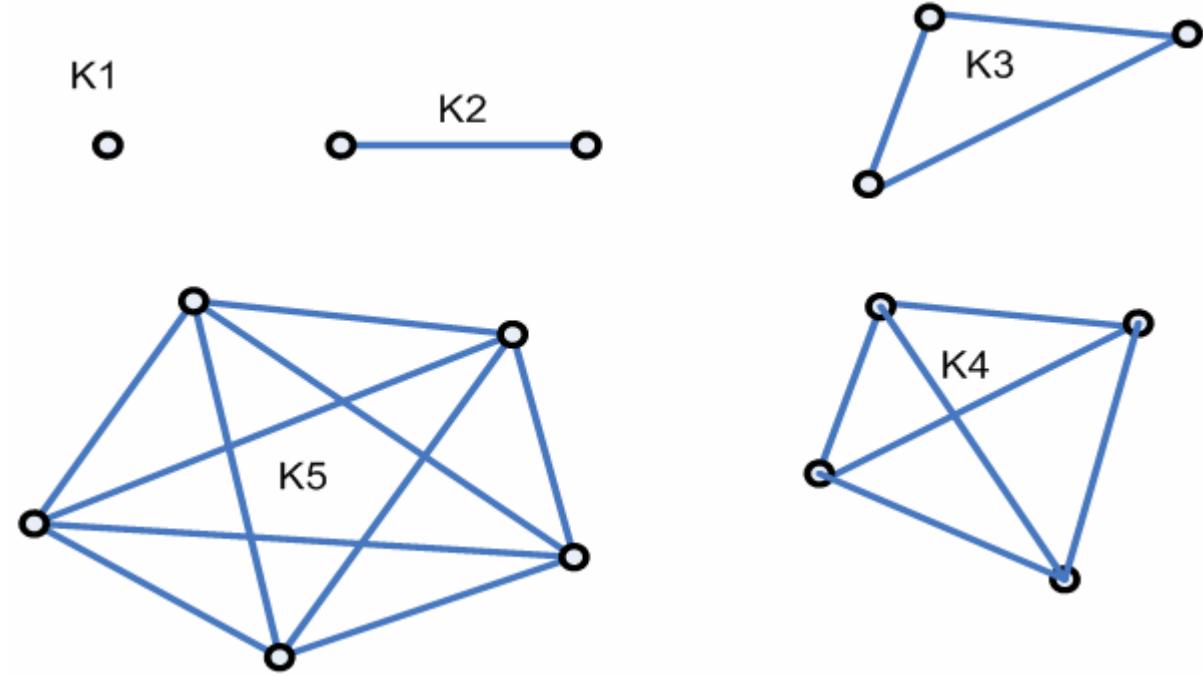
**V. Đồ thị liên thông**



**VI. Một số dạng đồ thị đặc biệt**

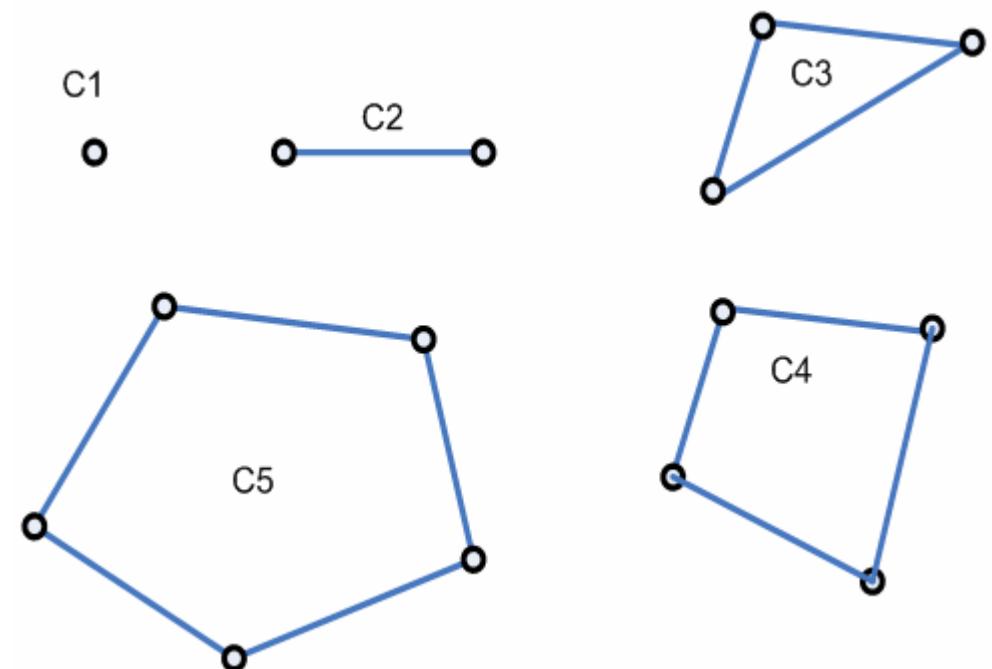
## VI. Một số dạng đồ thị đặc biệt

- ❖ **Đồ thị đầy đủ**: Một đồ thị đơn vô hướng n đỉnh được gọi là đồ thị đầy đủ nếu hai đỉnh bất kỳ đều được nối với nhau bằng 1 cạnh.
- ❖ **Ký hiệu**:  $K_n$



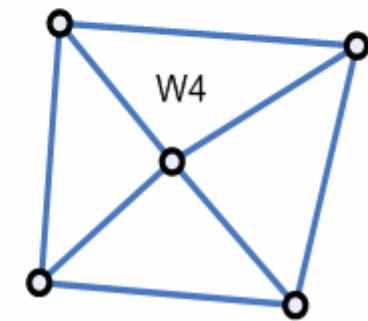
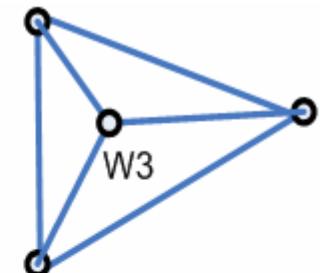
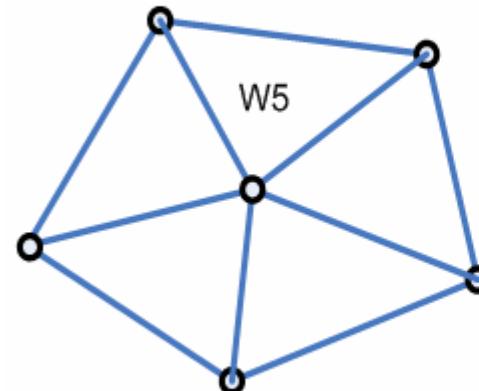
## VI. Một số dạng đồ thị đặc biệt

- ❖ **Đồ thị vòng**: Một đồ thị đơn vô hướng n đỉnh được gọi là đồ thị vòng nếu nó có duy nhất một chu trình đơn đi qua tất cả các đỉnh.
- ❖ **Ký hiệu**:  $C_n$



## VI. Một số dạng đồ thị đặc biệt

- ❖ **Đồ thị bánh xe** với  $n \geq 3$  đỉnh là đồ thị thu được từ đồ thị  $C_n$  bằng cách bổ xung thêm một đỉnh mới nối với tất cả các đỉnh của  $C_n$ .
- ❖ Ký hiệu:  $W_n$



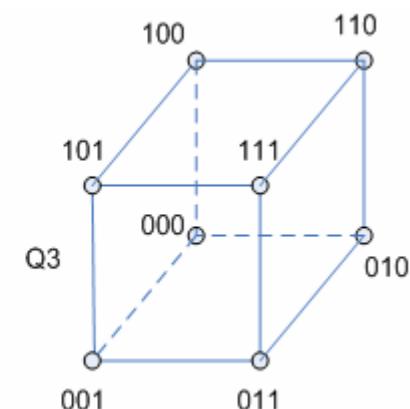
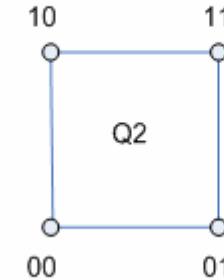
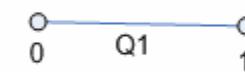
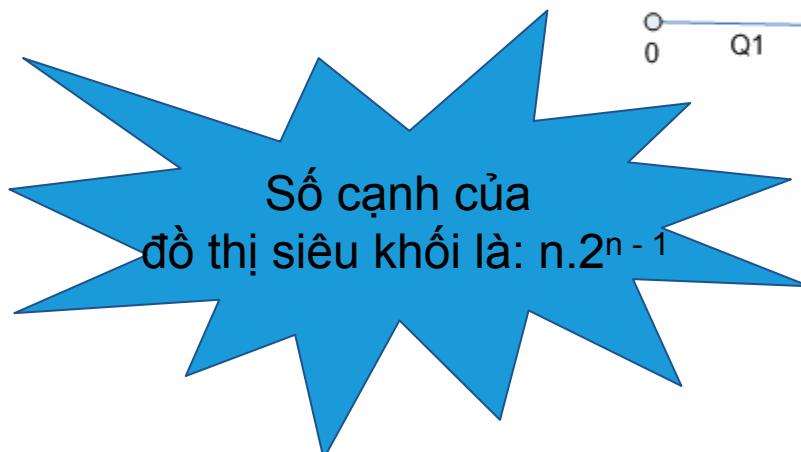
# VI. Một số dạng đồ thị đặc biệt

## ❖ Đồ thị siêu khối

Đồ thị siêu khối  $k=2^n$  đỉnh là đồ thị có các đỉnh được đánh số bằng các chuỗi nhị phân độ dài  $n$ .

## ❖ Ký hiệu: $Q_n$

❖ Hai đỉnh kề nhau nếu 2 chuỗi nhị phân tương ứng chỉ khác nhau 1 bit.

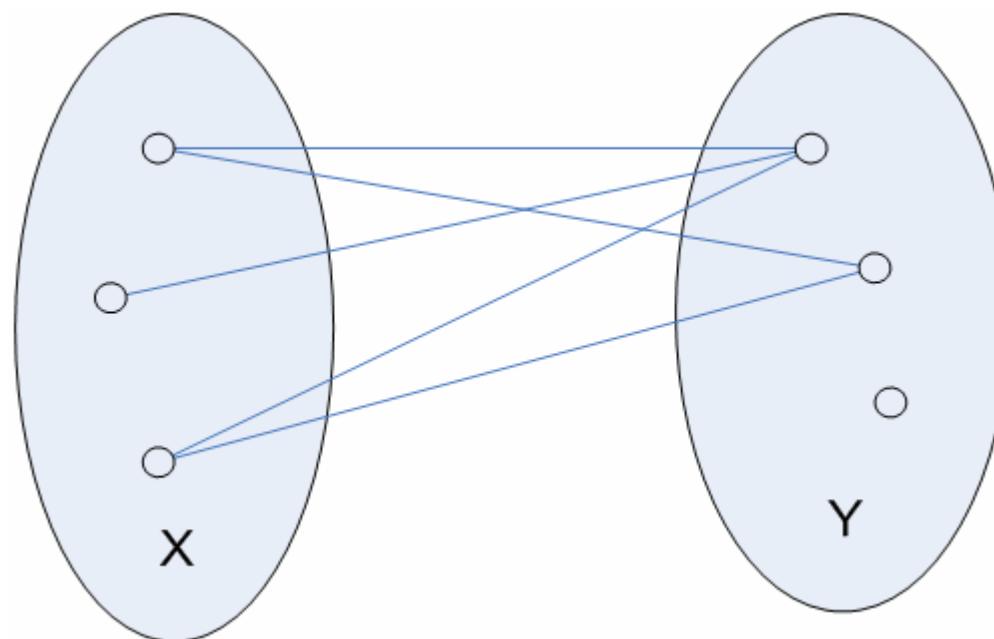


# VI. Một số dạng đồ thị đặc biệt

## ❖ Đồ thị hai phía

Đơn đồ thị  $G=(V, E)$  gọi là đồ thị hai phía nếu:

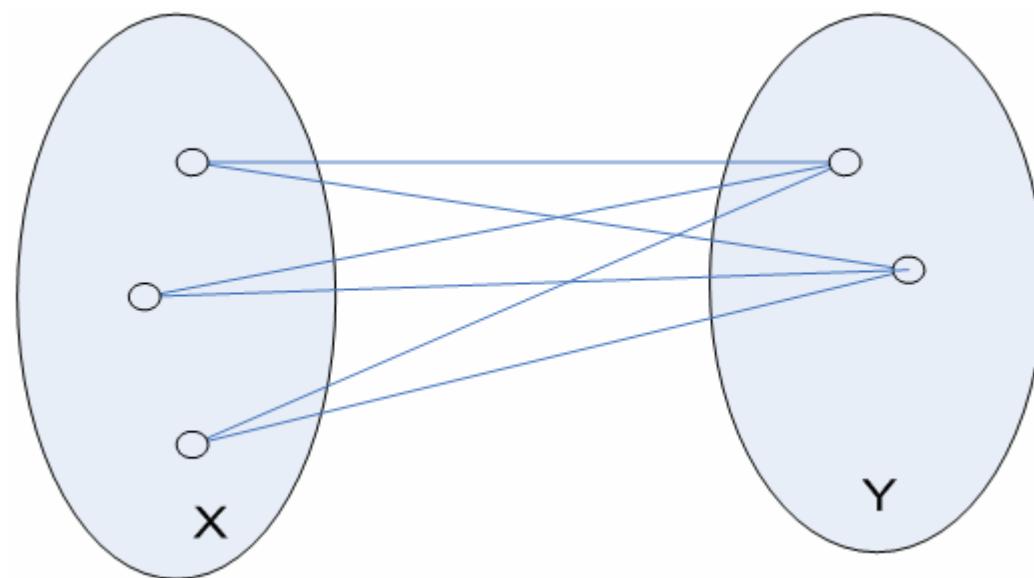
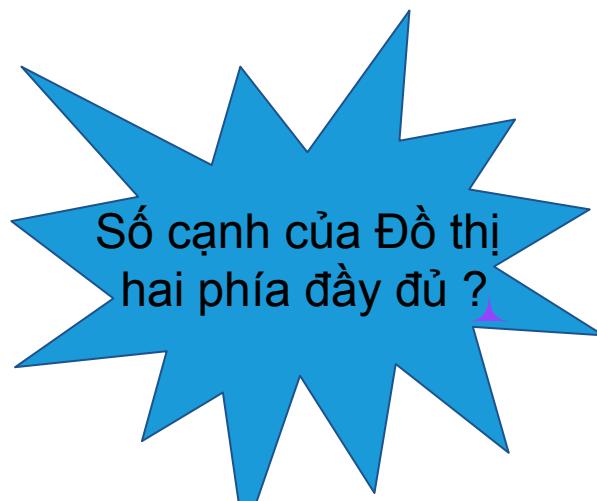
- $V = X \cup Y$ ,  $X \neq \emptyset$ ,  $Y \neq \emptyset$ ,  $X \cap Y = \emptyset$
- Mọi cạnh của  $G$  sẽ có một đỉnh thuộc  $X$  và một đỉnh thuộc  $Y$ .



# VI. Một số dạng đồ thị đặc biệt

## ❖ Đồ thị hai phía đầy đủ

Đơn đồ thị  $G = (X \cup Y, E)$  được gọi là đồ thị hai phía đầy đủ nếu: Mọi đỉnh thuộc  $X$  sẽ được nối với mỗi đỉnh thuộc  $Y$ . Nếu  $|X| = m$  và  $|Y| = n$  thì ta sẽ ký hiệu là:  $K_{m,n}$



# VI. Một số dạng đồ thị đặc biệt

## ❖ Định lý:

Đơn đồ thị  $G = (V, E)$  là đồ thị hai phía khi và chỉ khi nó không chứa chu trình độ dài lẻ.

## ❖ Chứng minh:

$\forall$  Đồ thị hai phía

$\Rightarrow$  Không chứa chu trình độ dài lẻ

$\forall$  Đồ thị, không chứa chu trình độ dài lẻ

$\Rightarrow$  hai phía

# VI. Một số dạng đồ thị đặc biệt

## ❖ Thuật toán kiểm tra đồ thị hai phía

1. Chọn  $v$  là đỉnh bất kỳ. Đặt  $X = \{v\}$
2.  $Y = \{ u \mid u \text{ kề với } v, \forall v \in X\}$
3. Nếu  $X \cap Y \neq \emptyset \Rightarrow G$  không là đồ thị hai phía
4. Ngược lại, đặt  $X := Y$  Quay trở lại 2.
5. Nếu tất cả các đỉnh được xét hết mà không xảy ra 3. thì  $G$  là đồ thị hai phía. Ngược lại  $G$  không là đồ thị hai phía.

# VI. Một số dạng đồ thị đặc biệt

❖ Ví dụ:

$$X = \{1\}$$

$$Y = \{5\}, X \cap Y = \emptyset, \Rightarrow X := Y$$

$$Y = \{1, 2\}, X \cap Y = \emptyset, \Rightarrow X := Y$$

$$Y = \{5, 6, 7\}, X \cap Y = \emptyset, \Rightarrow X := Y$$

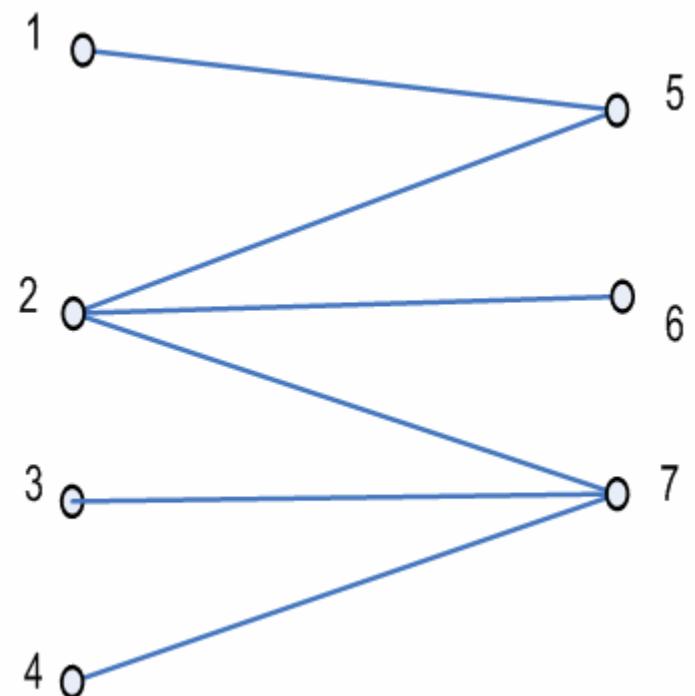
$$Y = \{1, 2, 3, 4\}$$

DÙNG

Khi đó đồ thị là hai phía:

$$X = \{1, 2, 3, 4\}$$

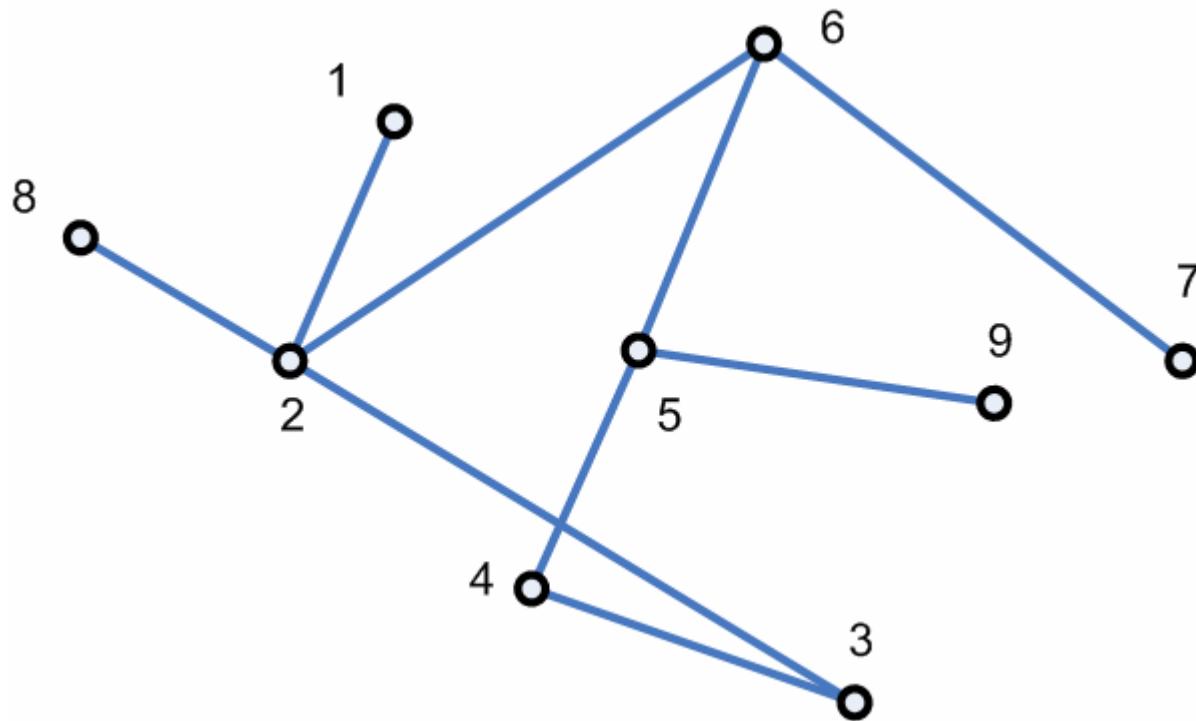
$$Y = \{5, 6, 7\}$$



# VI. Một số dạng đồ thị đặc biệt

## ❖ Bài tập:

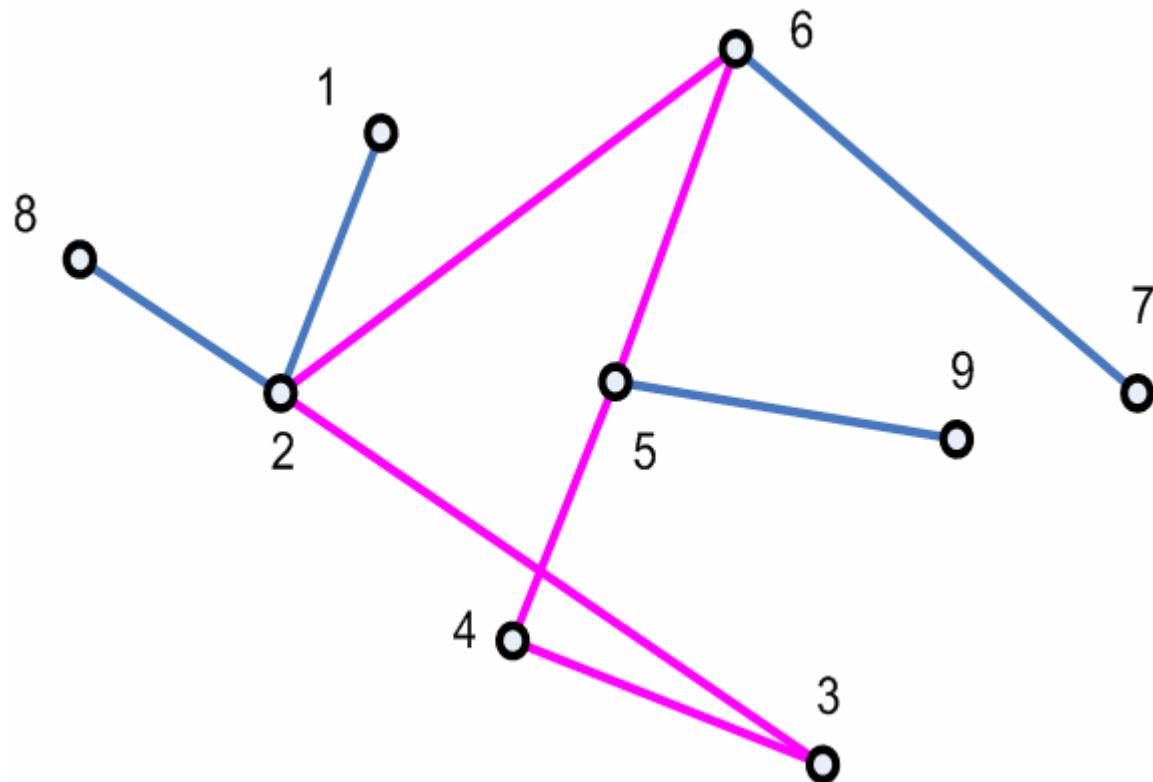
Kiểm tra đồ thị sau có phải là đồ thị hai phía hay không?



# VI. Một số dạng đồ thị đặc biệt

## ❖ Bài tập:

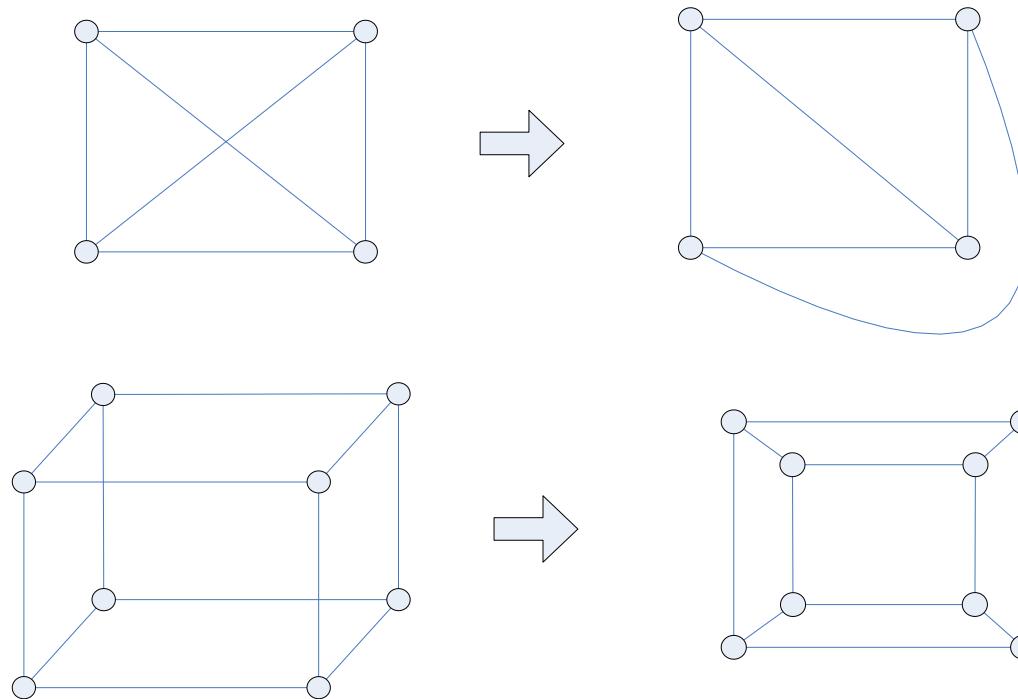
Không phải là đồ thị hai phía



# VI. Một số dạng đồ thị đặc biệt

## ❖ Đồ thị phẳng

Đồ thị được gọi là đồ thị phẳng nếu ta có thể vẽ nó trên một mặt phẳng mà các cạnh không giao nhau.



# VI. Một số dạng đồ thị đặc biệt

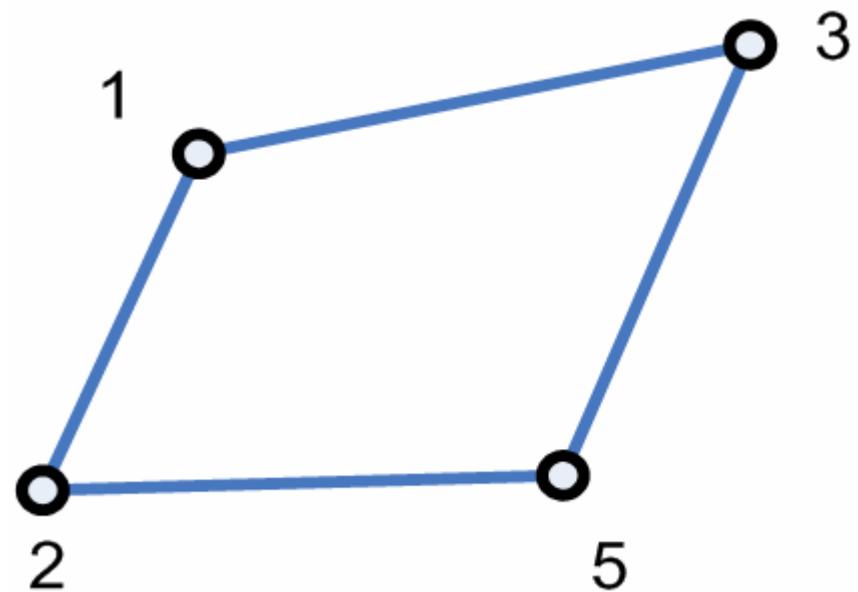
## ❖ Định lý Euler

Giả sử  $G = (V, E)$  là đồ thị phẳng, liên thông với  $e$  cạnh và  $v$  đỉnh. Gọi  $f$  là số mặt của đồ thị. Khi đó:  $f = e - v + 2$ .

Số cạnh:  $e = 4$

Số đỉnh:  $v = 4$

Số mặt:  $f = 4 - 4 + 2 = 2$



# VI. Một số dạng đồ thị đặc biệt

## ❖ Định lý Euler

Chứng minh: Bằng PP Quy nạp

- ❖ Gọi  $f_n$ ,  $e_n$ ,  $v_n$  lần lượt là số mặt, số cạnh, số đỉnh của đồ thị phẳng  $G_n$  do biểu diễn phẳng của đồ thị  $G$  với  $n$  cạnh sinh ra
  - + Trường hợp:  $e_1=1$ ,  $v_1=2$  thì  $f_1 = 1 - 2 + 2 = 1$



+ Giả sử đồ thị  $G_n$  ( $n$  cạnh) thỏa đẳng thức:  $f_n = e_n - v_n + 2$ .

Thêm vào đồ thị  $G_n$  một cạnh  $(a_{n+1}, b_{n+1})$  để được đồ thị  $G_{n+1}$ .

Ta phải chứng minh:  $f_{n+1} = e_{n+1} - v_{n+1} + 2$

Xảy ra hai trường hợp

# VI. Một số dạng đồ thị đặc biệt

## ❖ Định lý Euler (Chứng minh)

+ Cả 2 đỉnh  $a_{n+1}$ ,  $b_{n+1}$  thuộc  $G_n$ :

$$f_{n+1} = f_n + 1$$

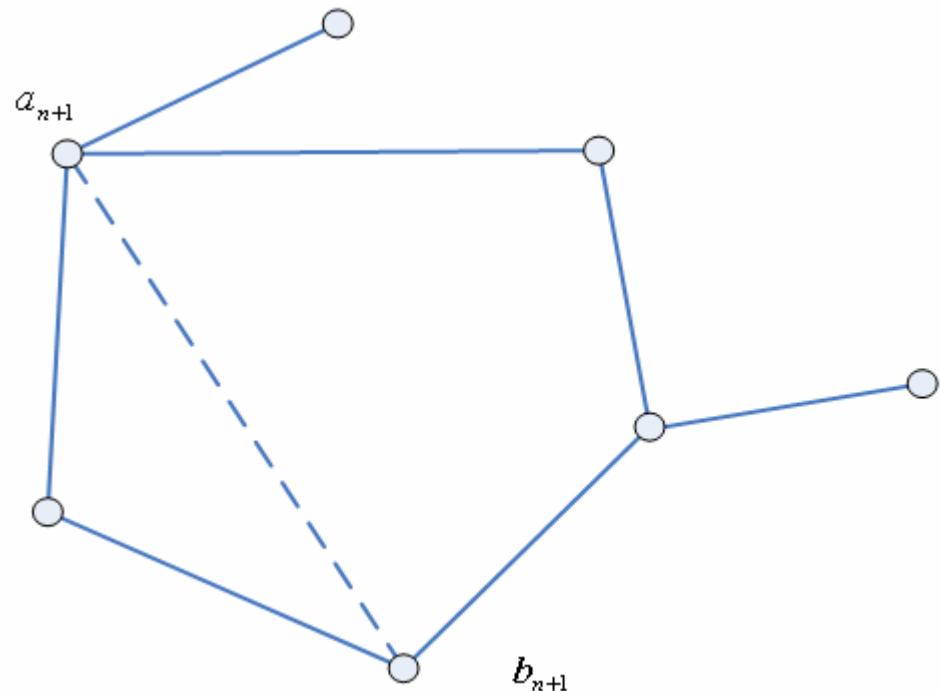
$$e_{n+1} = e_n + 1$$

$$v_{n+1} = v_n$$

$$\implies f_{n+1} = e_{n+1} - v_{n+1} + 2$$

$$f_n + 1 = e_n + 1 - v_n + 2$$

$$f_n = e_n - v_n + 2$$



# VI. Một số dạng đồ thị đặc biệt

## ❖ Định lý Euler (Chứng minh)

+ Cả 2 đỉnh  $a_{n+1}$ ,  $b_{n+1}$  thuộc

$G_n$ :

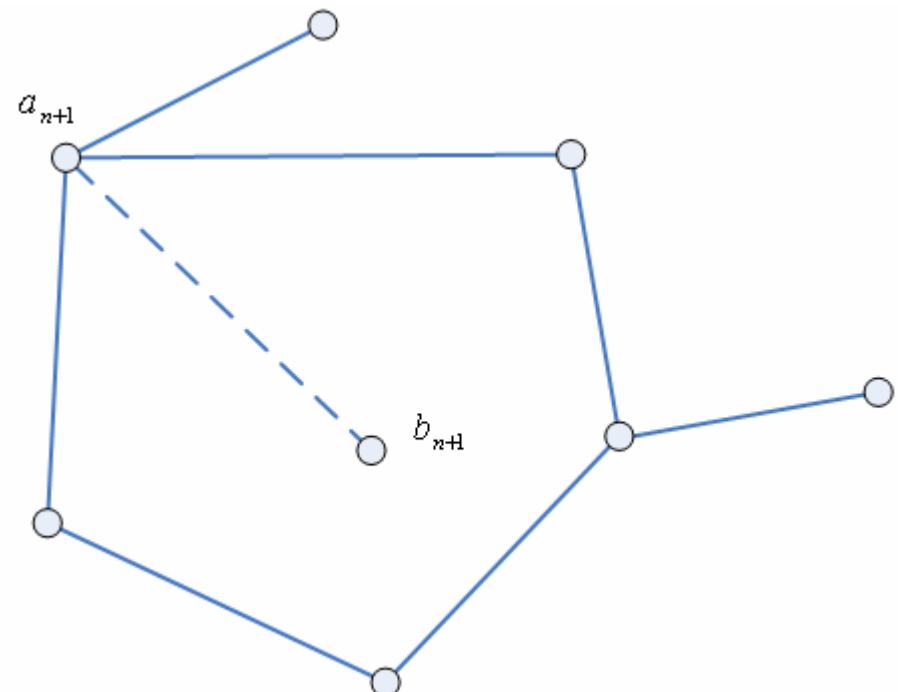
$$f_{n+1} = f_n$$

$$e_{n+1} = e_n + 1$$

$$v_{n+1} = v_n + 1$$

→  $f_{n+1} = e_{n+1} - v_{n+1} + 2$   
 $f_n = e_n + 1 - v_n + 1 + 2$   
 $f_n = e_n - v_n + 2$

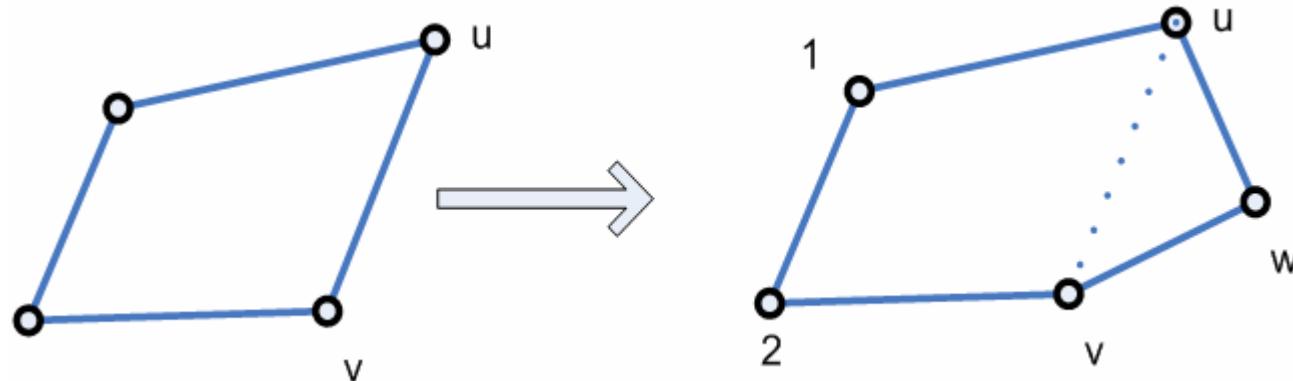
→ ĐPCM



# VI. Một số dạng đồ thị đặc biệt

## ❖ Định lý Kuratowski

Phép chia cạnh  $(u, v)$  là việc ta bỏ đi cạnh  $(u, v)$  và thêm vào một đỉnh mới  $w$  cùng với hai cạnh  $(u, w), (w, v)$ .



## ❖ Định nghĩa đồng cấu

Hai đồ thị được gọi là đồng cấu nếu chúng có thể thu được từ cùng một đồ thị nào đó nhờ các phép chia cạnh.

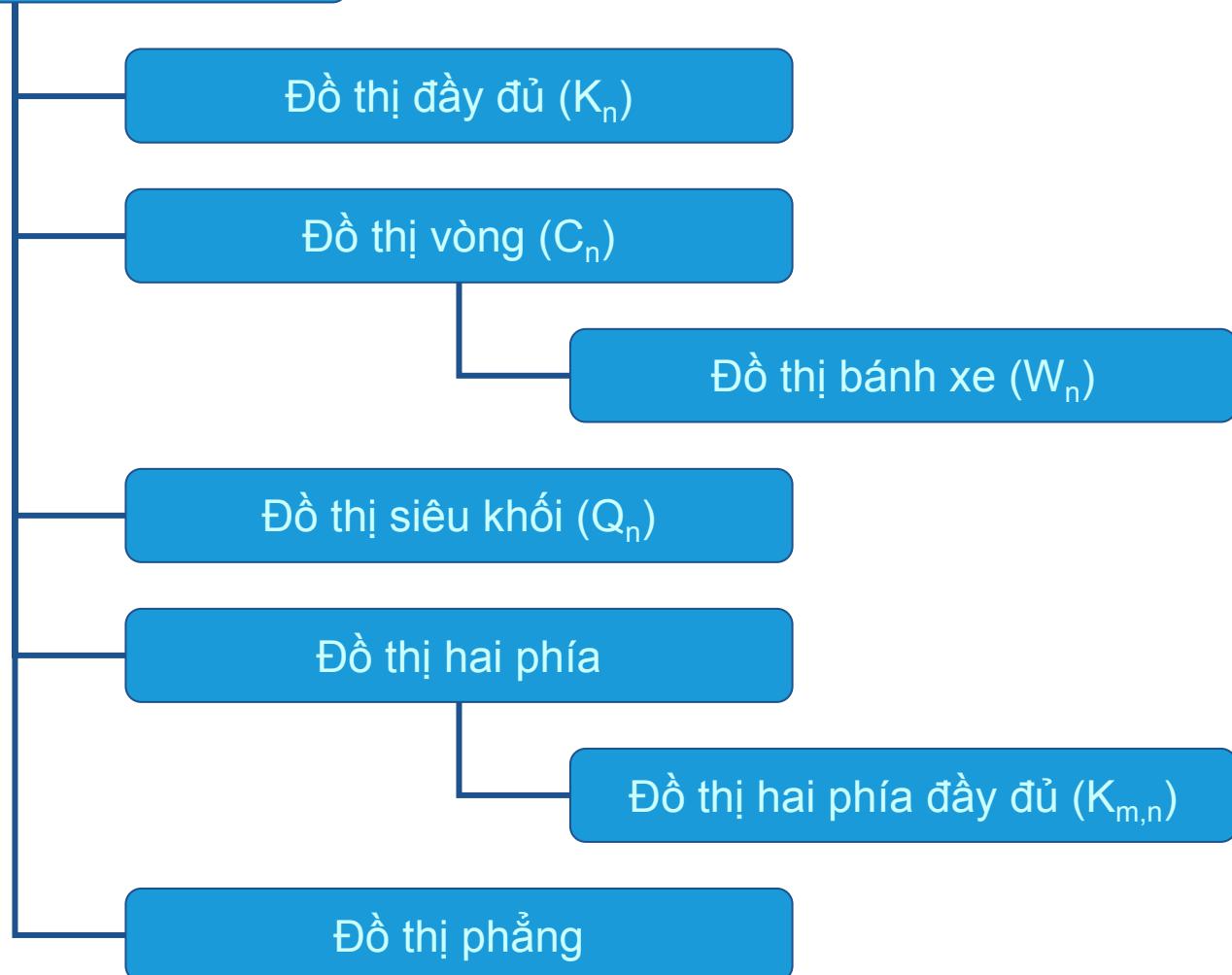
# VI. Một số dạng đồ thị đặc biệt

## ❖ Định lý Kuratovski

*Điều kiện cần và đủ để một đồ thị là phẳng là đồ thị này không chứa bất kỳ một đồ thi con nào đồng cấu với  $K_{3,3}$  và  $K_5$*

# VI. Một số dạng đồ thị đặc biệt

Các dạng đồ thị đặc biệt



# VI. Một số dạng đồ thị đặc biệt

## ❖ Bài tập

1. Số cạnh của đồ thị  $K_8$  ?
2. Số cạnh của đồ thị  $C_{2007}$  ?
3. Số cạnh của đồ thị  $W_{100}$  ?
4. Cho đồ thị  $G$  phẳng, liên thông có 20 đỉnh, bậc của mỗi đỉnh bằng 3. Đồ thị biểu diễn phẳng của  $G$  có bao nhiêu mặt?
5. Cho đồ thị phân đôi p đỉnh và q cạnh. CM:  
$$q \leq p^2/4$$
. Dấu = xảy ra khi nào?
6. Cho đồ thị  $G$  có n đỉnh, m cạnh với  $m \geq n$ . Chứng minh  $G$  có một chu trình.
7. Có bao nhiêu đồ thị đơn gồm 5 đỉnh và có 4 hoặc 6 cạnh ?

## Chương 2: Biểu diễn đồ thị

# Nội dung



**I. Các cách biểu diễn đồ thị**

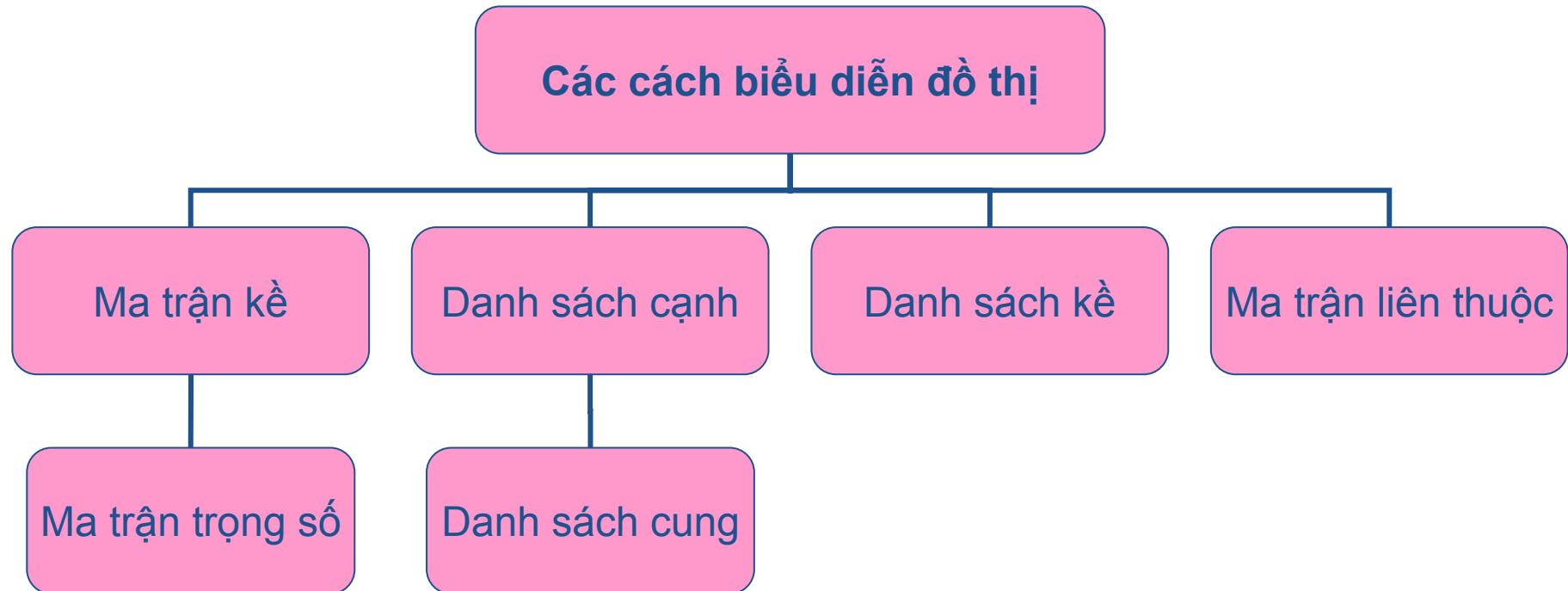


**II. Sự đẳng cấu của các đồ thị**



**III. Hướng dẫn cài đặt**

# I. Các cách biểu diễn đồ thị

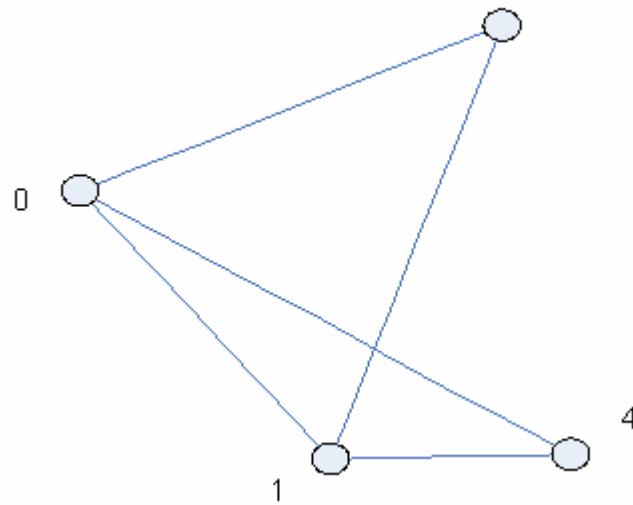


# I.1. Ma trận kề (đơn đồ thị vô hướng)

## ❖ Định nghĩa

- Đơn đồ thị  $G = (V, E)$  với tập đỉnh  $V = \{0, \dots, n-1\}$ , tập cạnh  $E = \{e_0, e_1, \dots, e_{m-1}\}$ . Ta gọi ma trận kề của  $G$  là
- $A = \{a_{i,j}, i, j = 0, \dots, n-1\}$ , với:

$$a_{i,j} = \begin{cases} 0, & \text{if } (i, j) \notin E \\ 1, & \text{if } (i, j) \in E \end{cases}$$



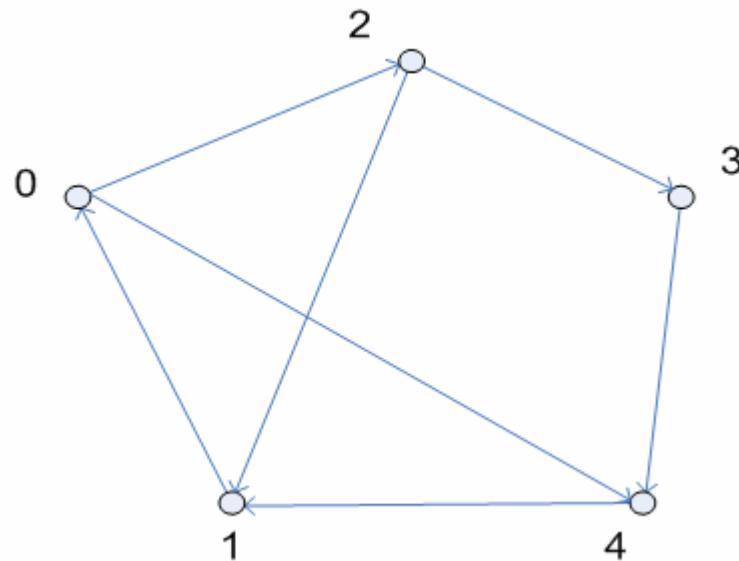
0	1	2	3	4	
0	0	1	1	0	1
1	1	0	1	0	1
2	1	1	0	0	0
3	0	0	0	0	0
4	1	1	0	0	0

# I.1. Ma trận kề (đơn đồ thị có hướng)

## ❖ Định nghĩa

- Giống đơn đồ thị có hướng
- E là tập các **cung**

$$a_{i,j} = \begin{cases} 0, & \text{if } (i, j) \notin E \\ 1, & \text{if } (i, j) \in E \end{cases}$$

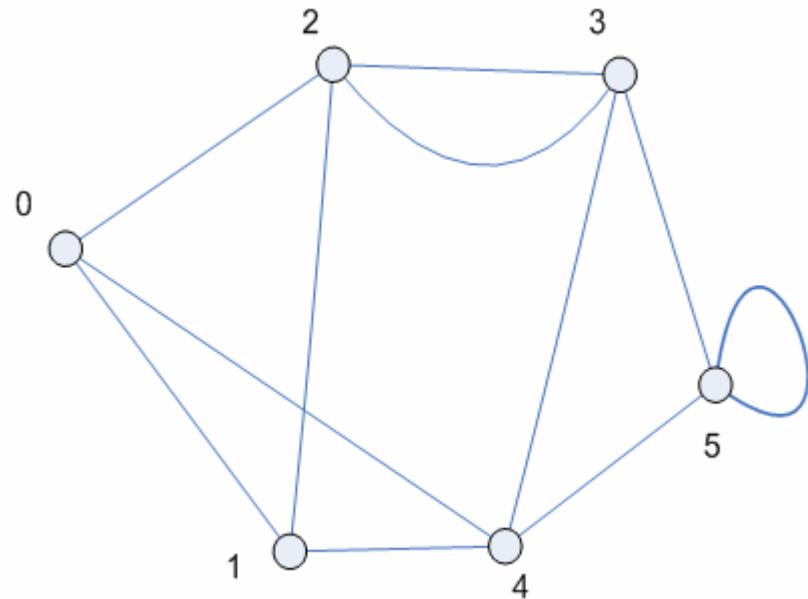


0	1	2	3	4
0	0	1	0	1
1	1	0	0	0
2	0	1	0	1
3	0	0	0	1
4	0	1	0	0

# I.1. Ma trận kề (Đa đồ thị)

## ❖ Định nghĩa

- E là tập các **cạnh/cung**
- $A_{i,j}$  là **số cạnh** nối đỉnh i và đỉnh j



	0	1	2	3	4	5
0	0	1	1	0	1	0
1	1	0	1	0	1	0
2	1	1	0	2	0	0
3	0	0	2	0	1	1
4	1	1	0	1	0	1
5	0	0	0	1	1	1

## I.1. Ma trận kề (Đa đồ thị)

### ❖ Một số tính chất của ma trận kề

- Ma trận kề của đồ thị vô hướng là đối xứng  $a[i,j] = a[j,i]$ . Ngược lại, ma trận đối xứng  $(0,1)$ , có đường chéo chính bằng  $0$ , bậc  $n$  sẽ tương ứng với đơn đồ thị vô hướng  $n$  đỉnh.

- **Nếu đồ thị vô hướng:**

Tổng dòng thứ  $i$  = Tổng cột thứ  $i$  =  $\deg(i)$

- **Nếu đồ thị có hướng:**

Tổng dòng  $i$  =  $\deg^+(i)$ , Tổng cột  $i$  =  $\deg^-(i)$



Ưu điểm và hạn chế  
của ma trận kề?

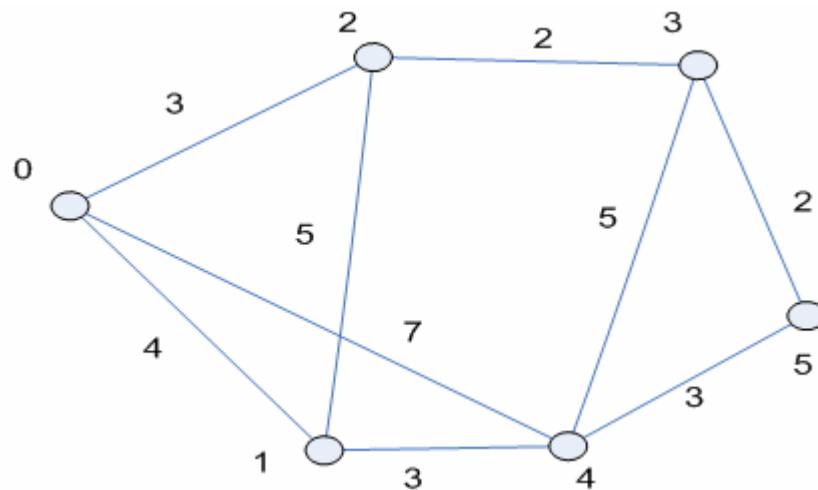
## I.2. Ma trận trọng số (đơn đồ thị)

### ❖ Định nghĩa

- Đơn đồ thị  $G = (V, E)$  với tập đỉnh  $V = \{0, \dots, n-1\}$ , tập cạnh  $E = \{e_0, e_1, \dots, e_{m-1}\}$ .
- Ta gọi **ma trận kè trọng số** của  $G$  là
  - $A = \{a_{i,j}, i, j = 0, \dots, n-1\}$ , với:

$$a_{i,j} = \begin{cases} b, & \text{if } (i, j) \notin E \\ c_k, & \text{if } (i, j) \in E \end{cases}$$

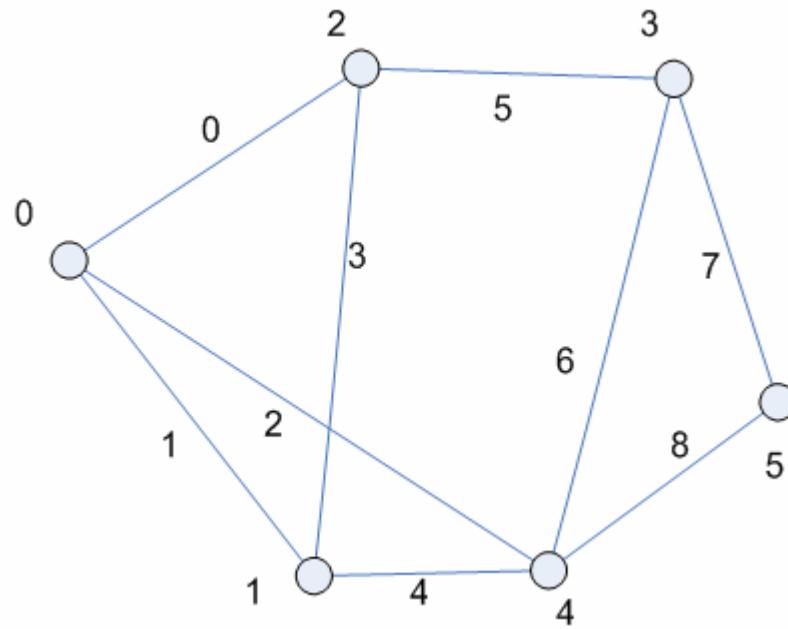
$c_k$  là một giá trị nào đó được quy định trước ( $0, -1, \infty, -\infty, \dots$ )



0	1	2	3	4	5
0	0	4	3	0	7
1	4	0	5	0	3
2	3	5	0	2	0
3	0	0	2	0	5
4	7	3	0	5	0
5	0	0	0	2	3

### I.3. Danh sách cạnh

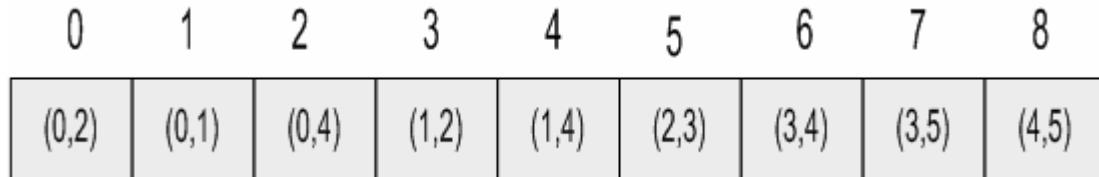
- Đối với các đồ thị thưa n đỉnh, m cạnh ( $m < 6n$ ) người ta thường dùng cách biểu diễn danh sách cạnh để tiết kiệm không gian lưu trữ
- Lưu các cạnh  $e=(u, v)$  của đồ thị trong một danh sách
- Danh sách có thể được cài đặt bằng **mảng 1 chiều** hoặc **danh sách liên kết**.



Cạnh	Đầu 1	Đầu 2
0	0	2
1	0	1
2	0	4
3	1	2
4	1	4
5	2	3
6	3	4
7	3	5
8	4	5

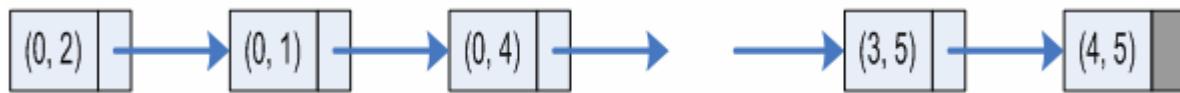
# I.3. Danh sách cạnh

Cài đặt bằng mảng 1 chiều



Cạnh	Đầu 1	Đầu 2
0	0	2
1	0	1
2	0	4
3	1	2
4	1	4
5	2	3
6	3	4
7	3	5
8	4	5

Cài đặt bằng danh sách liên kết



typedef struct tagNode

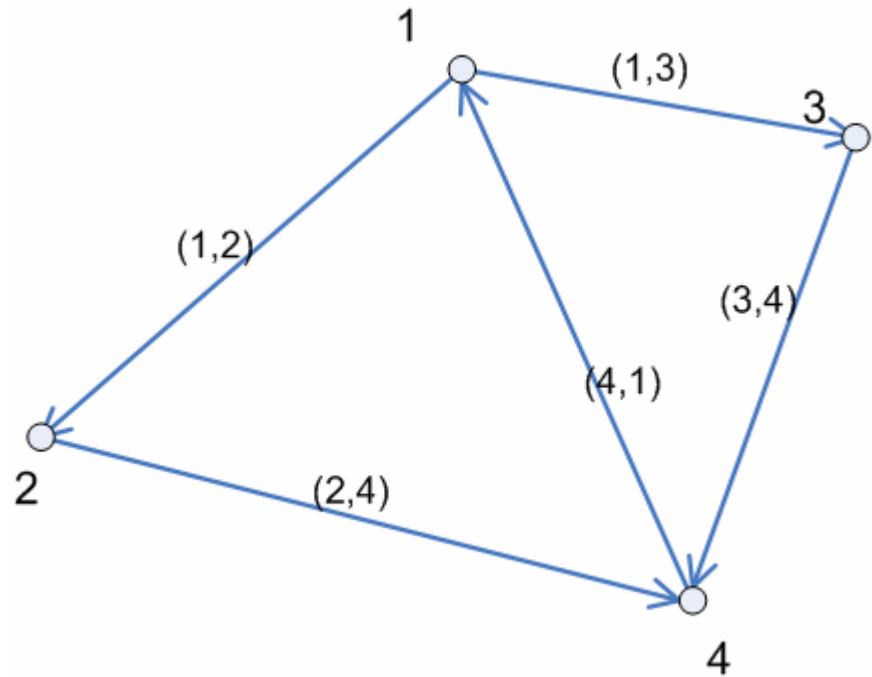
{

    int diemdau1, diemdau2;

} Canh;

## I.4. Danh sách cung

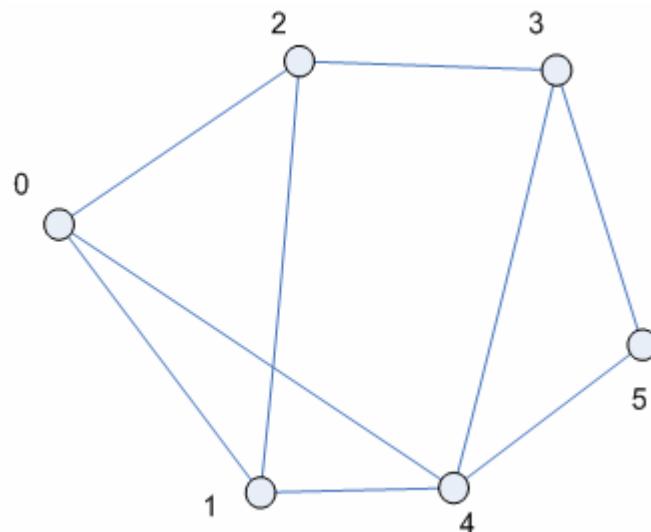
- Trong trường hợp đồ thị có hướng thì mỗi phần tử của danh sách (gọi là **danh sách cung**) là một cung  $e=(u, v)$ . Trong đó  $u$  là **đỉnh đầu**,  $v$  là **đỉnh cuối** của cung.



Cạnh	Đầu 1	Đầu 2
(1,2)	1	2
(4,1)	4	1
(1,3)	1	3
(2,4)	2	4
(3,4)	3	4

## I.4. Danh sách kề

- Tương ứng với mỗi đỉnh  $v$  của đồ thị, ta có tương ứng một danh sách để lưu các đỉnh kề với nó.
- Danh sách: mảng 1 chiều, hoặc danh sách liên kết



Đỉnh V	Các cạnh kề
0	1, 2, 4
1	0, 2, 4
2	0, 1, 3
3	2, 4, 5
4	0, 1, 3, 5
5	3, 4

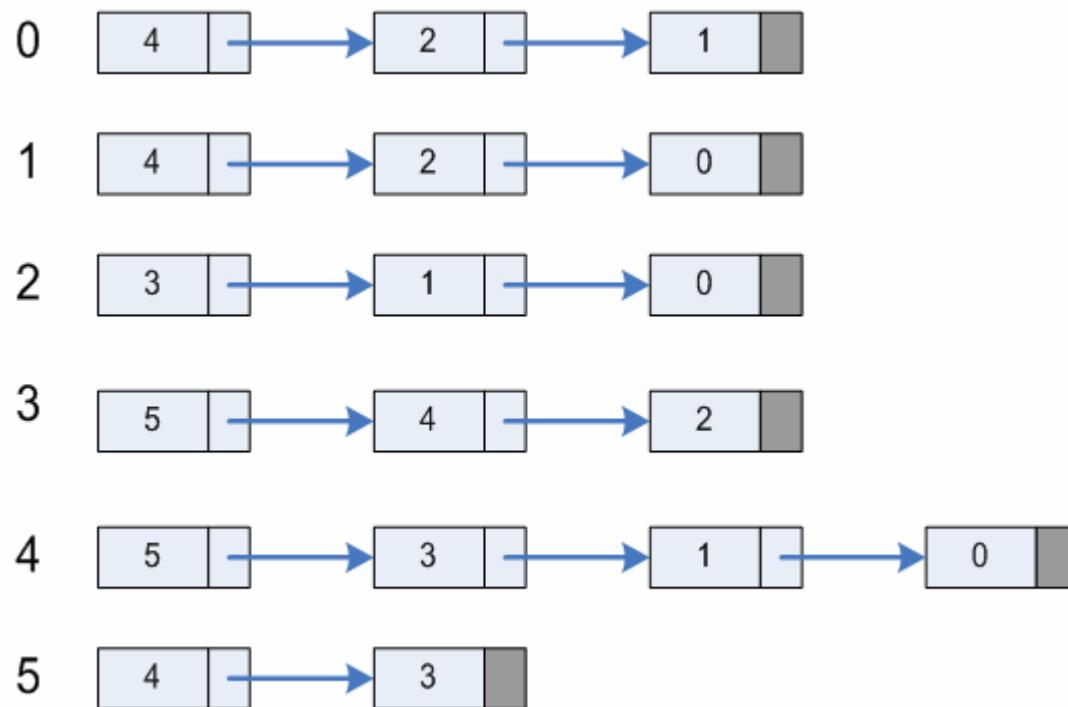
Cài đặt bằng mảng:

$Ke[] = \{1, 2, 4, 0, 2, 4, 0, 1, 3, 2, 4, 5, 0, 1, 3, 5, 3, 4\}$

$ViTri[] = \{0, 3, 6, 9, 12, 16\}$

## I.4. Danh sách kề

- Cài đặt bằng danh sách kề liên kết



Đỉnh V	Các cạnh kề
0	1, 2, 4
1	0, 2, 4
2	0, 1, 3
3	2, 4, 5
4	0, 1, 3, 5
5	3, 4

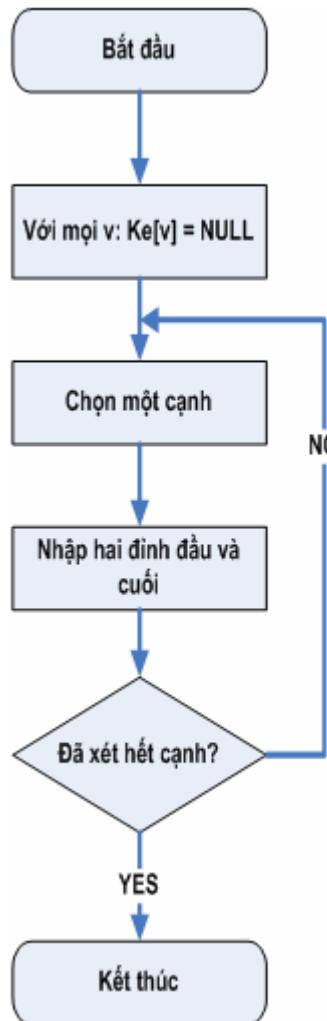
## I.4. Danh sách kề

- Thuật toán xây dựng danh sách kề liên kết

```
# include <iostream.h>
# include <stdlib.h>
const maxV = 99;
typedef struct Node {
    int v;
    struct Node*next;
}node;
int j, x, y, m, n, v ;
node *p, *ke[maxV];
```

# I.4. Danh sách kề

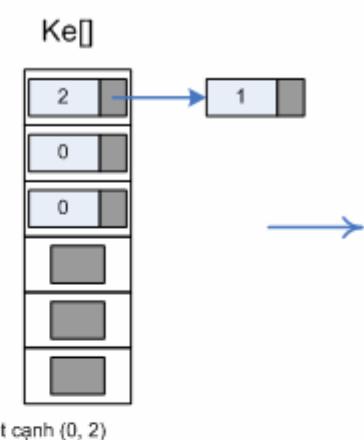
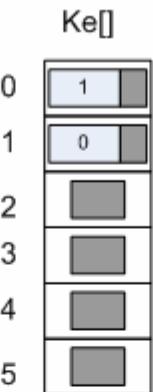
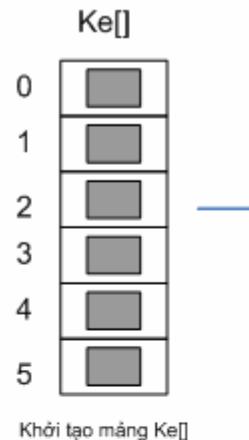
## ▪ Thuật toán xây dựng danh sách kề liên kết



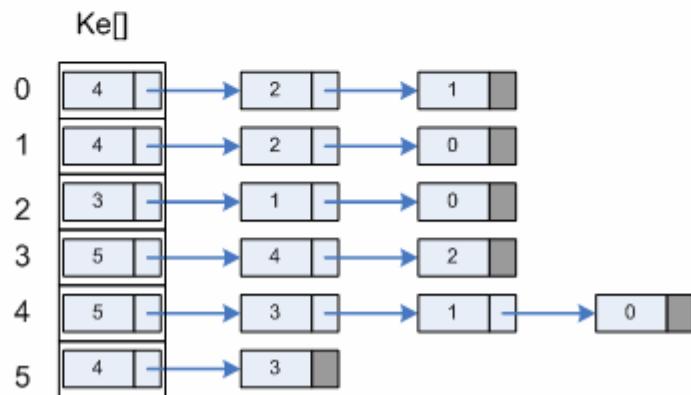
```
int main(int argc, char* argv[])
{
    cout<<"Cho so canh va so dinh cua do thi: ";
    cin>>m>>n;
    for(j=0;j<n;j++)
        ke[j]=NULL;
    for(j=1;j<=m;j++)
    {
        cout<<"Cho dinh dau, dinh cuoi cua canh "<<j<<":";
        cin>>x>>y;
        p = (node*)malloc(sizeof(node));
        p->v = x;
        p->next = ke[y];
        ke[y]=p;
        p = (node*)malloc(sizeof(node));
        p->v = y;
        p->next = ke[x];
        ke[x]=p;
    }
}
```

# I.4. Danh sách kề

## ▪ Ví dụ



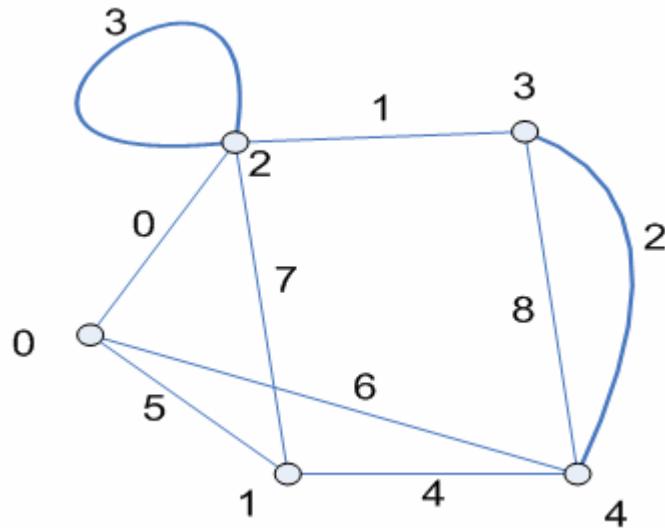
Đỉnh V	Các cạnh kề
0	1, 2, 4
1	0, 2, 4
2	0, 1, 3
3	2, 4, 5
4	0, 1, 3, 5
5	3, 4



# I.5. Ma trận liên thuộc (đồ thị vô hướng)

## ❖ Định nghĩa

- Đồ thị vô hướng  $G=(V, E)$ . Tập đỉnh  $V=\{0, 1, 2, \dots, n-1\}$ . Tập cạnh  $E=\{e_1, e_2, \dots, e_{m-1}\}$ . Ta gọi ma trận liên thuộc của  $G$  là  $B = \{b_{i,j}, i = 0, \dots, n-1, j = 0, \dots, m-1\}$ . Trong đó
  - $b_{i,j} = 1$  nếu đỉnh  $i$  kề cạnh  $j$
  - $b_{i,j} = 0$  nếu đỉnh  $i$  không kề cạnh  $j$

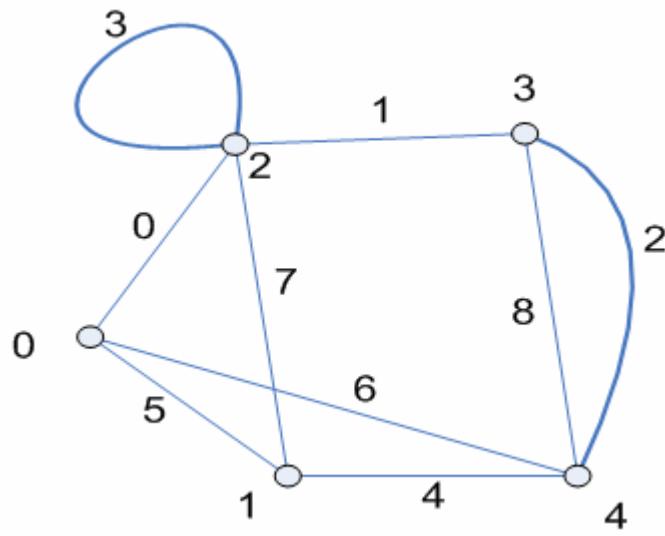


	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
<b>0</b>	1	0	0	0	0	1	1	0	0
<b>1</b>	0	0	0	0	1	1	0	1	0
<b>2</b>	1	1	0	1	0	0	0	1	0
<b>3</b>	0	1	1	0	0	0	0	0	1
<b>4</b>	0	0	1	0	1	0	1	0	1

# I.5. Ma trận liên thuộc (đồ thị vô hướng)

## ❖ Tính chất

- Mỗi cột chứa đúng hai số 1 chỉ hai đầu của cạnh tương ứng với đỉnh ứng với cột đó. Cột ứng với khuyên chứa đúng một số 1.
- Các cột ứng với các cạnh lặp thì giống nhau.
- Nếu đồ thị không có khuyên thì tổng hàng  $i$  là bậc của đỉnh  $v_i$ .

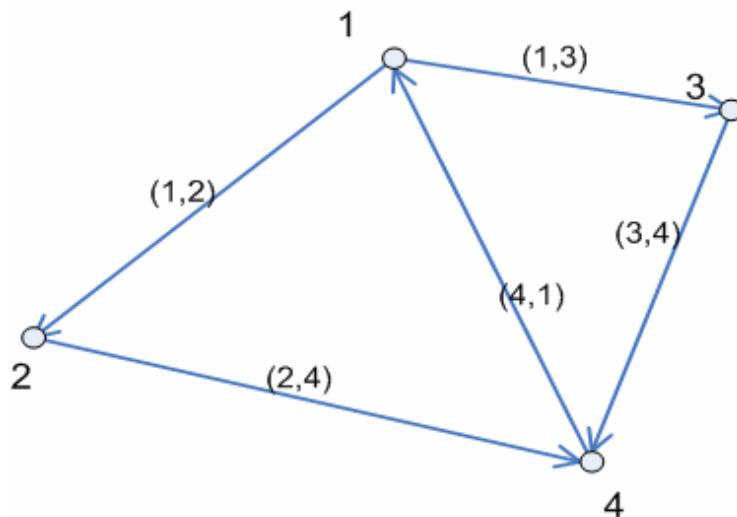


	0	1	2	3	4	5	6	7	8
0	1	0	0	0	0	1	1	0	0
1	0	0	0	0	1	1	1	0	1
2	1	1	0	1	0	0	0	1	0
3	0	1	1	0	0	0	0	0	1
4	0	0	1	0	1	0	1	0	1

# I.5. Ma trận liên thuộc (đồ thị có hướng)

## ❖ Định nghĩa

- Đơn đồ thị có hướng  $G=(V, E)$ . Tập đỉnh  $V=\{0, 1, 2, \dots, n-1\}$ . Tập cung  $E=\{e_1, e_2, \dots, e_{m-1}\}$ .  
Ta gọi ma trận liên thuộc của  $G$  là  $B = \{b_{i,j}, i = \underline{0, \dots, n-1}, j = \underline{0, \dots, m-1}\}$ .  
Trong đó
  - $b_{i,j} = 1$  nếu đỉnh  $i$  là đỉnh đầu của cung  $j$
  - $b_{i,j} = -1$  nếu đỉnh  $i$  là đỉnh cuối của cung  $j$
  - $b_{i,j} = 0$  nếu đỉnh  $i$  không là đầu mút của cung  $j$



	(1,2)	(4,1)	(1,3)	(3,4)	(2,4)
1	1	-1	1	0	0
2	-1	0	0	0	1
3	0	0	-1	1	0
4	0	1	0	-1	-1

# I. Các cách biểu diễn đồ thị

## Các cách biểu diễn đồ thị

Ma trận kề

Danh sách cạnh

Danh sách kề

Ma trận liên thuộc

- ☞  $n^2$  Đơn vị bộ nhớ
- ☞ Dễ kiểm tra đ/k kề nhau

- ☞  $2m$  Đơn vị bộ nhớ
- ☞ Đồ thị thừa
- ☞ Khó kiểm tra đ/k kề nhau

- ☞  $2m+n$  Đơn vị bộ nhớ
- ☞ Dễ dàng việc thêm bớt các cạnh, đỉnh

- ☞  $m*n$  Đơn vị bộ nhớ
- ☞ Dễ dàng việc thêm bớt các cạnh, đỉnh

# Nội dung



I. Các cách biểu diễn đồ thị



II. Sự đẳng cấu của các đồ thị



III. Hướng dẫn cài đặt

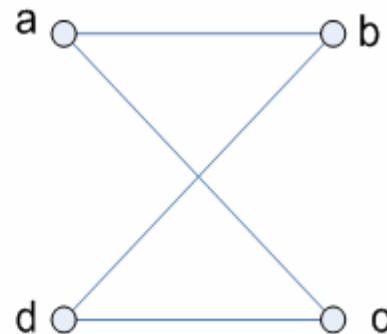
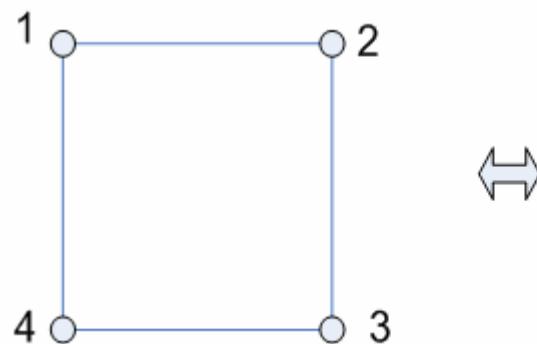
## II. SỰ ĐẲNG CẤU CỦA CÁC ĐỒ THỊ

### ❖ Định nghĩa

- Các đồ thị đơn  $G_1 = (V_1, E_1)$  và  $G_2 = (V_2, E_2)$  là đẳng cấu nếu có hàm **song ánh**:

$f : V_1 \rightarrow V_2$  sao cho  $\forall$  đỉnh  $a$  &  $b$  kề trong  $G_1 \Leftrightarrow f(a)$  &  $f(b)$  kề trong  $G_2$ .

- ➔ Tồn tại một phép tương ứng một – một giữa các đỉnh của hai đồ thị đồng thời đảm bảo quan hệ liền kề.

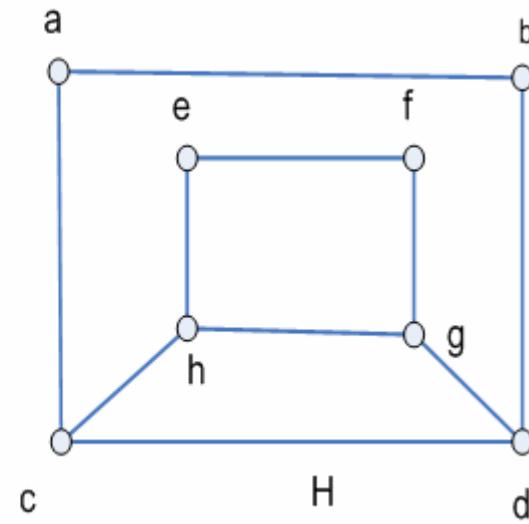
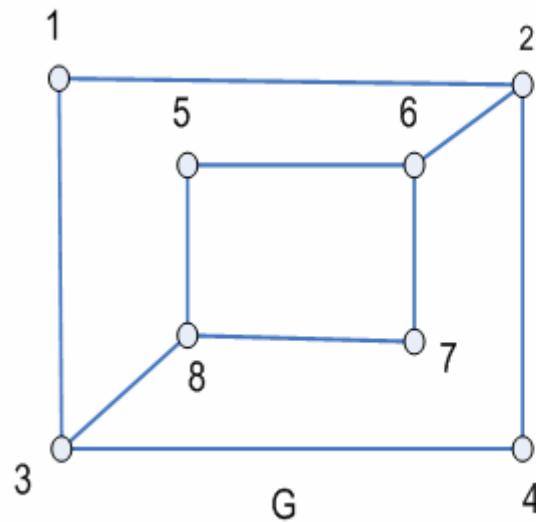


$$\begin{aligned}f(1) &= a, f(2) = b \\f(3) &= d, f(4) = b\end{aligned}$$

## II. SỰ ĐẲNG CẤU CỦA CÁC ĐỒ THỊ

### ❖ Tính bất biến

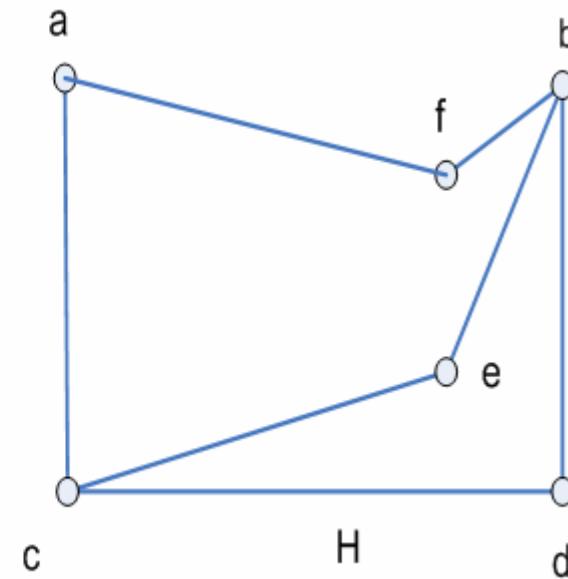
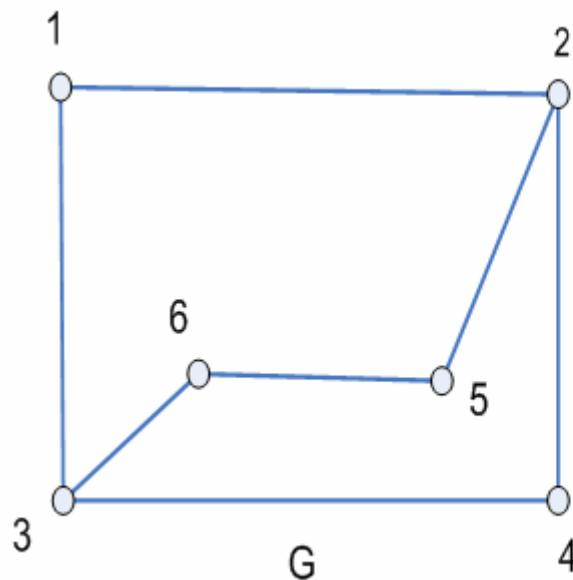
- Hai đồ thị đẳng cấu bất kỳ có **tính chất giống nhau** (số đỉnh, số cạnh, bậc của một đỉnh,...). Người ta gọi đó là tính bất biến trong các đồ thị đẳng cấu.



## II. SỰ ĐẲNG CẤU CỦA CÁC ĐỒ THỊ

### ❖ Chứng minh 2 đồ thị là đẳng cấu

- Tìm một ánh xạ  $f$  tương ứng một – một giữa các đỉnh
- So sánh 2 ma trận liền kề tạo ra dựa trên ánh xạ  $f$



# Nội dung



**I. Các cách biểu diễn đồ thị**



**II. Sự đẳng cấu của các đồ thị**



**III. Hướng dẫn cài đặt**

### III. Hướng dẫn cài đặt

- Khai báo file
- Kết nối biến file với tên thực của file ở trên đĩa (floppy or hard disk)
- Mở file, đóng file
- Đọc thông tin từ file và ghi thông tin vào file
- Để hiểu tốt danh sách kè liên kết cần tham khảo phần biến con trỏ trong các tài liệu về lập trình.

## Chương 3: Tìm kiếm trên đồ thị

# Nội dung



**I. Duyệt đồ thị theo chiều sâu**



**II. Duyệt đồ thị theo chiều rộng**



**III. Tìm đường đi**



**IV. Kiểm tra tính liên thông**

# I. Duyệt đồ thị theo chiều sâu

## ❖ Giới thiệu

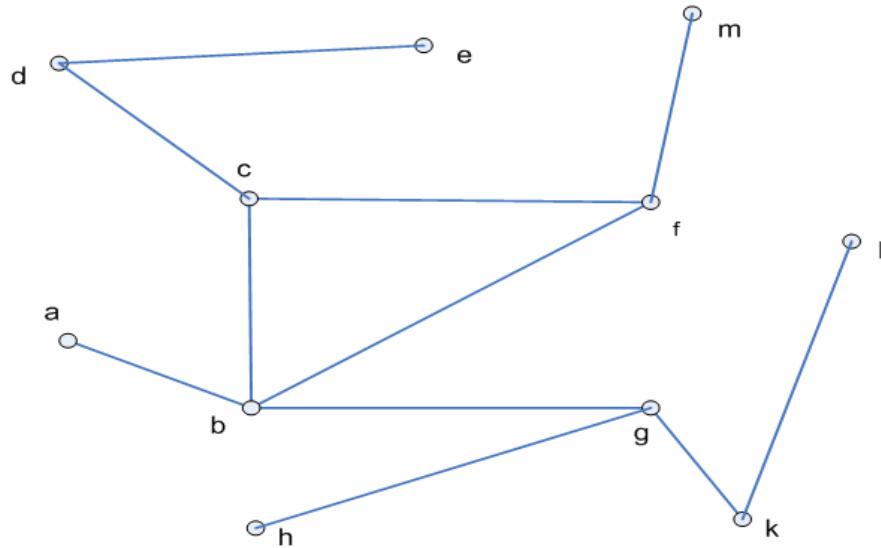
- Duyệt đồ thị là quá trình đi qua tất cả các đỉnh của đồ thị sao cho mỗi đỉnh của nó được viếng thăm đúng một lần.
- Duyệt theo chiều sâu (Depth First Search – DFS)
- Duyệt theo chiều rộng (Breadth First Search – BFS)

# I. Duyệt đồ thị theo chiều sâu

## ❖ Nguyên lý

- Bắt đầu tìm kiếm từ một đỉnh  $v$  nào đó của đồ thị.
- Sau đó chọn  $u$  là một đỉnh tùy ý kề với  $v$  (với đồ thị có hướng thì  $u$  là đỉnh sau,  $v$  là đỉnh đầu của cung  $uv$ )
- Lặp lại quá trình này với  $u$  cho đến khi không tìm được đỉnh kề tiếp theo nữa thì trở về **đỉnh ngay trước đỉnh mà không thể đi tiếp** để tìm qua nhánh khác.

# I. Duyệt đồ thị theo chiều sâu



Thứ tự duyệt:

d c b a

g k l

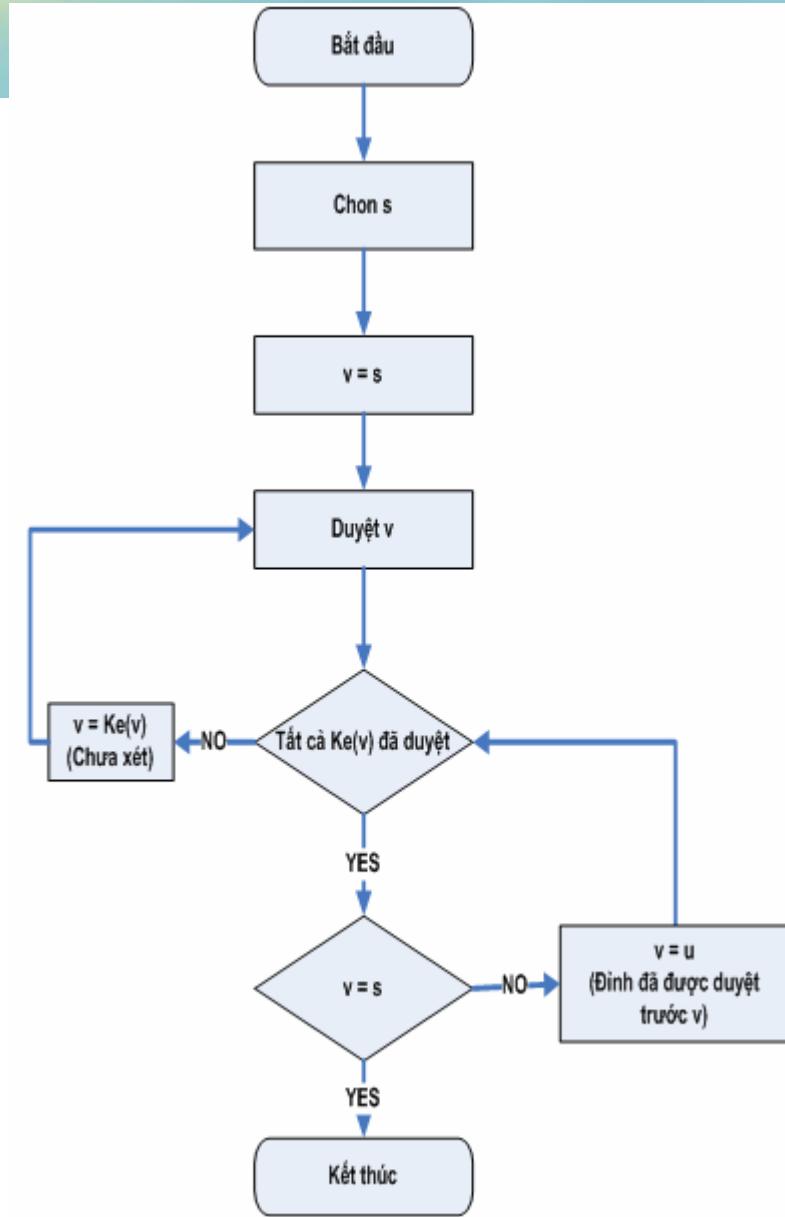
h

f m

e

# I.1. Cài đặt đệ quy

- B1:** Lấy  $s$  là một đỉnh của đồ thị
- B2:** Đặt  $v = s$
- B3:** Duyệt đỉnh  $v$
- B4:** Nếu  $\forall$  đỉnh kề của  $v$  đều được duyệt, đặt  $v=$ đỉnh đã được duyệt trước đỉnh  $v$ , Nếu  $v = s$  thì đi đến Bước 6, ngược lại trở lại Bước 3.
- B5:** Chọn  $u$  là đỉnh kề chưa được duyệt của  $v$ , đặt  $v = u$ , trở lại Bước 3
- B6:** Kết thúc



## I.1. Cài đặt đệ quy

### ❖ Cài đặt bằng mã giả

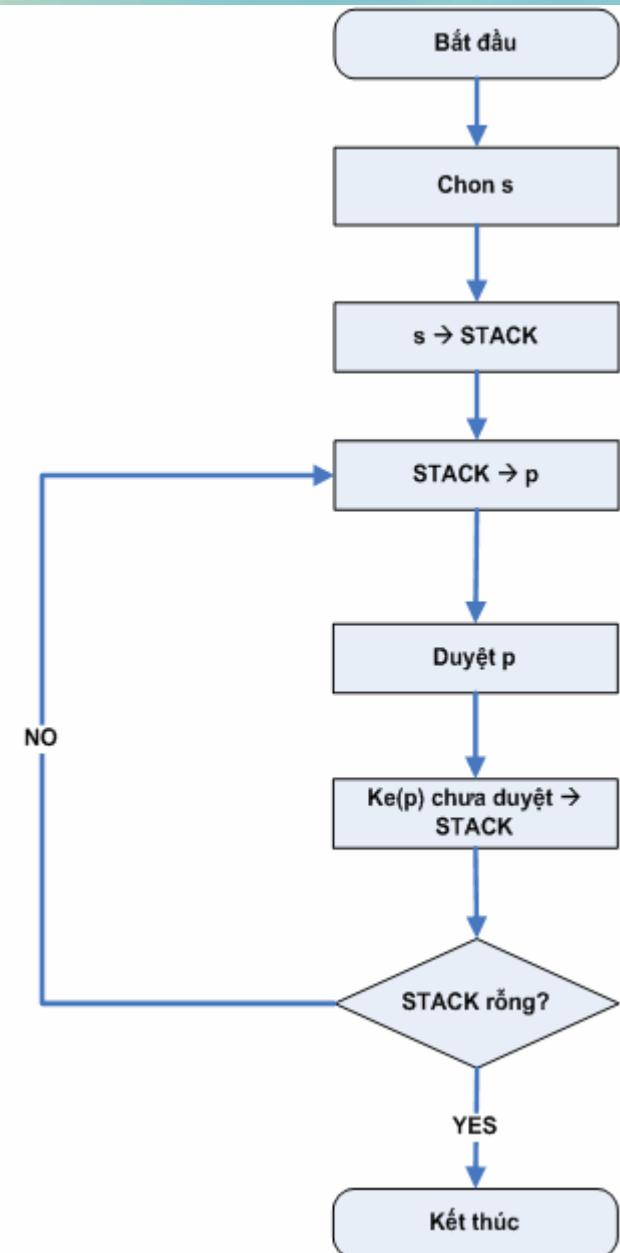
```
/* Khai báo các biến ChuaXet, Ke */
DFS(v)
{
    Duyệt đỉnh (v);
    ChuaXet[v] = 0; /*Đánh dấu đã xét đỉnh v*/
    for ( u ∈ Ke(v) )
        if ( ChuaXet[u] ) DFS(u);
};

void main()
{
/* Nhập đồ thị, tạo mảng Ke */
for (v ∈ V) ChuaXet[v] = 1; /*Khởi tạo cờ cho đỉnh */
for (v ∈ V)
    if ( ChuaXet[v] ) DFS(v);
}
```

## I.2. Cài đặt không đệ quy

### ❖ Thuật toán

- B1: Lấy s là một đỉnh của đồ thi
- B2: Đặt s vào STACK
- B3: Nếu STACK rỗng đi đến 7.
- B4: Lấy đỉnh p từ STACK
- B5: Duyệt đỉnh p
- B6: Đặt các đỉnh kề của p chưa được xét (*chưa từng có mặt trong STACK*) vào STACK, trở lại 3.
- B7: Kết thúc.

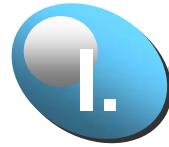


# I.Duyệt đồ thị theo chiều sâu

## ❖ Ý nghĩa

- Kiểm tra đường đi giữa 2 đỉnh
- Chia đồ thị thành các thành phần liên thông
- Xây dựng cây khung của đồ thị
- Kiểm tra xem đồ thị có chu trình hay không

# Nội dung



**I. Duyệt đồ thị theo chiều sâu**



**II. Duyệt đồ thị theo chiều rộng**



**III. Tìm đường đi**



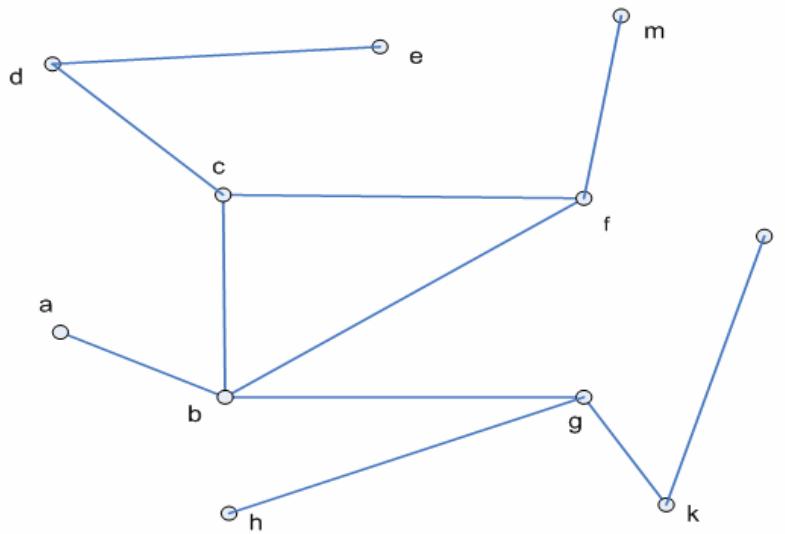
**IV. Kiểm tra tính liên thông**

## II. Duyệt đồ thị theo chiều rộng

### ❖ Nguyên lý

- Bắt đầu từ một đỉnh v bất kỳ.
- Duyệt tất cả những đỉnh kề của v lưu vào một tập  $T(u)$  (với đồ thị có hướng thì  $T(u)$  là tập các đỉnh u với u là đỉnh sau, v là đỉnh đầu của cung uv).
- Sau đó tiếp tục xét các đỉnh **u thuộc  $T(u)$**  và áp dụng lại cách duyệt giống như với v.

## II. Duyệt đồ thị theo chiều rộng

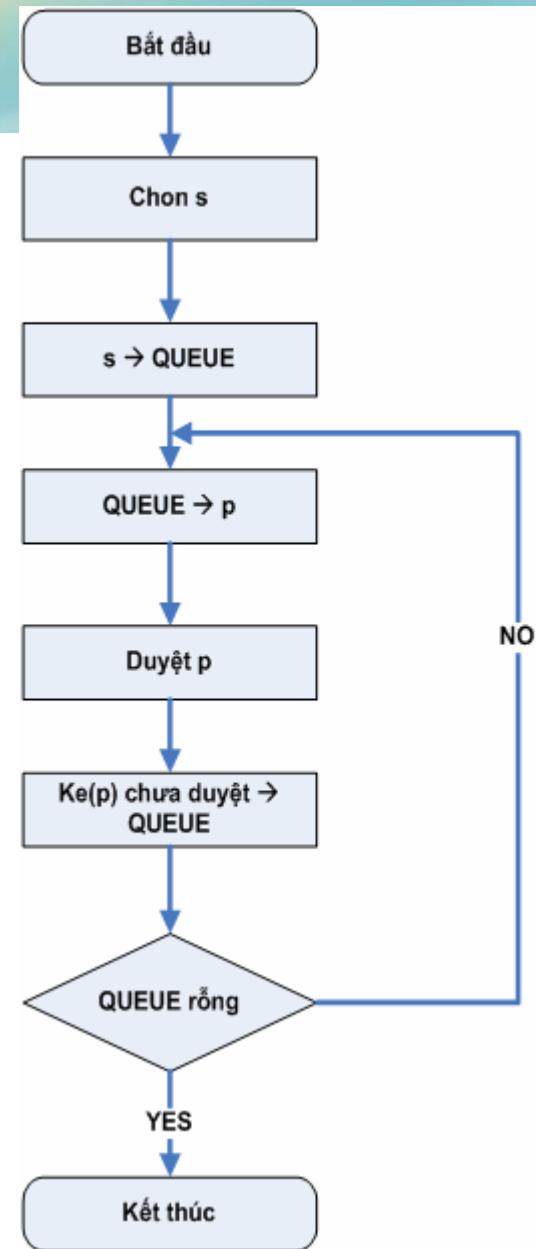


Thứ tự duyệt:

d  
e c  
b f  
a g m  
h k  
l

## II.1. Cài đặt bằng hàng đợi

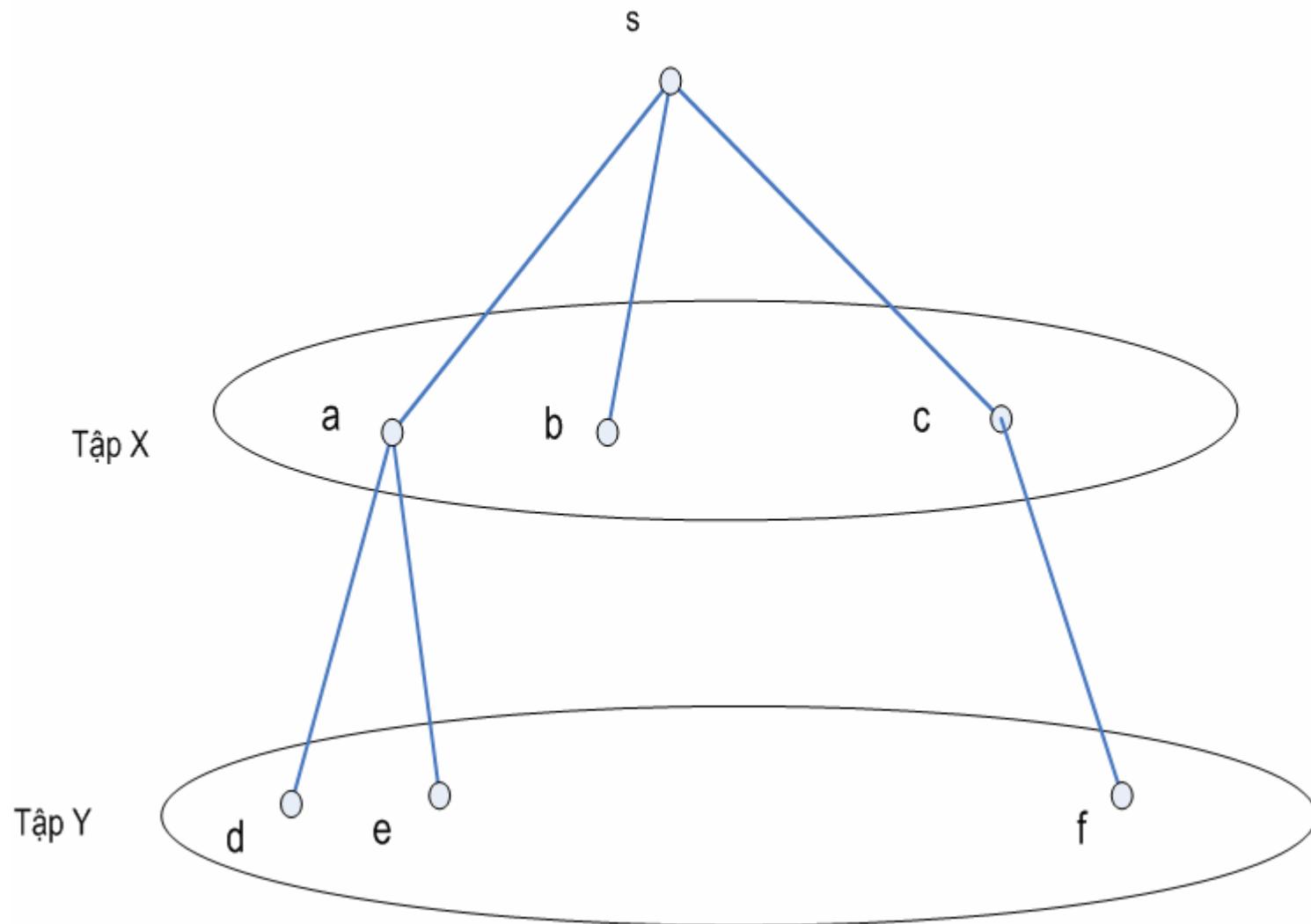
- B1: Lấy  $s$  là một đỉnh của đồ thị
- B2: Đặt  $s$  vào QUEUE
- B3: Lặp** nếu QUEUE chưa rỗng.
- Lấy đỉnh  $p$  từ QUEUE
  - Duyệt đỉnh  $p$
  - Đặt các đỉnh kề của  $p$  chưa được xét (*chưa từng có mặt trong QUEUE*) vào QUEUE.
  - Kết thúc lặp



## II.1. Cài đặt bằng hàng đợi

```
/* Khai báo các biến ChuaXet, Ke */
BFS(v)
{
    QUEUE = Ø;
    QUEUE ← v;
    ChuaXet[v] = 0; /* Đánh dấu đã xét đỉnh v */
    while ( QUEUE ≠ Ø )
    {
        p ← QUEUE;
        Duyệt đỉnh p;
        for ( u ∈ Ke(p) )
            if ( ChuaXet[u] )
            {
                QUEUE ← u;
                ChuaXet[u] = 0; /* Đánh dấu đã xét đỉnh */
            }
    }
}
void main()
/* Nhập đồ thị, tạo biến Ke */
{
    for ( v ∈ V ) ChuaXet[v] = 1; /* Khởi tạo cờ cho đỉnh */
    for ( v ∈ V )
        if ( ChuaXet[v] ) BFS(v);
}
```

## II.2. Cài đặt bằng thuật toán loang



## II.2. Cài đặt bằng thuật toán loang

### ❖ Bước 1: Khởi tạo

- Bắt đầu từ đỉnh s. Đánh dấu đỉnh s, các đỉnh khác s đều chưa bị đánh dấu
- $X = \{s\}$ ,  $Y = \emptyset$

### ❖ Bước 2: Lặp lại cho đến khi $X = \emptyset$

- Gán  $Y = \emptyset$ .
- Với mọi đỉnh  $u \in X$ 
  - Xét tất cả các đỉnh  $v$  kề với  $u$  mà chưa bị đánh dấu. Với mỗi đỉnh đó:
    - Đánh dấu  $v$
    - Lưu đường đi, đỉnh liền trước  $v$  trong đường đi từ  $s \rightarrow v$  là  $u$ .
    - Đưa  $v$  vào tập  $Y$
- Gán  $X = Y$

## II. Duyệt đồ thị theo chiều rộng

### ❖ Ý nghĩa

- Kiểm tra đường đi giữa 2 đỉnh
- Chia đồ thị thành các thành phần liên thông
- Xây dựng cây khung của đồ thị
- Tìm đường đi ngắn nhất từ 1 đỉnh đến các đỉnh còn lại

# Nội dung



**Duyệt đồ thị theo chiều sâu**



**Duyệt đồ thị theo chiều rộng**



**Tìm đường đi**



**Kiểm tra tính liên thông**

### III. Tìm đường đi

#### ❖ Bài toán

- Cho đồ thị G, s và t là hai đỉnh tùy ý của đồ thị. Hãy tìm đường đi từ s đến t.

#### ❖ Phương pháp

- Bắt đầu từ đỉnh s, Sử dụng DFS hoặc BFS để duyệt đồ thị.
  - Tìm thấy  $ChuaXet(t) = 0$
  - Không tìm thấy  $ChuaXet(t) = 1$
- Sử dụng thêm mảng **Truoc[]** để lưu vết

### III.1. Tìm đường đi theo chiều sâu

```
/* Khai báo các biến ChuaXet, Ke */
DFS(v);
{
    Duyệt đỉnh (v);
    ChuaXet[v] = 0;
    for ( u ∈ Ke(v) )
        if ( ChuaXet[u] )
            {
                Truoc[u] = v; /* Lưu vết*/
                DFS(u);
            }
}
main() // Nhập đồ thị, tạo biến Ke
{
    for ( v ∈ V ) ChuaXet[v] = 1; // Khởi tạo cờ cho đỉnh
    DFS(s);
}
```

## III.2. Tìm đường đi theo chiều rộng

```
/* Khai báo các biến ChuaXet, Ke , QUEUE */
BFS(v);
{
    QUEUE = ∅; QUEUE ← v; ChuaXet[v] = 0;
    while ( QUEUE ≠ ∅ )
    {
        p ← QUEUE;
        Duyệt đỉnh p;
        for ( u ∈ Ke(p) )
            if ( ChuaXet[u] )
            {
                QUEUE ← u;
                ChuaXet[u] = 0;
                Truoc[u] = p; /*Lưu vết*/
            }
    }
}
main() // Nhập đồ thị, tạo biến Ke
{
    for ( v ∈ V ) ChuaXet[v] = 1; // Khởi tạo cờ cho đỉnh
    BFS(s);
}
```

## III.2. Tìm đường đi theo chiều rộng

Khôi phục đường đi từ s đến t  
 $s \rightarrow x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_n \rightarrow t$

### Cài đặt:

```
v = t;  
while (v != s)  
{  
    printf (v);  
    v = Truoc[v];  
}
```

# Nội dung



**I. Duyệt đồ thị theo chiều sâu**



**II. Duyệt đồ thị theo chiều rộng**



**III. Tìm đường đi**



**IV. Kiểm tra tính liên thông**

## IV. Kiểm tra tính liên thông

### ❖ Bài toán

- Tính số thành phần liên thông của đồ thị, và xác định những đỉnh thuộc cùng một thành phần liên thông.

### ❖ Phương pháp

- Sử dụng DFS và BFS
- Biến **inconnect** đếm số thành phần liên thông của đồ thị.
- Mảng **index[]** lưu chỉ số của các thành phần liên thông.

## IV.1. Tìm theo chiều sâu

```
/* Khai báo các biến ChuaXet, Ke, index*/
DFS(v);
{
    Duyệt đỉnh (v);
    index[v] = inconnect;
    ChuaXet[v] = 0;
    for ( u ∈ Ke(v) )
        if ( ChuaXet[u] ) DFS(u);
}
main()
{
    /* Nhập đồ thị, tạo biến Ke */
    for ( v ∈ V ) ChuaXet[v] = 1; /* Khởi tạo cờ cho đỉnh */
    inconnect = 0;
    for ( v ∈ V )
        if ( ChuaXet[v] )
        {
            inconnect++; DFS(v);
        }
}
```

## IV.2. Tìm theo chiều rộng

```
/* Khai báo các biến toàn cục ChuaXet, Ke, QUEUE, index */
BFS(v) {
    QUEUE = 0; QUEUE ← v; ChuaXet[v] = 0;
    while ( QUEUE ≠ 0 ) {
        p ← QUEUE; Duyệt đỉnh p;
        index[p] = inconnect;
        for ( u ∈ Ke(p) )
            if ( ChuaXet[u] ) {
                QUEUE ← u; ChuaXet[u] = 0;
            }
    }
}
main() {
    for ( v ∈ V ) ChuaXet[v] = 1;
    inconnect = 0;
    for ( v ∈ V )
        if ( ChuaXet[v] ) {
            inconnect + + ; BFS(v);
        }
}
```

## Chương 4: Đồ thị Euler và đồ thị Hamilton

# Nội dung



I.

Đồ thị Euler



II.

Đồ thị Hamilton

# I. Đồ thị Euler

## Đồ thị Euler

1. Định nghĩa

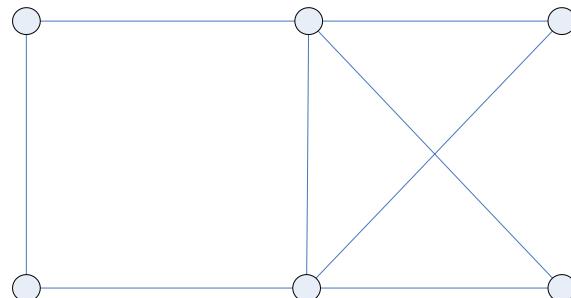
2. Định lý Euler

3. Giải thuật xây dựng chu trình Euler

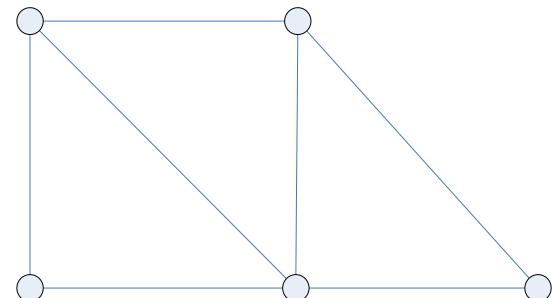
## I.1. Định nghĩa

❖ Giả sử  $G$  là đơn (đa) đồ thị vô (có) hướng:

- **Chu trình Euler** trong  $G$  là **chu trình đơn** đi qua tất cả các cạnh của đồ thị. Nếu  $G$  có chu trình Euler thì  $G$  được gọi là **đồ thị Euler**.
- **Đường đi Euler** trong  $G$  là **đường đi đơn** qua tất cả các cạnh của đồ thị. Nếu  $G$  có đường đi Euler thì  $G$  được gọi là **đồ thị nửa Euler**.



Đồ thị Euler



Đồ thị nửa Euler

## I.2. Định lý

### ❖ Định lý 1

- *Đồ thị vô hướng, liên thông  $G=(V, E)$  có chu trình Euler khi và chỉ khi **mọi đỉnh** của  $G$  đều có **bậc chẵn**.*

### ❖ Chứng minh

- $G$  có chu trình Euler  $\Rightarrow$  Mọi đỉnh đều bậc chẵn
- Mọi đỉnh đều bậc chẵn  $\Rightarrow G$  có chu trình Euler

## I.2. Định lý

### ❖ Bổ đề

- “Cho đồ thị  $G=(V, E)$ , nếu mọi đỉnh của  $G$  có  $\deg(u) \geq 2$  thì  $G$  có chu trình”

### ❖ Chứng minh ?

## I.2. Định lý

### ❖ Định lý 2:

- *Đồ thị vô hướng, liên thông  $G=(V, E)$  có đường đi Euler mà không có chu trình Euler khi và chỉ khi  $G$  có đúng hai đỉnh bậc lẻ.*
- *Chứng minh: ?*

### ❖ Định lý 3:

- *Đồ thị có hướng, liên thông yếu  $G=(V, E)$  có chu trình Euler khi và chỉ khi mọi đỉnh của  $G$  có bán bậc vào bằng bán bậc ra.*  
=> Khi  $G$  (có hướng) có chu trình Euler thì nó liên thông mạnh.

### ❖ Định lý 4:

- *Đồ thị có hướng, liên thông yếu  $G=(V, E)$  có đường đi Euler nhưng không có chu trình Euler khi và chỉ khi  $G$  tồn tại duy nhất hai đỉnh sao cho:  $\deg^+(u) - \deg^-(u) = \deg^+(v) - \deg^-(v) = 1$ , và tất cả các đỉnh còn lại có bán bậc vào bằng bán bậc ra.*

# I.3.Giải thuật x/d chu trình Euler

CT, CTcon là các chu trình

**Bước 1:** Đầu tiên, xây dựng 1 chu trình CT trong G

**Bước 2:**  $H \leftarrow (G \setminus CT) \setminus \{\text{Các đỉnh cô lập sau khi bỏ CT khỏi } G\}$ .

**Bước 3:** Nếu H vẫn còn cạnh thì đến bước 4. Ngược lại đến bước 8.

**Bước 4:** Xây dựng chu trình con CTcon trong H với đỉnh đầu thuộc chu trình CT

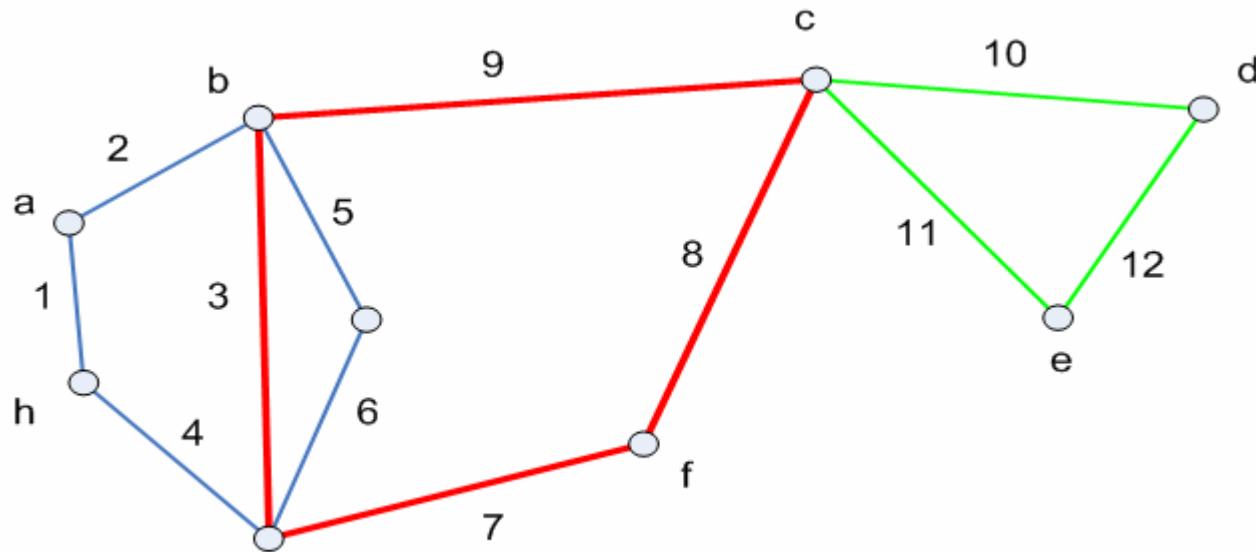
**Bước 5:**  $H \leftarrow (H \setminus CTcon) \setminus \{\text{Các đỉnh cô lập sau khi bỏ CTcon khỏi } H\}$

**Bước 6:**  $CT \leftarrow CT \cup CTcon$

**Bước 7:** Đến bước 3.

**Bước 8:** Kết thúc. CT là chu trình Euler

# I.3.Giải thuật x/d chu trình Euler



$$CT = \{3, 7, 8, 9\}.$$

$$H = \{G \setminus CT\} \setminus \{\text{Các đỉnh cô lập}\} = \{1, 2, 4, 5, 6, 10, 11, 12\}.$$

+ Lần 1:

$$CT_{con} = \{10, 11, 12\}.$$

$$H = \{H \setminus H_{con}\} \setminus \{\text{Các đỉnh cô lập}\} = \{1, 2, 4, 5, 6\}.$$

+ Lần 2:

$$CT_{con} = \{1, 2, 5, 6, 4\}$$

$$H = \{H \setminus H_{con}\} \setminus \{\text{Các đỉnh cô lập}\} = \emptyset. \text{ DỪNG.}$$

Cuối cùng ta có chu trình Euler: 3, 2, 1, 4, 6, 5, 9, 10, 12, 11, 8, 7.

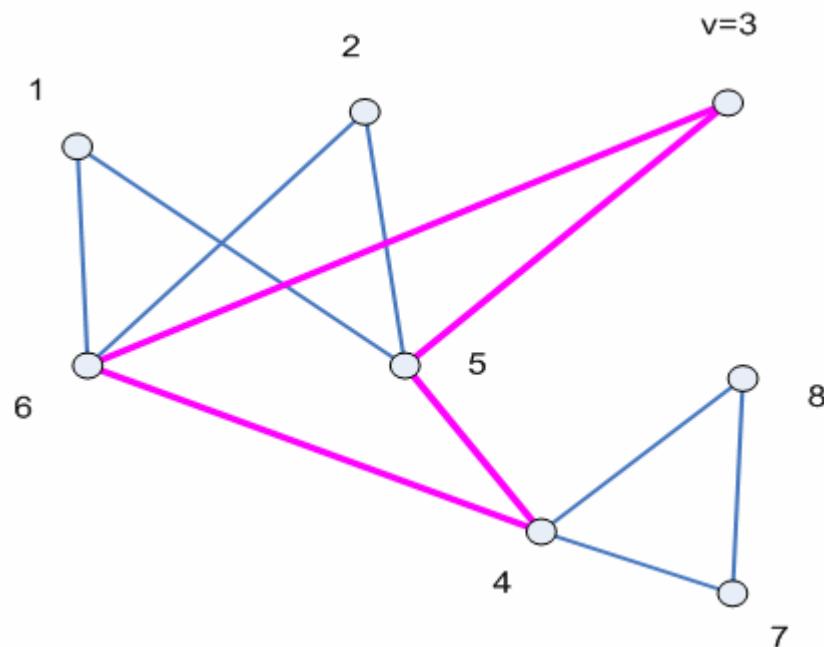
# I.3.Giải thuật x/d chu trình Euler

## ❖ Cài đặt

```
main(){
    STACK = ∅;
    CE = ∅; /* CE - Chu trình Euler */
    Chọn u là 1 đỉnh bất kỳ của đồ thị;
    STACK ← u;
    while (STACK != ∅){
        x = top(STACK);
        if (Ke(x) != ∅ ){
            y = Đỉnh đầu trong danh sách Ke(x);
            STACK ← y;
            Ke(x) = Ke(x) \ {y};
            Ke(y) = Ke(y) \ {x};      /* Bỏ cạnh (x,y) */
        }else {
            x ← STACK;
            CE ← x;
        }
    }
}
```

# I.3.Giải thuật x/d chu trình Euler

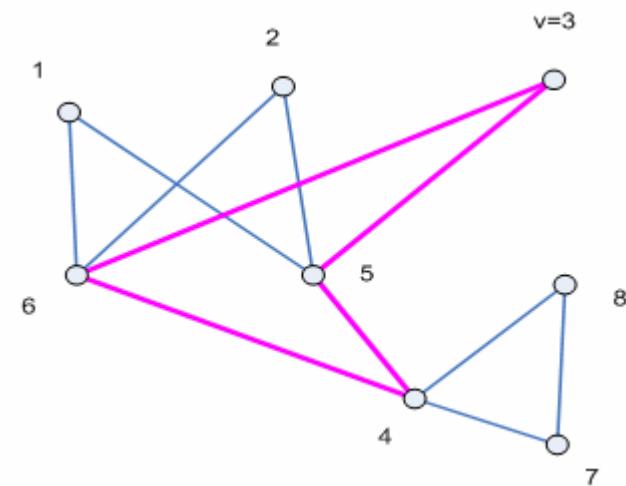
## ❖ Cài đặt



Đỉnh v	Ke(v)
1	6, 5
2	5, 6
3	6, 5
4	6, 5, 7, 8
5	4, 3, 2, 1
6	4, 3, 2, 1
7	4, 8
8	4, 7

# I.3.Giải thuật x/d chu trình Euler

STACK	CE
3, 6	$\emptyset$
3, 6, 4	$\emptyset$
3, 6, 4, 5	$\emptyset$
3, 6, 4, 5, 3	$\emptyset$
3, 6, 4, 5	3
3, 6, 4, 5, 2	3
3, 6, 4, 5, 2, 6	3
3, 6, 4, 5, 2, 6, 1	3
3, 6, 4, 5, 2, 6, 1, 5	3
3, 6, 4	3, 5, 1, 6, 2, 5
3, 6, 4, 7	3, 5, 1, 6, 2, 5
3, 6, 4, 7, 8	3, 5, 1, 6, 2, 5
3, 6, 4, 7, 8, 4	3, 5, 1, 6, 2, 5
$\emptyset$	3, 5, 1, 6, 2, 5, 4, 8, 7, 4, 6, 3



Đỉnh v	Ke(v)
1	6, 5
2	5, 6
3	6, 5
4	6, 5, 7, 8
5	4, 3, 2, 1
6	4, 3, 2, 1
7	4, 8
8	4, 7

## I.3.Giải thuật x/d chu trình Euler

### ❖ Thuật toán Fleury

*Bắt đầu từ một đỉnh bất kỳ, đi theo các cạnh của đồ thị theo quy tắc sau:*

- *Qui tắc 1: Xóa các cạnh đã đi qua và các đỉnh cô lập nếu có*
- *Qui tắc 2: Tại mỗi đỉnh, ta chỉ đi qua cầu nếu không còn đường nào khác.*

# Nội dung



I.

Đồ thị Euler



II.

Đồ thị Hamilton

## II. Đồ thị Hamilton

### Đồ thị Hamilton

1. Định nghĩa

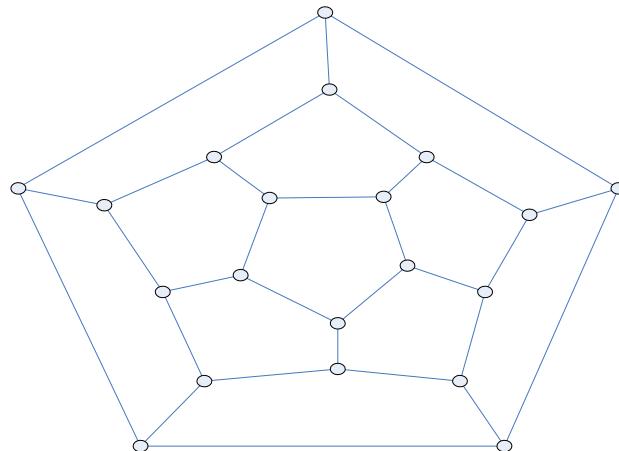
2. Định lý

3. Giải thuật xây dựng chu trình Hamilton

## II.1. Định nghĩa

### ❖ Lịch sử

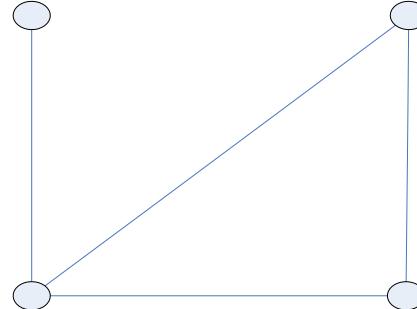
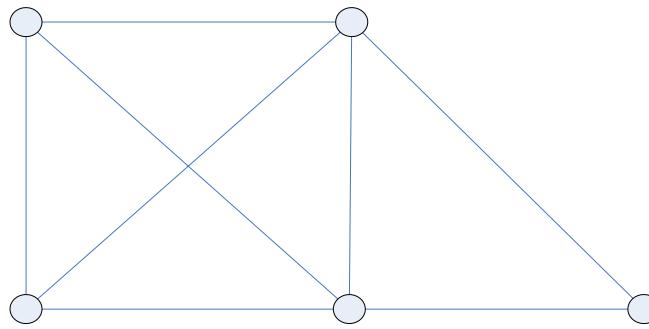
- “ Giả sử ta có một khối 12 mặt, mỗi mặt là một hình ngũ giác đều. Mỗi đỉnh trong 20 đỉnh của khối này được đặt bằng tên của một thành phố. Hãy tìm một đường xuất phát từ một thành phố, đi dọc theo các cạnh của khối, ghé thăm mỗi một trong 19 thành phố còn lại đúng một lần, cuối cùng trở lại thành phố ban đầu”



Trong đồ thị hình trên có hay không một chu trình đi qua tất cả các đỉnh của đồ thị, mỗi đỉnh đúng một lần ?

## II.1. Định nghĩa

- ❖ Giả sử  $G$  là đơn đồ thị vô (có) hướng, ta có các định nghĩa sau:
  - **Chu trình Hamilton** là chu trình xuất phát từ một đỉnh, đi thăm tất cả các đỉnh còn lại mỗi đỉnh đúng một lần, cuối cùng quay trở lại đỉnh xuất phát. Đồ thị có chu trình Hamilton gọi là **đồ thị Hamilton**.
  - **Đường đi Hamilton** là đường đi qua tất cả các đỉnh của đồ thị, mỗi đỉnh đúng một lần. Đồ thị có đường đi Hamilton gọi là **đồ thị nửa Hamilton**.



## II.2. Định lý

### ❖ Nhận biết đồ thị Hamilton

- Chưa có chuẩn để nhận biết 1 đồ thị có là Hamilton hay không
- Chưa có thuật toán để kiểm tra
- Các kết quả thu được ở dạng điều kiện đủ
- Nếu  $G$  có số cạnh đủ lớn thì  $G$  là Hamilton

## II.2. Định lý

### ❖ Định lý Dirac

- Cho đồ thị vô hướng  $G=(V, E)$  có  $n$  đỉnh ( $n \geq 3$ ). Nếu mọi đỉnh  $v$  của đồ thị đều có  $\deg(v) \geq n/2$  thì  $G$  có chu trình Hamilton.

## II.2. Định lý

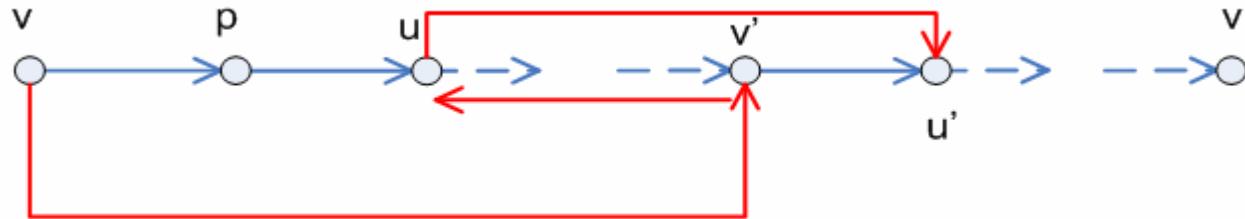
### ❖ Chứng minh

- *Thêm vào  $G$  k đỉnh mới và nối chúng với tất cả các đỉnh của  $G$  ta được  $G'$ .*
- *Giả sử  $k$  là số nhỏ nhất sao cho  $G'$  là đồ thị Hamilton.*
- *Ta sẽ chứng minh là  $k = 0$ .*

## II.2. Định lý

### ❖ Chứng minh

- Giả sử  $k > 0$ , Xét chu trình Hamilton trong  $G'$ :  $v \rightarrow p \rightarrow w \rightarrow \dots v$ . Với  $p$  là 1 trong những đỉnh mới. Ta thấy:
  - $v$  và  $w$  không thể kề nhau ( Ngược lại khi đó có thể bỏ  $p$  – vô lý vì  $k$  là min )
  - Nếu  $v'$  kề  $v$  và  $w'$  kề  $w$  thì  $w'$  không thể đi liền sau  $v'$ . Trái lại: Ta thay  $v \rightarrow p \rightarrow w \rightarrow \dots v' \rightarrow w' \rightarrow \dots \rightarrow v$  bởi:  $v \rightarrow v' \rightarrow \dots \rightarrow w \rightarrow w' \rightarrow \dots \rightarrow v$  bỏ qua  $p$ . Do đó: **Với mỗi đỉnh kề với  $v$  ta luôn tìm được 1 đỉnh không kề với  $w$ :**



- Số đỉnh không kề với  $w \geq$  số đỉnh kề với  $v \geq (n/2 + k)$
- Mà số đỉnh kề với  $w \geq (n/2 + k)$
- Do đó  $|VG'| \geq (n + 2k) > n + k$  Vô lý !!! (ĐPCM)

## II.2. Định lý

### ❖ Định lý Dirac cho đồ thị có hướng

- Cho đồ thị có hướng, liên thông mạnh  $G=(V, E)$  và có  $n$  đỉnh. Nếu mọi đỉnh  $v \in V$  đều có  $\deg(v) \geq n/2$  và  $G$  có chu trình Hamilton.

## II.3. Giải thuật x/d chu trình Hamilton

### ❖ Dùng giải thuật quay lui

- *Bắt đầu từ 1 đỉnh, đi theo con đường dài nhất có thể được (depth – first)*
- *Nếu đường đó chứa mọi đỉnh và có thể nối 2 đỉnh đầu và cuối bằng 1 cạnh thì đó là chu trình Hamilton*
- *Nếu trái lại ta lùi lại một đỉnh để mở con đường theo chiều sâu khác*
- *Cứ tiếp tục quá trình trên cho đến khi thu được chu trình Hamilton.*

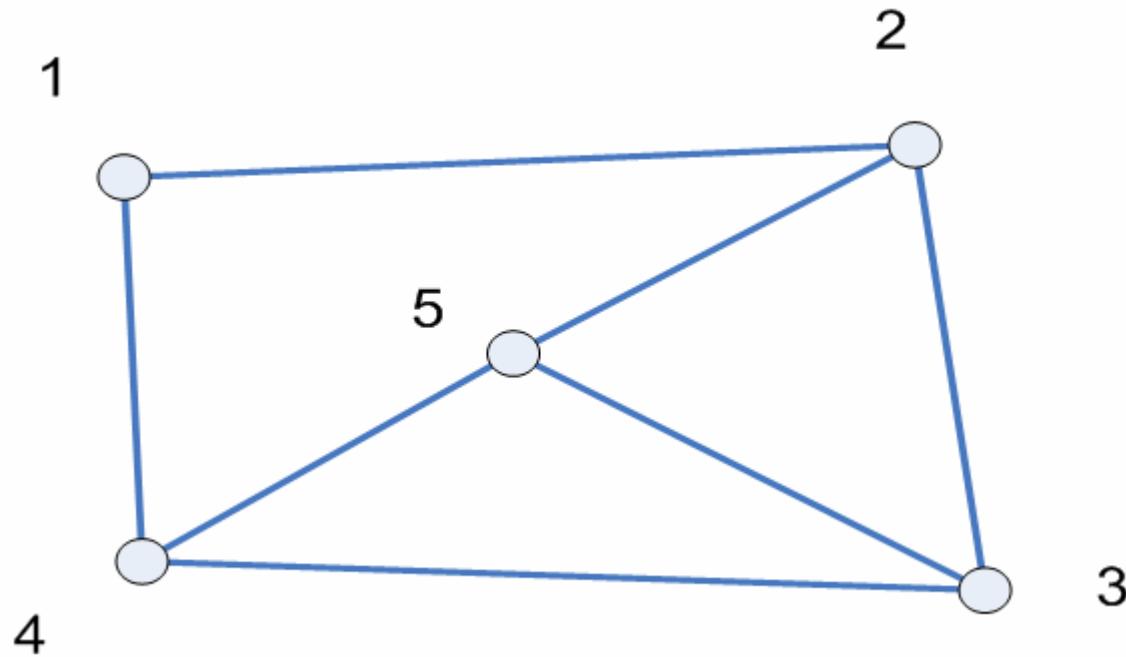
## II.3. Giải thuật x/d chu trình Hamilton

### ❖ Cài đặt thuật toán

```
void hamilton(k)
/*Phát triển dãy X1,X2,...,Xk-1
G=(V,E) được cho bởi Danh Sách kề: Ke(v), v ∈ V */
{
    for ( y ∈ Ke(Xk-1) )
        if ( ( k == n+1 ) && ( y == v0 ) ) Xuất(X1,...Xn,v0);
        else if ( Chuaxet[y] ) {
            Xk = y;
            Chuaxet[y] = 0;
            Hamilton(k+1);
            Chuaxet[y] = 1; //Quay lui
        }
}
main(){
    for (v ∈ V) Chuaxet[v] = 1;
    X1 = v0; Chuaxet[v0] = 0; Hamilton(2);
}
```

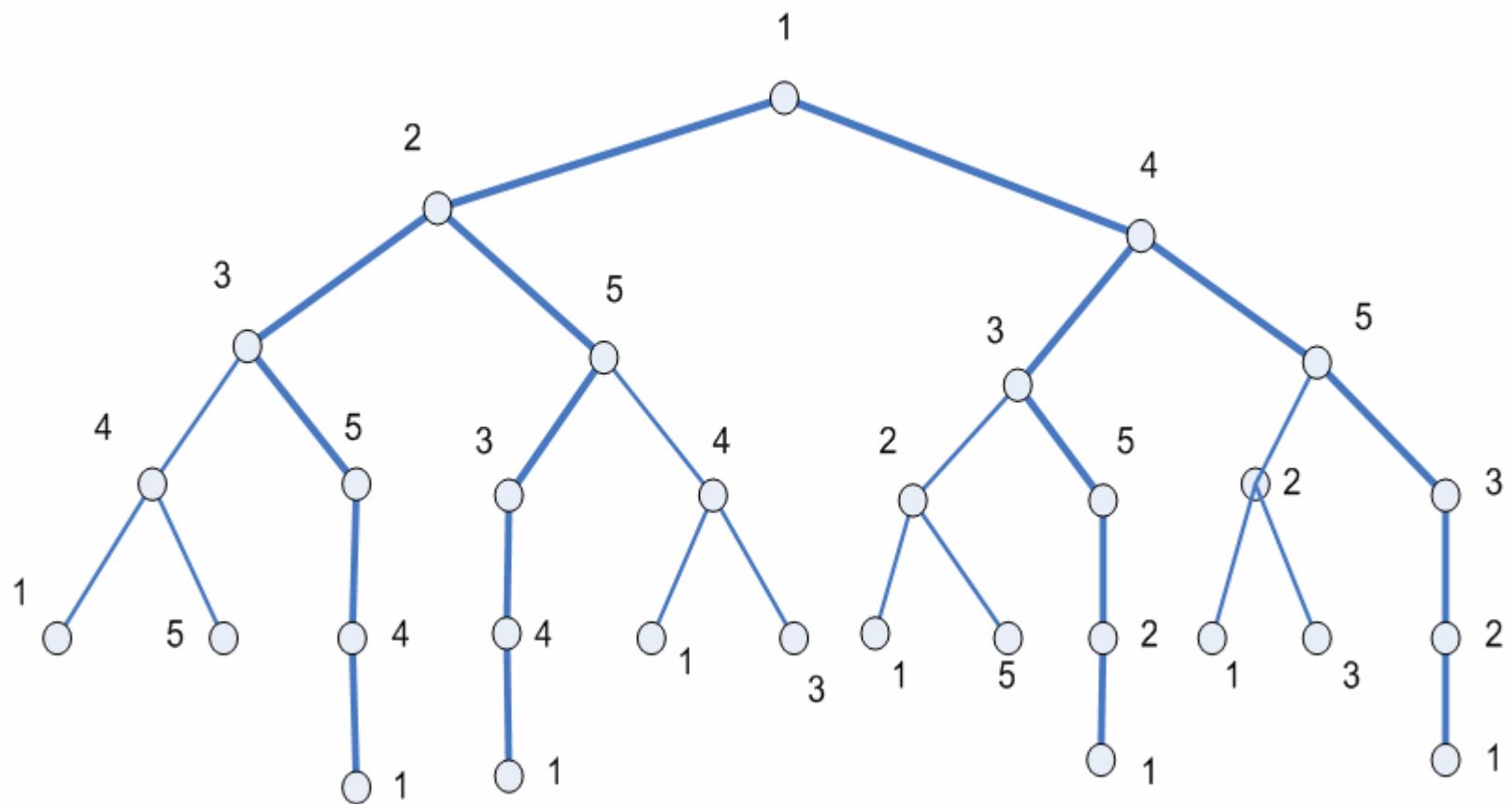
## II.3. Giải thuật x/d chu trình Hamilton

❖ Ví dụ



## II.3. Giải thuật x/d chu trình Hamilton

❖ Ví dụ



## Chương 5: Cây

# Nội dung



**I.** Định nghĩa



**II.** Cây khung của đồ thị



**III.** Tập các chu trình cơ bản



**IV.** Cây khung nhỏ nhất

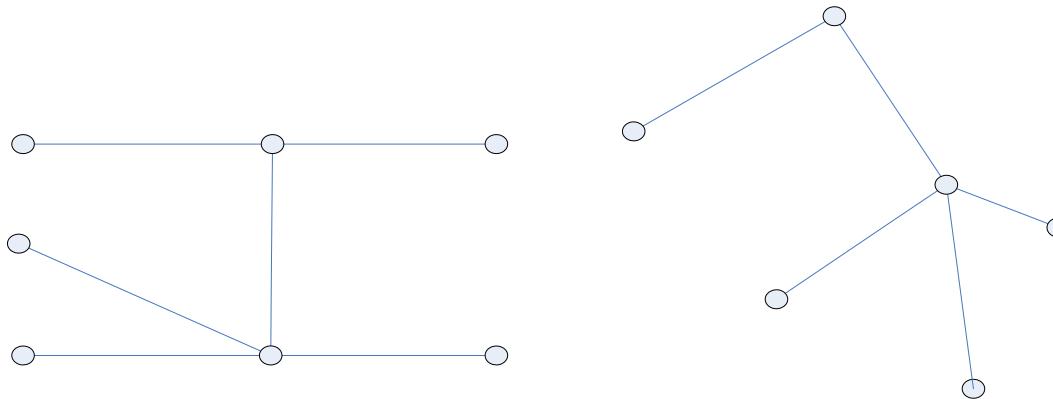


**V.** Cây có gốc

# I. Định nghĩa

## ❖ Cây là đồ thị vô hướng

- Liên thông
- Không có chu trình



## ❖ Rừng là đồ thị vô hướng

- Không có chu trình

# I. Định nghĩa

## ❖ Định lý nhận biết cây

Cho  $T = (V, E)$  là đồ thị vô hướng  $n$  đỉnh. Các mệnh đề sau đây là tương đương:

- **MĐ1:**  $T$  là cây ( $T$  liên thông và không chứa chu trình).
- **MĐ2:**  $T$  không chứa chu trình và có  $n-1$  cạnh.
- **MĐ3:**  $T$  liên thông và có  $n-1$  cạnh.
- **MĐ4:**  $T$  liên thông và mỗi cạnh của nó đều là cầu.
- **MĐ5:** Hai đỉnh bất kỳ của  $T$  được nối với nhau bởi đúng 1 đường đi đơn.
- **MĐ6:**  $T$  không chứa chu trình nhưng nếu cứ thêm vào nó một cạnh ta thu được đúng 1 chu trình.

# I. Định nghĩa

❖ Định lý nhận biết cây

❖ Chứng minh:

Ta sẽ chứng minh định lý trên theo sơ đồ sau:

- $MĐ1 \Rightarrow MĐ2 \Rightarrow MĐ3 \Rightarrow MĐ4 \Rightarrow MĐ5 \Rightarrow MĐ6 \Rightarrow MĐ1$

# I. Định nghĩa

❖ **Chứng minh MĐ1  $\Rightarrow$  MĐ2: Nếu  $T$  là cây  $n$  đỉnh thì  $T$  không có chu trình và có  $n-1$  cạnh**

Chứng minh bằng phương pháp quy nạp

- Với  $n=1$  thì đồ thị có  $n-1 = 1 - 1 = 0$  (Đúng)
- Giả sử khẳng định đúng  $\forall$  cây có  $k \geq 1$  đỉnh. Ta sẽ chỉ ra  $\forall$  cây  $T$  có  $k+1 \geq 1$  đỉnh sẽ có số cạnh là  $k$ .

Chọn đường đi dài nhất trong  $G$  là  $P = (v_1, v_2, \dots, v_m)$ . Rõ ràng **v1 là đỉnh treo**:

- $v_1$  không thể kề với các đỉnh  $v_3, \dots, v_m$  vì  $G$  không có chu trình.
- $v_1$  không thể được nối với các đỉnh khác vì  $P$  là dài nhất

Xét  $G' = G \setminus \{v_1, (v_1, v_2)\}$  (**Không thể bỏ các đỉnh trung gian**). Ta được  $G'$  có  $k$  đỉnh. Theo giả thiết quy nạp  $G'$  có  $k-1$  cạnh. Do đó  $G$  có  $k$  cạnh (ĐPCM)

# I. Định nghĩa

❖ **Chứng minh MĐ2  $\Rightarrow$  MĐ3:** Nếu  $T$  không chứa chu trình và có  $n-1$  cạnh thì  $T$  liên thông.

Chứng minh bằng phương pháp phản chứng.

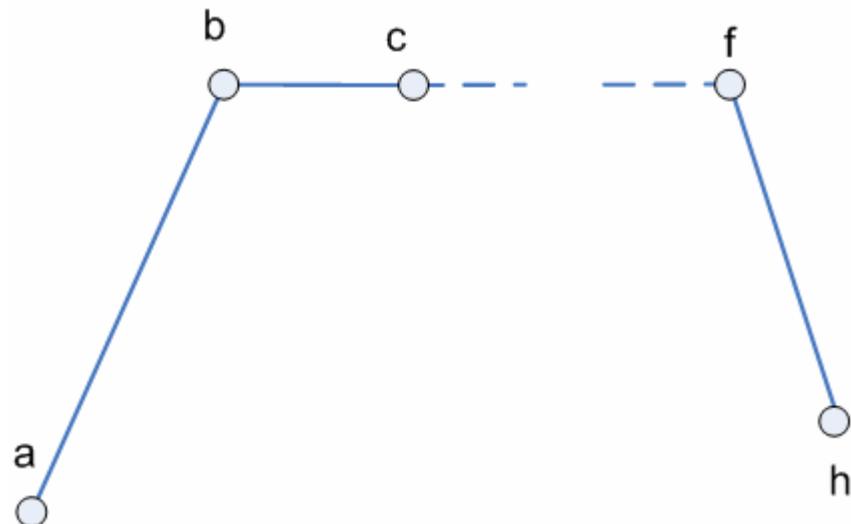
- Giả sử  $T$  không liên thông, khi đó  $T$  được phân rã thành  $k > 1$  thành phần liên thông  $T_1, T_2, \dots, T_k$ . Vì  $T$  không chứa chu trình (theo giả thiết) nên các cây cũng vậy, suy ra  $T_i$  là cây.
- Gọi  $v(T)$  và  $e(T)$  tương ứng là số đỉnh và cạnh của  $T$ . Theo phần trước MĐ1  $\Rightarrow$  MĐ2 ta có:  $e(T_i) = v(T_i) - 1$ . Suy ra:
  - $\sum e(T_i) = \sum (v(T_i) - 1) = \sum v(T_i) - k$
  - $\Leftrightarrow e(T) = v(T) - k$
  - $\Leftrightarrow n - 1 = n - k$ . Vô lý với  $k > 1$  (ĐPCM)

# I. Định nghĩa

- ❖ **Chứng minh MĐ3  $\Rightarrow$  MĐ4:** Nếu  $T$  liên thông và có  $n-1$  cạnh thì mỗi cạnh của  $T$  là cầu
  - Suy luận tương tự như chứng minh MĐ1  $\Rightarrow$  MĐ2.
  - Chọn đường đi dài nhất  $P = (v_1, v_2, v_3, \dots, v_m)$ .
  - Nếu từ đồ thị  $T$  ta bỏ đi một cạnh nào đó trên đường đi  $P$ , thì rõ ràng **không còn con đường nào khác để đi từ  $v_1$  đến  $v_m$**  (vì nếu ngược lại thì  $T$  có chu trình). Vì vậy các cạnh của  $T$  đều là cầu.

# I. Định nghĩa

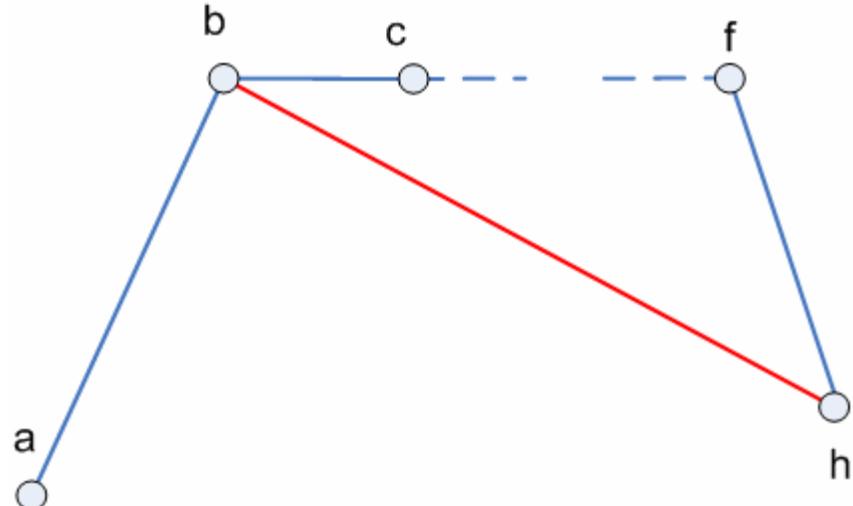
- ❖ **Chứng minh MĐ4  $\Rightarrow$  MĐ5:** Nếu  $T$  liên thông và mỗi cạnh của  $T$  là cầu thì hai đỉnh bất kỳ của  $T$  được nối với nhau đúng bởi 1 đường đơn.
  - $T$  liên thông nên mọi 2 đỉnh của  $T$  tồn tại đường nối giữa chúng. Đường nối này là duy nhất vì trái lại  $T$  sẽ có chu trình và các cạnh trên chu trình đó sẽ không thể là cầu.(ĐPCM)



# I. Định nghĩa

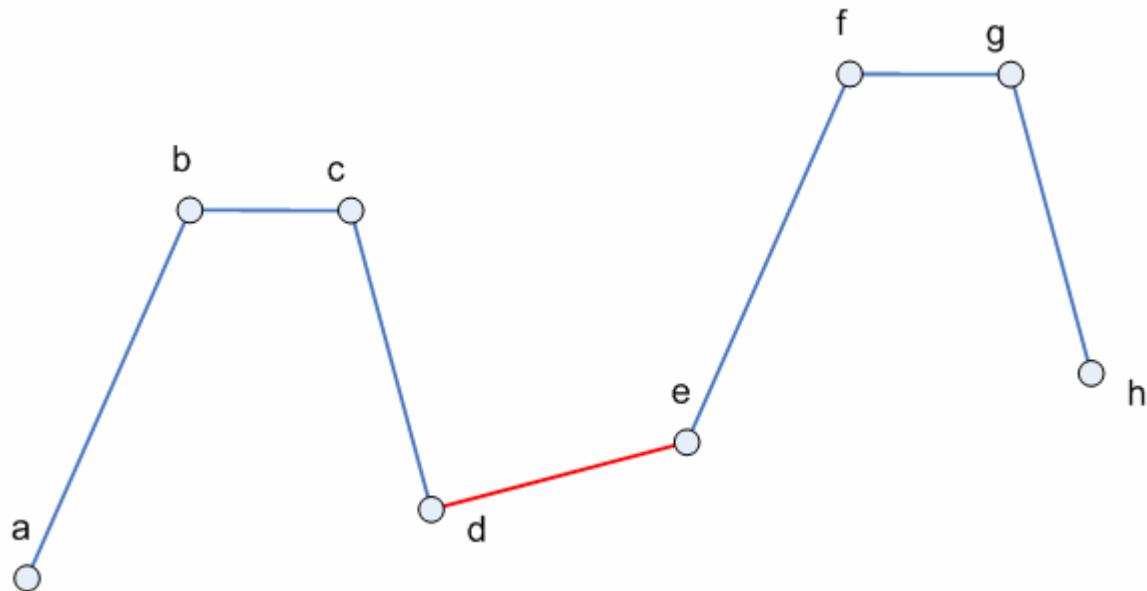
❖ **Chứng minh MĐ5  $\Rightarrow$  MĐ6:** Nếu hai đỉnh bất kỳ của  $T$  được nối với nhau đúng bởi 1 đường đơn thì  $T$  không chứa chu trình nhưng nếu cứ thêm vào nó 1 cạnh ta thu được đúng 1 chu trình

- $T$  không chứa chu trình vì nếu  $T$  có chu trình thì sẽ có cặp đỉnh được nối với nhau bởi 2 đường đơn.
- Thêm vào cạnh  $(u,v)$  ta sẽ nhận được chu trình gồm đường đơn nối  $u$  với  $v$  và cạnh  $(u,v)$  mới.
- Do đường đơn nói trên là duy nhất nên chu trình nhận được cũng là duy nhất.
- (ĐPCM)



# I. Định nghĩa

- ❖ **Chứng minh MĐ6  $\Rightarrow$  MĐ1:** *T không chứa chu trình nhưng nếu cứ thêm vào nó một cạnh ta thu được đúng 1 chu trình thì T là cây (liên thông và không có chu trình).*
  - Chứng minh bằng phản chứng



# Nội dung



**I.** Định nghĩa



**II.** Cây khung của đồ thị



**III.** Tập các chu trình cơ bản



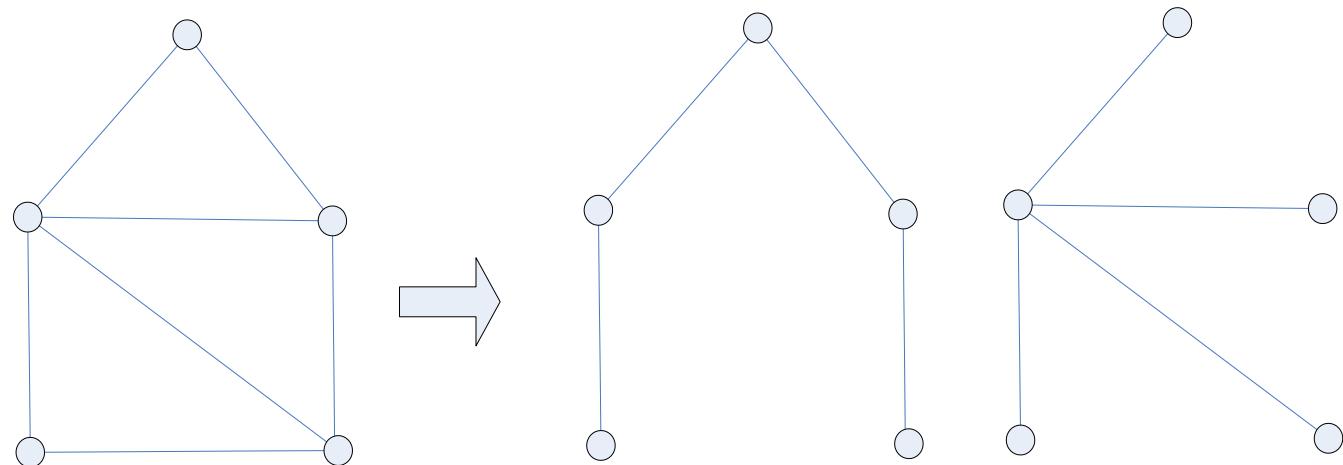
**IV.** Cây khung nhỏ nhất



**V.** Cây có gốc

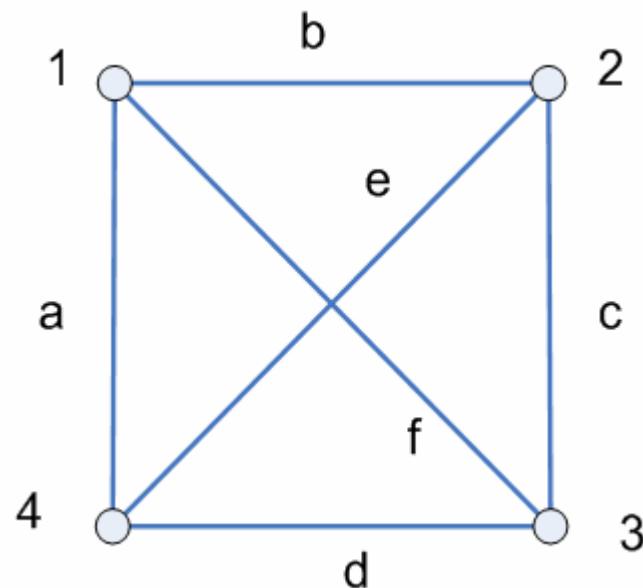
## II.1. Định nghĩa

- Cho đồ thị  $G = (V, E)$  vô hướng, liên thông. Một cây  $T = (V, F)$  được xây dựng từ  $G$  với  $F \subset E$  ( $T$  chứa tất cả các đỉnh của  $G$  và tập cạnh  $F$  là con của tập cạnh  $E$ ) được gọi là **cây khung** của đồ thị  $G$ .
- Cây bao trùm hay cây tối đại.



## II.2. Định lý Cayley

- Số cây khung của đồ thị  $K_n$  là  $n^{n-2}$



abc, bcd, cda, dab,  
afc, dfb, aec, deb,  
aed, afb,bec, cfd,  
efc,efd,efa,efb.

Số cây khung là:  $4^2 = 16$

## II.3. Xây dựng cây khung

- Xây dựng theo chiều sâu
- Xây dựng theo chiều rộng

### Tham số

- Input: Đồ thị G lưu dưới dạng danh sách kề - Mảng **Ke[]**
- Output: Cây khung T của đồ thị

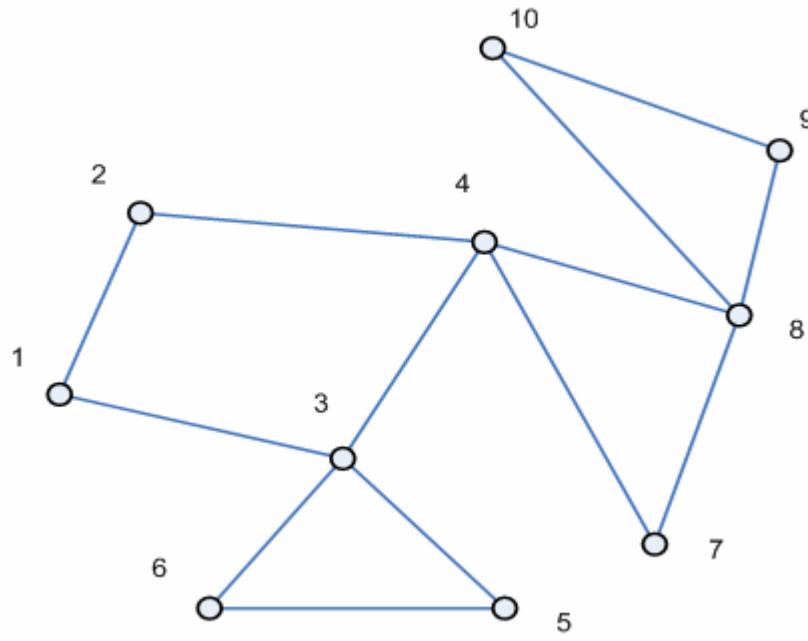
Mảng **ChuaXet[]** dùng để đánh dấu các đỉnh đã được xét hay chưa.

## II.3.a. X/d theo chiều sâu

```
/* Khai báo các biến toàn cục ChuaXet, Ke, T */
void Tree_DFS(v);
{
    ChuaXet[v] = 0;
    for (u ∈ Ke(v))
        if (ChuaXet[u]) {
            T = T ∪ (v,u);
            Tree_DFS(u);
        };
}
main(){
    /* Nhập đồ thị, tạo biến Ke */
    for (v ∈ V)
        ChuaXet[v] = 1; /* Khởi tạo cờ cho đỉnh */
    T = ∅;           /* T là tập cạnh cây khung */
    Tree_DFS(root); /* root là đỉnh nào đó của đồ thị */
}
```

## II.3.a. X/d theo chiều sâu

### ❖ Ví dụ



Đỉnh v	K <sub>e</sub> (v)
1	2, 3
2	4, 1
3	1, 6, 5, 4
4	2, 3, 7, 8
5	6, 3
6	3, 5
7	4, 8
8	7, 4, 10, 9
9	8, 10
10	8, 9

1 → 2 → 4 → 3 → 6 → 5

7 → 8 → 10 → 9

Cây khung của G là:

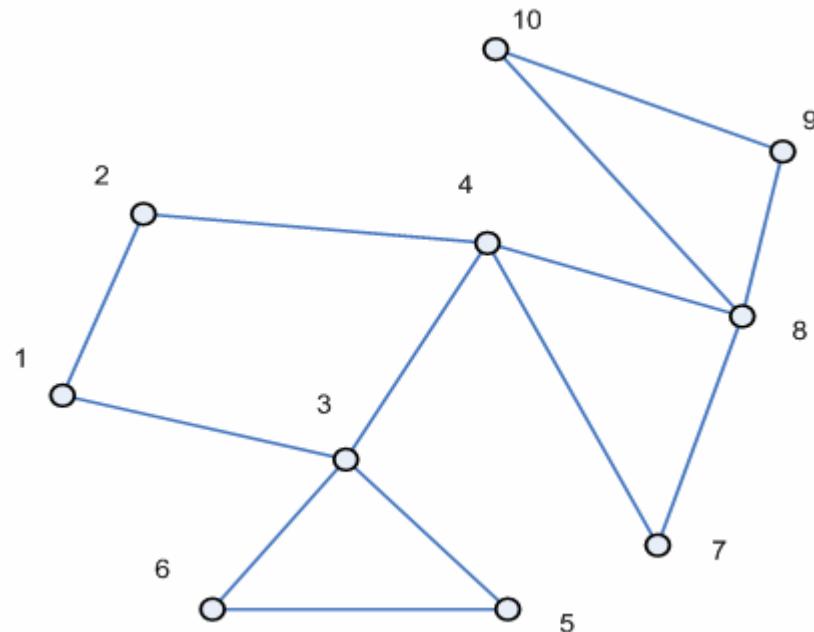
$\{(1, 2), (2, 4), (4, 3), (3, 6), (6, 5), (4, 7), (7, 8), (8, 10), (10, 9)\}$

## II.3.b. X/d theo chiều rộng

```
/* Khai báo các biến toàn cục ChuaXet, Ke, QUEUE */
void Tree_BFS(r){
    QUEUE = Ø;
    QUEUE ⇐ r;
    ChuaXet[r] = 0;
    while (QUEUE != Ø ){
        v ⇐ QUEUE;
        for (u ∈ Ke(v))
            if ( ChuaXet[u] ){
                QUEUE ⇐ u;
                ChuaXet[u] = 0;
                T = T ∪ (v,u);
            };
    }
}
main() /* Nhập đồ thị, tạo biến Ke */{
    for (v ∈ V)
        ChuaXet[v] = 1; /* Khởi tạo cờ cho đỉnh */
    T = Ø;             /* T là tập cạnh cây khung */
    Tree_DFS(root);   /* root là đỉnh nào đó của đồ thị */
}
```

## II.3.b. X/d theo chiều rộng

### ❖ Ví dụ



$1 \rightarrow 2 \rightarrow 3$

$4 \rightarrow 6 \rightarrow 5$

$7 \rightarrow 8$

$10 \rightarrow 9$

Cây khung của G là:  $\{(1, 2), (2, 3), (2, 4), (4, 6), (6, 5), (4, 7), (7, 8), (8, 10), (10, 9)\}$

Đỉnh v	Ke(v)
1	2, 3
2	4, 1
3	1, 6, 5, 4
4	2, 3, 7, 8
5	6, 3
6	3, 5
7	4, 8
8	7, 4, 10, 9
9	8, 10
10	8, 9

# Nội dung



**I.** Định nghĩa



**II.** Cây khung của đồ thị



**III.** Tập các chu trình cơ bản



**IV.** Cây khung nhỏ nhất

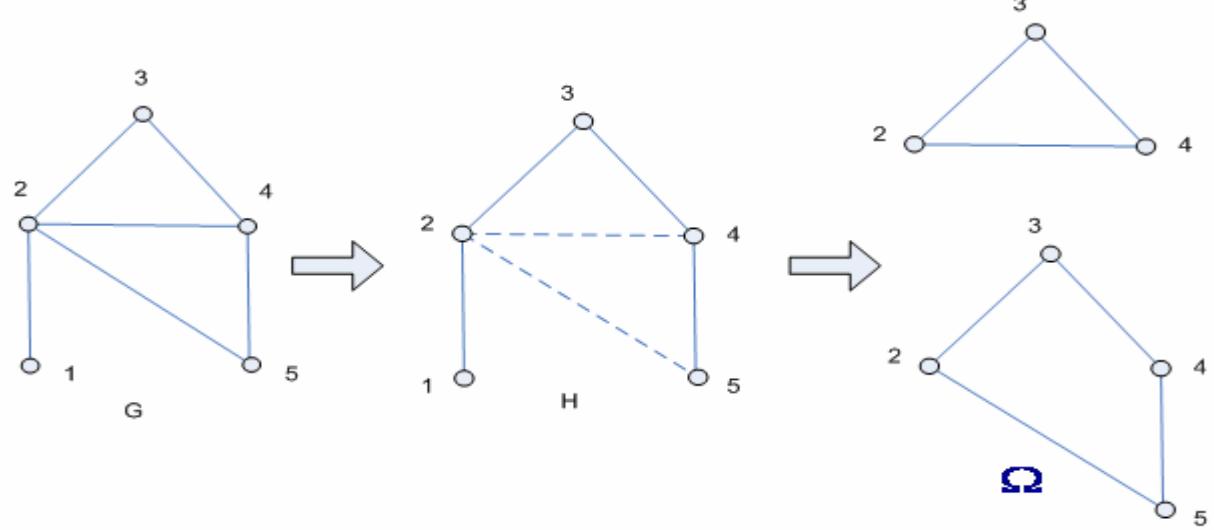


**V.** Cây có gốc

### III. Tập các chu trình cơ bản

#### ❖ Định nghĩa

- Giả sử  $G=(V,E)$  là đơn đồ thị vô hướng liên thông,  $H=(V,T)$  là cây khung của  $G$ .
- Nếu thêm một cạnh  $e \in E \setminus T$  vào cây khung  $H$  ta sẽ thu được đúng 1 chu trình trong  $H$ , ký hiệu nó là  $C_e$ . Tập các chu trình:  
 $\Omega = \{ C_e : e \in E \setminus T \}$  được gọi là tập các chu trình cơ bản của đồ thị  $G$



### III.Tập các chu trình cơ bản

#### ❖ Tính chất

- Tập các chu trình cơ bản phụ thuộc vào cây khung của đồ thị. Hai cây khung khác nhau có thể cho hai tập chu trình cơ sở khác nhau.
- Nếu một đồ thị liên thông có  $n$  đỉnh,  $m$  cạnh. Khi đó cây khung có  $n-1$  cạnh, còn lại  $m-n+1$  cạnh ngoài. Tương ứng với mỗi cạnh ngoài, ta có một chu trình cơ bản. Vì vậy, số chu trình cơ bản của một đồ thị liên thông là  $m-n+1$ .
- Tập các chu trình cơ bản là một tập nhiều nhất các chu trình thỏa mãn điều kiện: Mỗi chu trình có đúng một cạnh riêng, cạnh đó không nằm trong các chu trình còn lại và việc loại bỏ cạnh này không ảnh hưởng đến tính liên thông của đồ thị và không ảnh hưởng đến các chu trình còn lại. Như vậy ta có thể bỏ tối đa  $m - n+1$  cạnh mà vẫn đảm bảo tính liên thông của đồ thị.

### III.Tập các chu trình cơ bản

#### ❖ Ý nghĩa

- Các bài toán về mạch điện
- Mỗi mạch vòng tương ứng với một chu trình cơ bản.
- Tổng hiệu điện thế dọc theo một mạch vòng bằng 0. (ĐL Kirchoff)
- Lập hệ PT tuyến tính → Tính toán hiệu điện thế trên mọi đường dây của mạng điện.

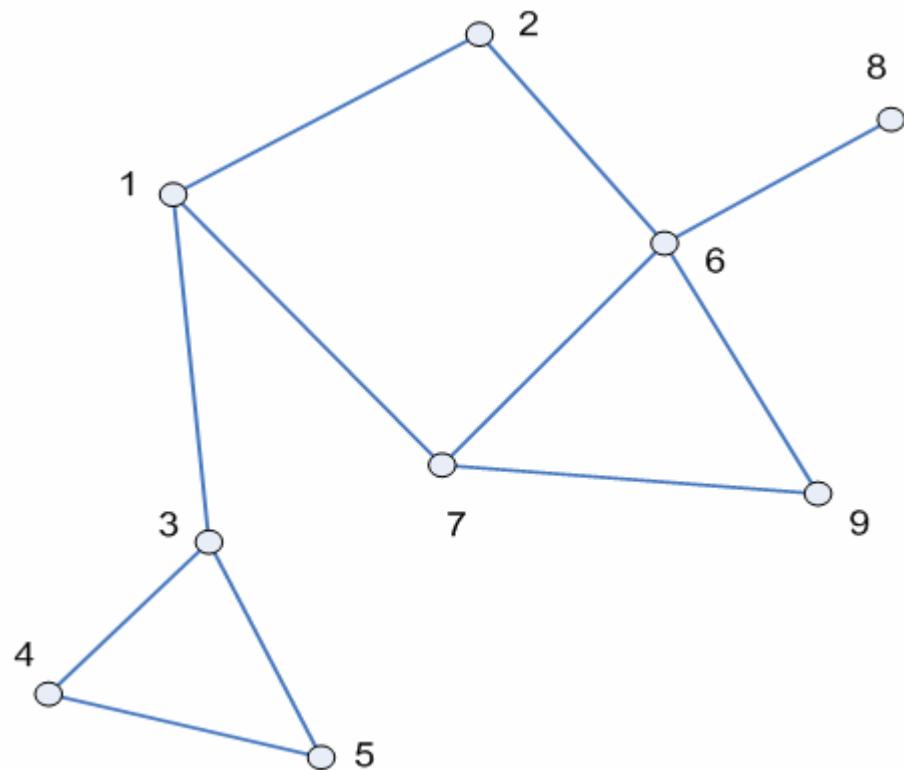
### III.Tập các chu trình cơ bản

#### ❖ Thuật toán

```
/* Khai báo các biến toàn cục d, num, STACK, Index, Ke */
void Cycle(int v){
    d ++;  STACK[d] = v;  num ++;  Index[v] = num;
    for (u ∈ Ke(v))
        if (Index[u] ==0 )      Cycle(u);
        else if (( u != STACK[d-1] ) && ( Index[v] > Index[u] ) )
            Ghi nhận chu trình STACK[d], ..., STACK[c], với STACK[c] =u;
    d --;
}
main(){
    for (v ∈ V) Index[v] = 0; /* Khởi tạo cờ cho đỉnh */
    num = 0; d = 0; STACK[0] = 0;
    for (v ∈ V)
        if (Index[v] == 0) Cycle(v);
}
```

### III. Tập các chu trình cơ bản

#### ❖ Ví dụ



Đỉnh v	K <sub>e</sub> (v)
1	2, 7, 3
2	6, 1
3	5, 4, 1
4	3, 5
5	3, 4
6	8, 9, 7, 2
7	6, 9, 1
8	6
9	7, 6

### III.Tập các chu trình cơ bản

v	d	Stack	Index	u	Ke(v)	Chu trình
1	1	1	1	2	2, 7, 3	
2	2	2	2	6	6, 1	
6	3	6	3	8	8, 9, 7, 2	
8	4	8	4	6	6	
6	3			9		
9	4	9	5	7	7, 6	
7	5	7	6	6	6, 9, 1	7, 9, 6
				9		
				1		7, 9, 6, 2, 1
9	4			6		
6	3			7		
				2		
2	2			1		
1	1			7		
				3		
3	2	3	7	5	5, 4, 1	
5	3	5	8	3	3, 4	
				4		
4	4	4	9	3	3, 5	4, 5, 3
				5		
5	3					
3	2			4		
				1		

# Nội dung



**I. Định nghĩa**



**Cây khung của đồ thị**



**Tập các chu trình cơ bản**



**Cây khung nhỏ nhất**



**Cây có gốc**

# IV. Cây khung nhỏ nhất

## Cây khung nhỏ nhất

1. Khái niệm

2. Thuật toán Kruskal

3. Thuật toán Prim

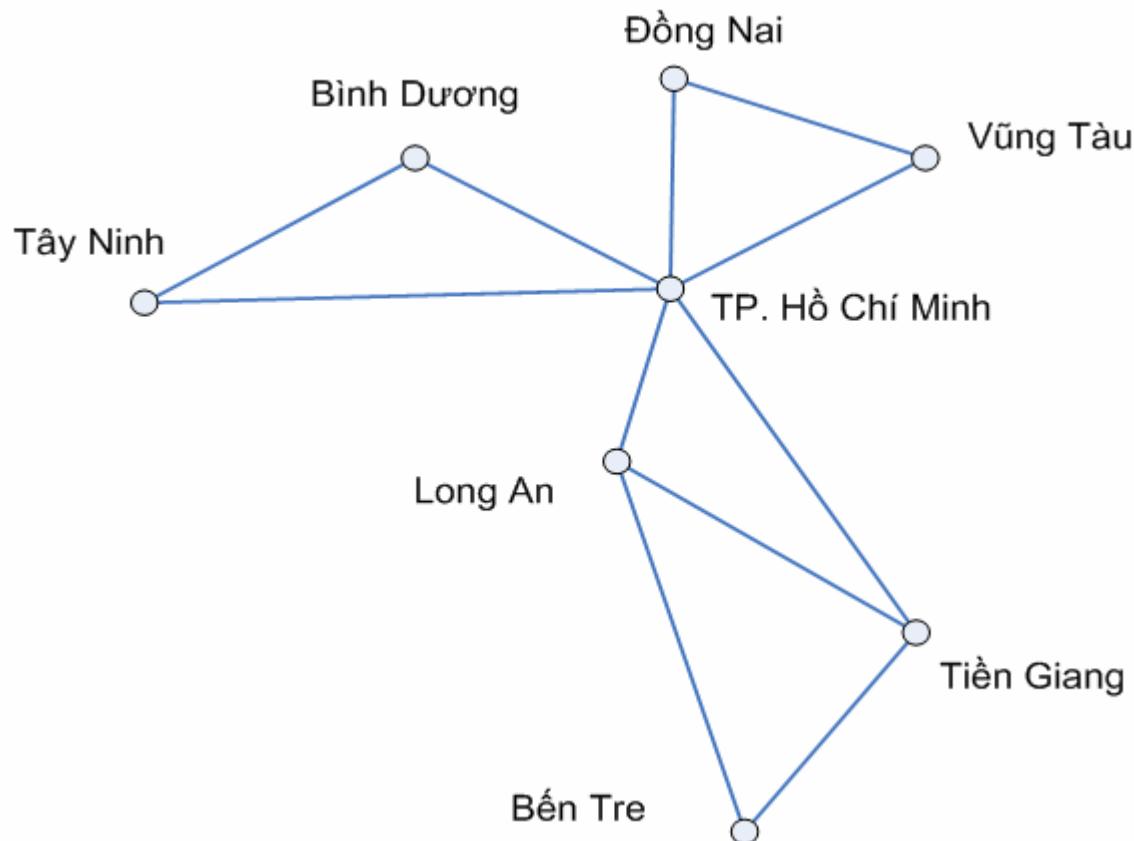
## IV.1. Khái niệm

- Cho  $G = (V, E)$  là đồ thị vô hướng liên thông.
- Mỗi cạnh  $e$  của đồ thị được gán với một số không âm  $w(e)$  gọi là **độ dài (Trọng số)** của nó.
- Giả sử  $T = (V, F)$  là cây khung của  $G$
- Trọng số của cây khung  $T$ :  
$$w(T) = \sum_{e \in F} w(e)$$
- Bài toán: Tìm  $T$  sao cho  $w(T)$  **nhỏ nhất**

## IV.1. Khái niệm

### ❖ Ứng dụng

- Bài toán xây dựng hệ thống đường sắt
- Bài toán nối mạng máy tính

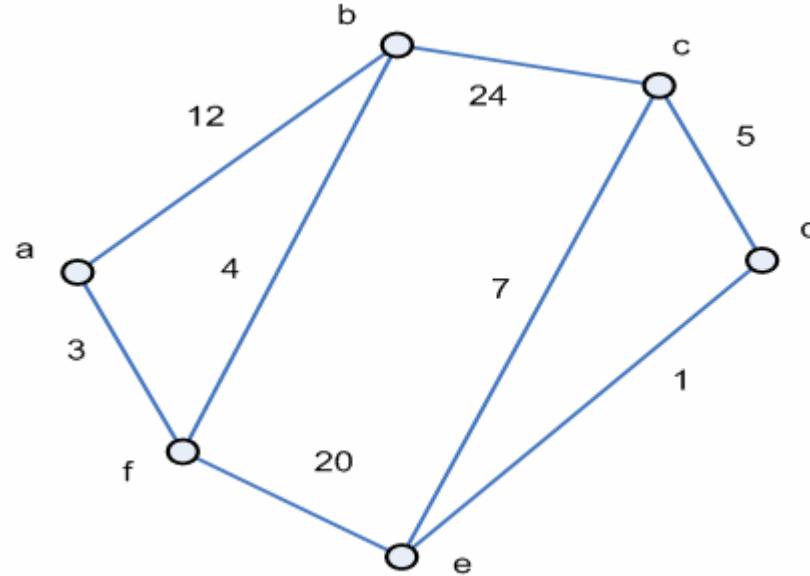


## IV.2. Thuật toán Kruskal

- Đồ thị  $G=(V, E)$ , Xây dựng tập cạnh  $F$  của  $T=(V, F)$  theo từng bước:
  - Sắp xếp các cạnh của  $G$  theo thứ tự trọng số (độ dài) tăng dần
  - Bắt đầu với  $F= \emptyset$  bổ xung dần các cạnh của  $G$  vào  $F$  với điều kiện không tạo nên chu trình trong  $T$ .
  - Thuật toán dừng lại khi có  $n-1$  cạnh được chọn.

## IV.2. Thuật toán Kruskal

### ❖ Ví dụ



(d, e)	(a, f)	(b, f)	(c, d)	(c, e)	(a, b)	(f, e)	(b, c)
1	3	4	5	7	12	20	24

(d, e)	(a, f)	(b, f)	(c, d)	(f, e)
1	3	4	5	20

## IV.2. Thuật toán Kruskal

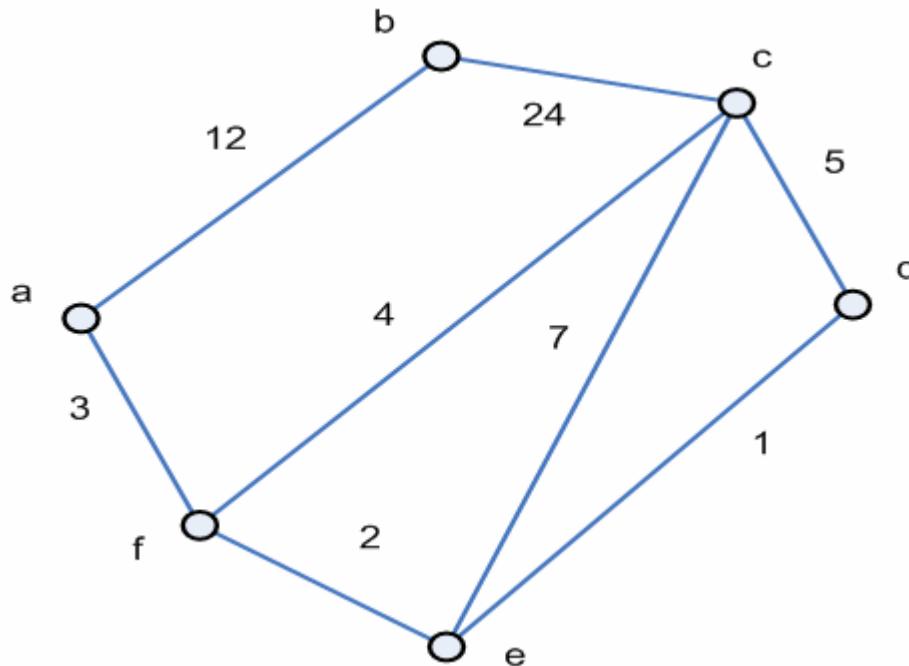
```
Void Kruskal;  
{  
    F = Ø;  
    while ( ( |F| < n-1 ) && ( E != Ø ) )  
    {  
        Chọn e = min ∈ E;  
        E = E \ {e};  
        if ( F ∪ {e} không chứa chu trình )  
            F = F ∪ {e};  
    }  
    if ( |F| < n-1 ) cout << “Đồ thị không liên thông”;  
}
```

### IV.3. Thuật toán Prim

Cho đồ thị  $G=(V, E)$ , Xây dựng tập đỉnh  $V_T$  và tập cạnh  $F$  của cây khung  $T=(V_T, F)$  theo từng bước:

1. Bắt đầu với  $V_T = s$ , một đỉnh bất kỳ và  $T=\emptyset$ . Trong tất cả các cạnh có 1 đỉnh  $\notin V_T$  và 1 đỉnh  $\in V_T$  chọn cạnh có trọng số nhỏ nhất.
2. Bổ sung cạnh đó vào  $F$  và đỉnh tương ứng vào  $V_T$ .
3. Thuật toán dừng lại khi có  $n-1$  cạnh được chọn (hoặc  $V_T=V$ ).

## IV.3. Thuật toán Prim



(a, f)	(e, f)	(d, e)	(c, f)	(a, b)
3	2	1	4	12

## IV.3. Thuật toán Prim

### ❖ Cài đặt

```
void Prim()
```

```
{
```

```
    F = Ø; VT = u;
```

```
    while ( |F| < n-1 )
```

```
{
```

*Chọn e = { min w(u,v) (u ∈ VT) & (v ∉ VT) };*

```
    F = F ∪ {e};
```

```
    VT = VT ∪ {v};
```

```
}
```

```
}
```

## IV. Cây khung nhỏ nhất

- Chứng minh tính đúng đắn và nhận xét hai thuật toán Kruskal và Prim ?

# Nội dung



**I.** Định nghĩa



**II.** Cây khung của đồ thị



**III.** Tập các chu trình cơ bản



**IV.** Cây khung nhỏ nhất



**V.** Cây có gốc

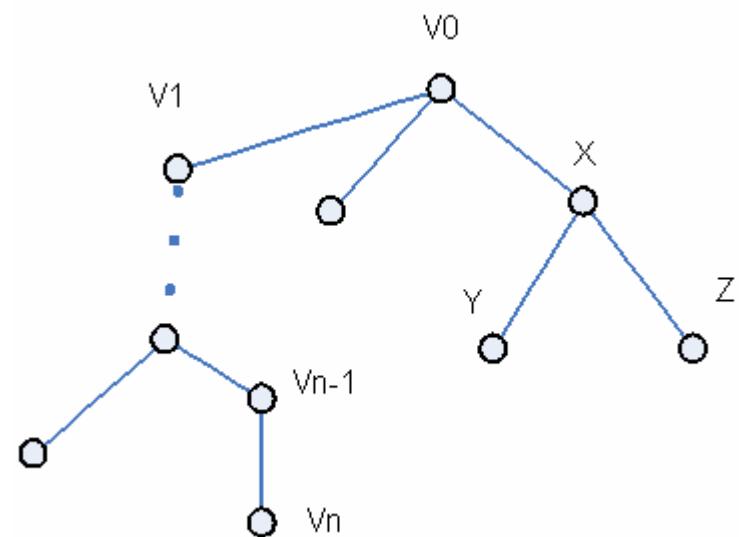
# V. Cây có gốc

## Cây có gốc

- 1. Các khái niệm
- 2. Cây tìm kiếm nhị phân
- 3. Cây quyết định
- 4. Các phương pháp duyệt cây

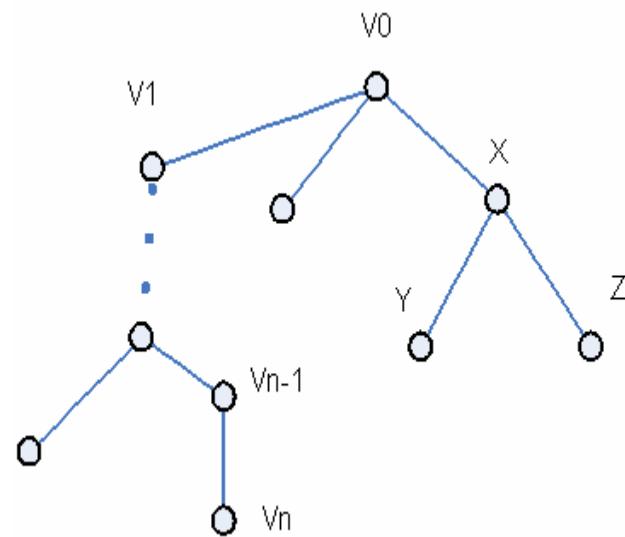
## V.1. Các khái niệm

- T là một cây có gốc
- x, y, z là các **đỉnh** trong T
- $v_0, v_1, \dots, v_n$  là một **đường đi đơn** trong T
- $v_{n-1}$  là **cha (parent)** của  $v_n$
- $v_0, v_1, \dots, v_{n-1}$  là các **tiền bối (ancestor)** của  $v_n$
- $v_n$  là **con (child)** của  $v_{n-1}$
- Nếu x là tiền bối của y thì y là **hậu duệ (descendant)** của x
- Nếu y, z là con của x thì y và z là **anh em (siblings)**

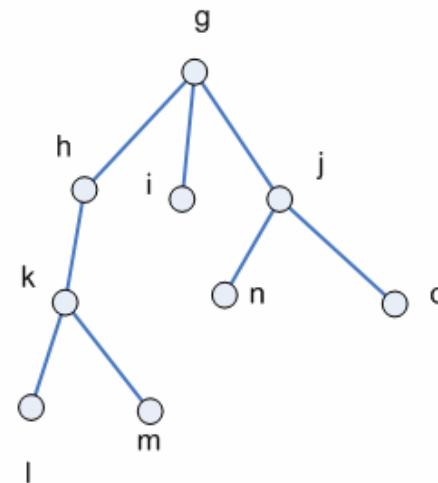
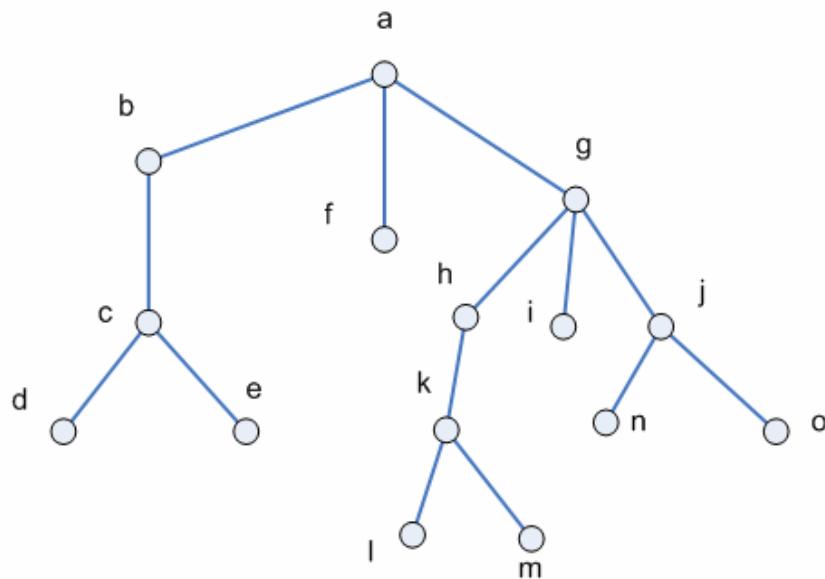


## V.1. Các khái niệm

- Nếu  $x$  không có con thì  $x$  là **lá** (*leaf*)
- Nếu  $x$  không là lá thì  $x$  là **đỉnh trong** (*branch vertex*)
- **Mức** (*level*) của đỉnh  $x$  là chiều dài (số cạnh) của đường đơn từ gốc  $v_0$  tới  $x$ .  
 $\text{level}(v_0) = 0$
- **Chiều cao** (*height*) của một cây là mức lớn nhất trong cây
- **Cây con** (*subtree*) của  $T$  gốc tại  $x$  là đồ thị con của  $T$  mà
  - Tập đỉnh gồm  $x$  và tất cả các hậu duệ của  $x$
  - Tập các cạnh gồm mọi cạnh nối tới các hậu duệ của  $x$



# V.1. Các khái niệm



- **Cha** của c là b
- **Con** của g là h, i, j
- Các **tiền bối** của e là c, b, a
- Các **hậu duệ** của b là c, d, e
- Các **đỉnh trong** : a, b, c, g, h, j, k

- Các **lá** : d, e, f, l, m, i, n, o
- **Mức** của c là 2, của k là 3
- **Chiều cao** của cây là 4
- Cây con gốc g.

## V.1. Các khái niệm

Một cây có gốc gọi là:

- **m – cây** (m-ary tree) nếu mỗi đỉnh trong không có quá m con
- **m – cây đầy** (full m-ary tree) nếu mỗi đỉnh trong có đúng m con
- **Cây nhị phân** (binary tree) nếu mỗi đỉnh không có quá 2 con
- **Cây có gốc thứ tự** (Ordered rooted tree) nếu các con của mỗi đỉnh trong được xếp thứ tự từ trái qua phải

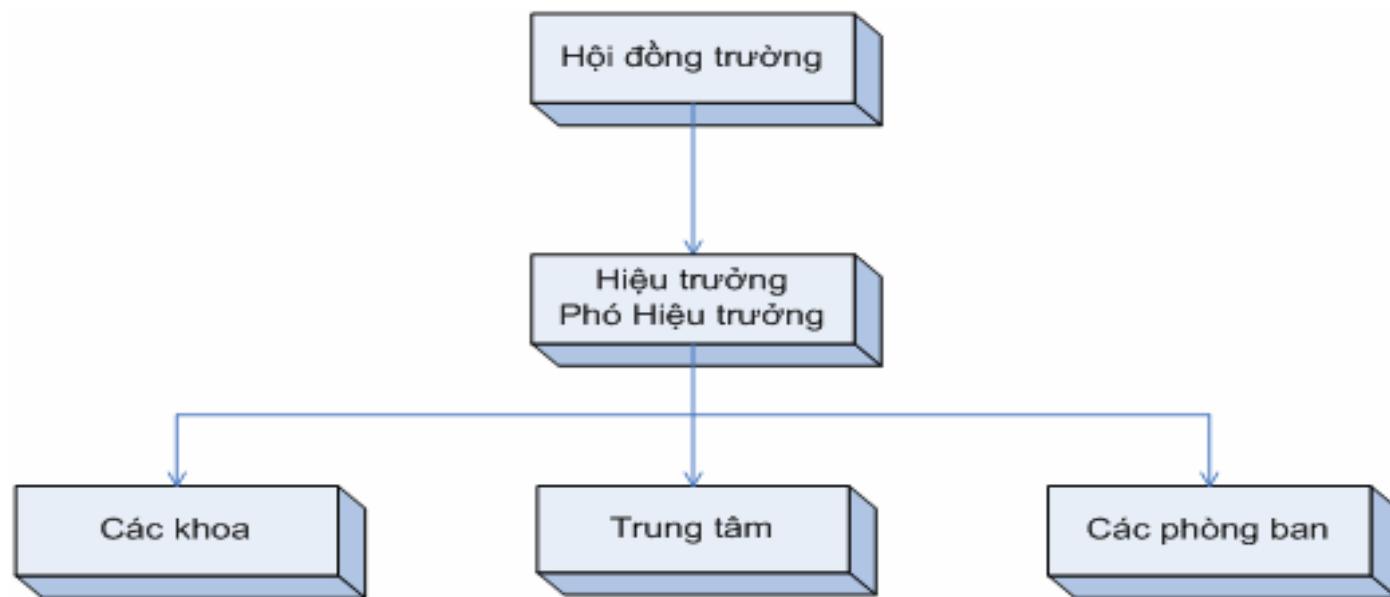
## V.1. Các khái niệm

- Đặc biệt: Cây nhị phân có thứ tự:  
Nếu một đỉnh trong có đủ 2 con thì
  - Con thứ nhất là **con bên trái** ( left child)
  - Con thứ 2 là **con bên phải** ( right child)
- Một m – cây với chiều cao h gọi là **thăng bằng** ( balanced) nếu tất cả các **lá** đều ở mức h hay h-1.

## V.1. Các khái niệm

### ❖ Một số ví dụ

- Mô hình gia phả một dòng họ
- Mô hình biểu diễn của các tổ chức
  - Ví dụ: [Mô hình tổ chức Trường Đại Học](#)



# V.1. Các khái niệm

## ❖ Một số ví dụ

- Mô hình các tập tin trong máy tính
  - Các tập tin trong máy tính được tổ chức thành các thư mục, các thư mục được tổ chức dưới dạng cây, trong đó thư mục gốc là gốc của cây



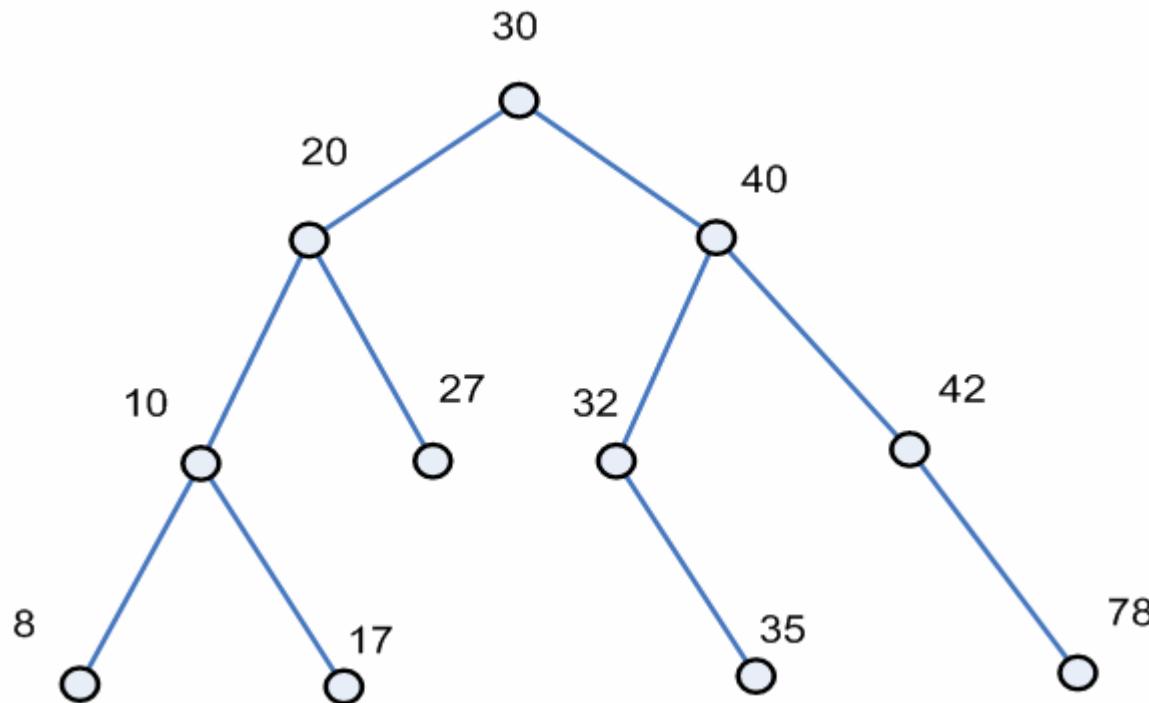
## V.2. Cây tìm kiếm nhị phân

❖ Một cây tìm kiếm nhị phân là một cây nhị phân  $T$  mà trong đó:

- Mỗi đỉnh được gán cho một nhãn
- Các nhãn có thể so sánh được với nhau
- $\forall$  đỉnh  $v \in T$ , các nhãn trong cây con bên trái của  $v$  đều nhỏ hơn nhãn của  $v$  và các nhãn trong cây con bên phải của  $v$  đều lớn hơn nhãn của  $v$

## V.2. Cây tìm kiếm nhị phân

❖ Ví dụ:: 30, 20, 10, 40, 32, 27, 17, 8, 42, 78, 35.



## V.2. Cây tìm kiếm nhị phân

### ❖ Thuật toán tìm kiếm trên cây tìm kiếm nhị phân

- Giả sử ta có một cây tìm kiếm,  $x$  là một giá trị nào đó
- Xác định vị trí của biến  $x$  nếu  $x$  là nhãn của một đỉnh  $v$
- Nếu thấy rằng  $x$  không là nhãn của một đỉnh nào cả thì tạo ra một đỉnh mới và gán nhãn  $x$  cho đỉnh đó
- Độ phức tạp thuật toán:  $O(\log n)$

## V.2. Cây tìm kiếm nhị phân

### ❖ Thuật toán tìm kiếm trên cây tìm kiếm nhị phân

```
void TK( Cây NPTK T, phần tử x);
{
    v = gốc của T;
    if (v == NULL ) thêm đỉnh r vào cây và gán cho nó nhãn là x
    while ((v != NULL) && (label(v) != x) )
    {
        if (x == label(v)) cout << “Tìm được x”;
        if (x < label(v))
            if (con bên trái v != NULL) v = con bên trái v;
            else thêm đỉnh nhãn x là con bên trái v và đặt v := NULL;
        if (x > label(v))
            if (con bên phải v != NULL) v = con bên phải v;
            else thêm đỉnh nhãn x là con bên phải v và đặt v:=NULL;
    }
}
```

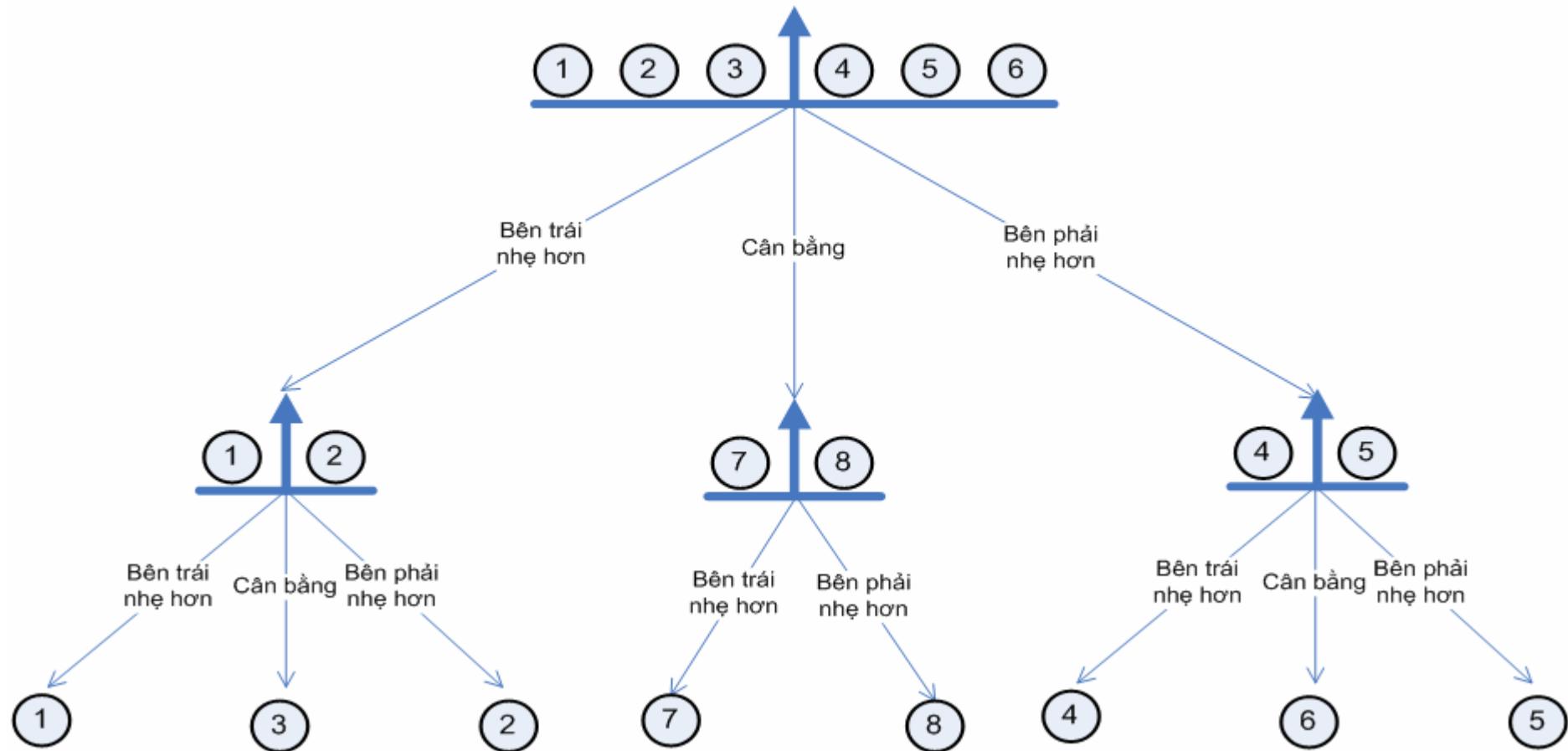
## V.3. Cây quyết định

- ❖ Thuật toán tìm kiếm trên cây tìm kiếm nhị phân
  - **Cây quyết định** là cây có gốc mà:
    - Mỗi đỉnh tương ứng với 1 quyết định
    - Mỗi cây con tại các đỉnh này ứng với mỗi kết cục có thể của của quyết định
  - **Một lời giải** là một đường đi từ gốc đến lá
  - Ví dụ: Cho 8 đồng xu, trong đó có một đồng nhẹ hơn. Xác định nó bằng 1 cái cân thăng bằng.

### V.3. Cây quyết định

- Có 3 trạng thái sau mỗi lần cân. Do đó cây quyết định cho một dãy các lần cân là **cây tam phân**
- Có ít nhất 8 lá trong cây quyết định vì có 8 kết cục có thể và mỗi kết cục cần biểu diễn bằng ít nhất 1 lá
- Số lần cân nhiều nhất để xác định đồng xu giả là chiều cao của cây h
- Ta có  $h \geq \lceil \log_3 8 \rceil = 2$  (làm tròn tăng)

## V.3. Cây quyết định



## V.4. Các phương pháp duyệt cây

- ❖ Thuật toán viếng thăm mọi đỉnh của một cây có gốc có thứ tự **đúng 1 lần** một cách có hệ thống gọi là **thuật toán duyệt cây**
- ❖ Có 3 thuật toán phổ thông:
  - Duyệt tiền tự (*Preorder traversal*)
  - Duyệt trung tự (*Inorder traversal*)
  - Duyệt hậu tự (*Postorder traversal*)

## V.4. Các phương pháp duyệt cây

- ❖ Thuật toán duyệt tiền tự

***void Preorder( cây thứ tự có gốc T);***

{

***r = gốc của T;***

***Thăm r;***

***for ( Mỗi cây con c của r từ trái sang phải )***

{

***T(c) = Cây con với gốc c***

***Preorder( T(c) )***

}

}

## V.4. Các phương pháp duyệt cây

- ❖ Thuật toán duyệt trung tự

*void Inorder( cây thứ tự có gốc T)*

{

*r := gốc của T*

*if (r là lá) Thăm r;*

*else*

{

*s = con đầu tiên từ trái sang phải của r*

*T(s) = Cây con với gốc s;*

*Inorder( T(s) ); Thăm r*

*for (Mỗi cây con c của r từ trái sang phải trừ s)*

*T(c) = Cây con với gốc c*

*Inorder( T(c) )*

}

}

## V.4. Các phương pháp duyệt cây

### ❖ Thuật toán duyệt hậu tự

**Void Postorder( cây thứ tự có gốc  $T$ );**

{

**$r = \text{gốc của } T$**

**for (Mỗi cây con c của  $r$  từ trái sang phải)**

{

**$T(c) = \text{Cây con với gốc c}$**

**$\text{Postorder}( T(c) )$**

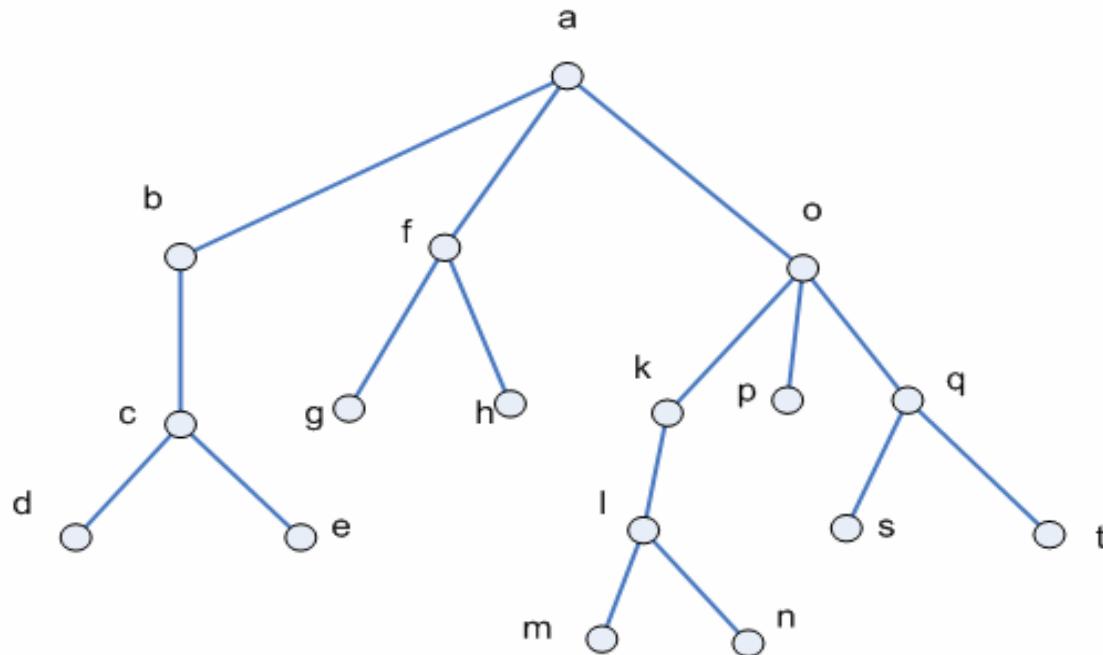
}

**$\text{Thăm } r$**

}

## V.4. Các phương pháp duyệt cây

### ❖ Ví dụ



- + Duyệt tiền tự: a, b, c, d, e, f, g, h, o, k, l, m, n, p, q, s, t
- + Duyệt trung tự: d, c, e, b, a, g, f, h, m, l, n, k, o, p, s, q, t
- + Duyệt hậu tự: d, e, c, b, g, h, f, m, n, l, k, p, s, t, q, o, a

## Chương 6: Bài toán tô màu đồ thị

# Nội dung



**I. Định nghĩa**



**II. Định lý 4 màu**



**III. Nhận biết đồ thị 2-màu**



**IV. Thuật toán SequentialColor**



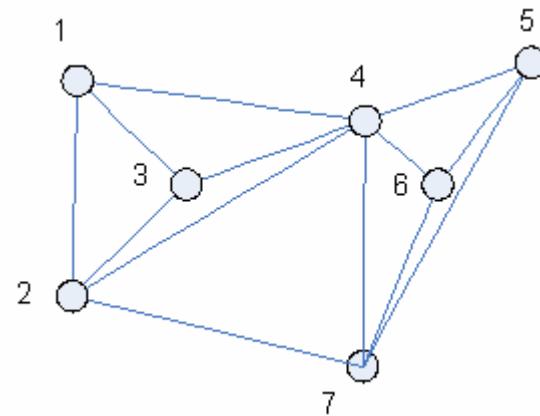
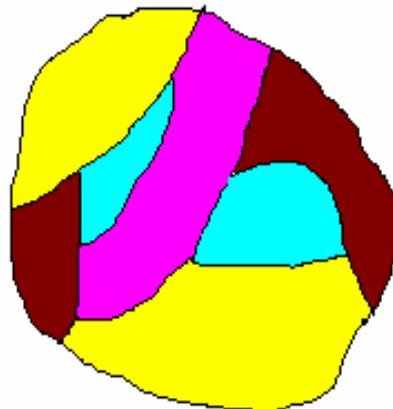
**V. Một số bài toán ứng dụng**

# I. Định nghĩa

## ❖ Cân phải tô màu một bản đồ với điều kiện:

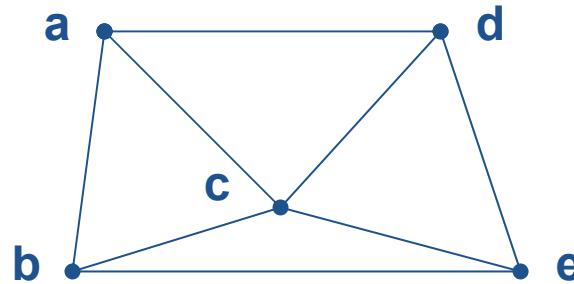
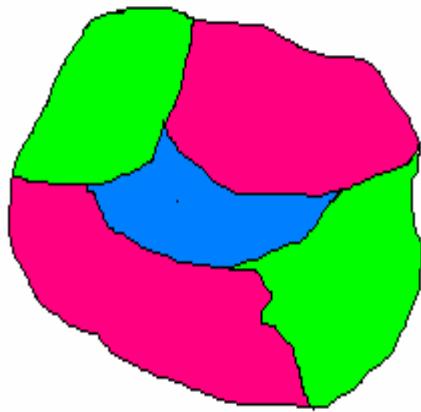
- Hai miền chung biên giới được tô hai màu khác nhau
- Số màu cần dùng là tối thiểu

## ❖ Hãy xác định số màu tối thiểu cho mọi bản đồ



Bản đồ này cần dùng 4 màu để tô

# I. Định nghĩa



Bài toán tô màu bản đồ quy về bài toán tô màu các **Đỉnh** của đồ thị

## Định nghĩa 1

Tô màu một đơn đồ thị là sự gán màu cho các đỉnh của nó sao cho hai đỉnh liền kề nhau được gán màu **khác nhau**.

## Định nghĩa 2

Số màu của một đồ thị là **số tối thiểu** các màu cần thiết để tô màu đồ thị này.

## II. Định lý 4 màu

❖ **Định lý:** Số màu của một đồ thị phẳng là không lớn hơn 4

- Định lý này được phát biểu lần đầu tiên năm 1850 và được 2 nhà toán học Mỹ Appel và Haken chứng minh năm 1976 bằng phản chứng.
- Đối với các đồ thị không phẳng số màu có thể tùy ý lớn
- Để chứng minh đồ thị  $G$  là  $n$ -màu ta phải
  - Chỉ ra 1 cách tô màu  $G$  với  $n$  màu
  - CMR không thể tô màu  $G$  với ít hơn  $n$  màu

## II. Định lý 4 màu

- ❖ Các bài toán tô màu đồ thị
  1. Cho đồ thị  $G$  và số nguyên  $k$ . Xây dựng một thuật toán để kiểm tra xem có thể tô màu  $G$  bằng  $k$  màu, nếu được thì thực hiện việc đó.
  2. Cho đồ thị  $G$  hãy xác định số màu  $k$  của đồ thị và hãy tô màu  $G$  bằng  $k$  màu đó

## II. Nhận biết đồ thị 2-màu

### ❖ Định lý

*Một đồ thị G là 2-màu khi và chỉ khi G không chứa một chu trình lẻ nào.*

### ❖ Chứng minh

1. Giả sử G là đồ thị 2-màu ta phải CMR G không chứa chu trình lẻ.

Thật vậy nếu G có chu trình lẻ  $C=(v_1, v_2, \dots, v_{2n+1}, v_1)$

Do C chỉ được tô bởi 2 màu  $\Rightarrow$  các đỉnh lẻ sẽ được tô bằng 1 màu. Nhưng lúc đó  $v_1$  và  $v_{2n+1}$  là 2 đỉnh kề nhau có cùng màu vô lý !!! (ĐPCM)

## II. Nhận biết đồ thị 2-màu

### ❖ Chứng minh

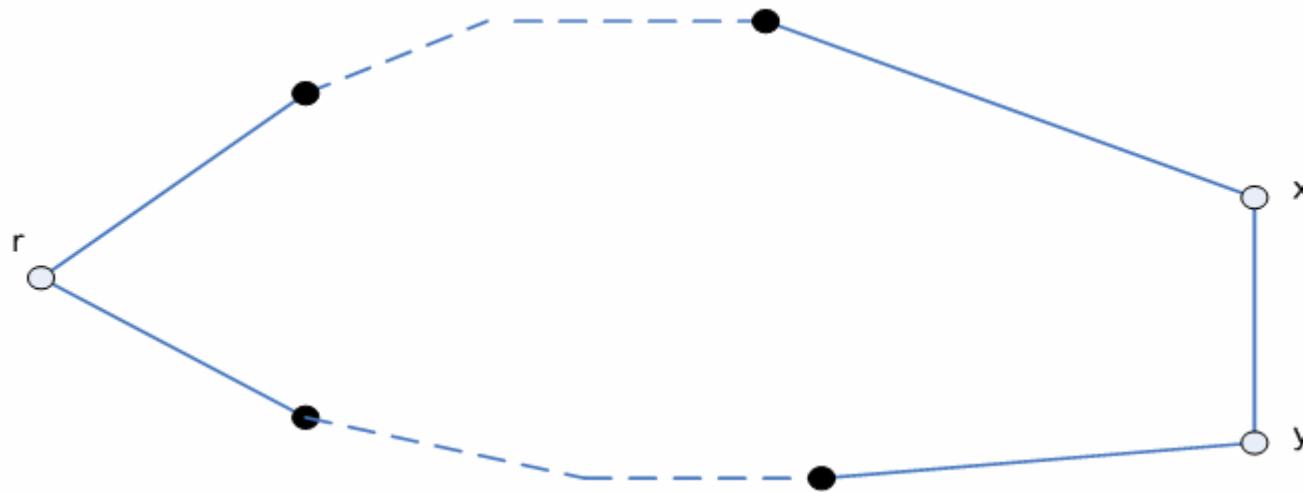
2. Giả sử  $G$  không chứa chu trình lẻ.Ta sẽ CMR  $G$  là đồ thị 2-màu.

- Chọn 1 đỉnh  $r$  làm gốc và tô nó màu đỏ.  $\forall x \in V$  sẽ được tô màu đỏ nếu đường đi ngắn nhất từ  $x$  tới  $r$  có số cạnh chẵn. Trái lại tô  $x$  màu xanh.
- Ta sẽ chứng minh rằng đỉnh  $x, y$  của cạnh  $(x,y)$  bất kỳ được tô hai màu khác nhau.
- **Trái lại** giả sử  $x$  và  $y$  là 2 đỉnh của cạnh  $(x,y)$  nào đó được tô cùng màu

## II. Nhận biết đồ thị 2-màu

### ❖ Chứng minh

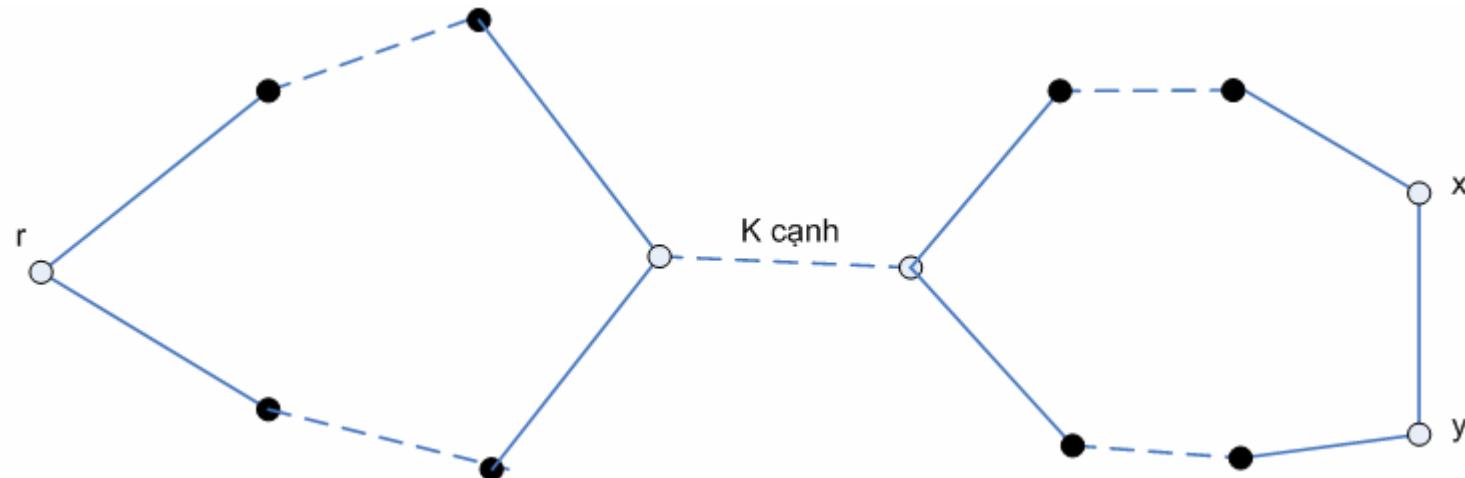
- Trường hợp 1:  $Px$  và  $Py$  không có chung cạnh. Ta có  $Px + (x,y) + Py$  là chu trình có số cạnh lẻ. (Mâu thuẫn giả thiết).



## II. Nhận biết đồ thị 2-màu

### ❖ Chứng minh

- Trường hợp 2: Px và Py có chung k cạnh từ đỉnh a tới đỉnh b. Ta sẽ nhận được hai chu trình Ca , Cb và k cạnh chung. Ta có Px + (x,y) + Py có số lẻ cạnh mà:  
 $|Px + (x,y) + Py| = |Ca| + |Cb| + 2k$



Do đó một trong hai chu trình Ca hoặc Cb sẽ có số cạnh lẻ  
Vô lý !!! (ĐPCM)  
Vậy G là 2 - màu

### III. Thuật toán SequentialColor

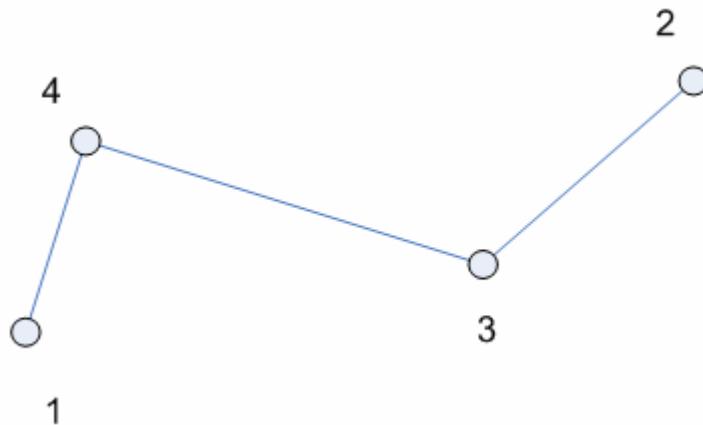
Với  $k=2$  việc nhận biết đồ thị 2 – màu đã được giải quyết  
Tuy vậy việc nhận biết đồ thị  $k$  – màu với  $k > 2$  vẫn chưa có  
lời giải

Thuật toán **SequentialColor** tô màu 1 đồ thị với  $k$  màu:  
Xem các đỉnh theo thứ tự từ 1 đến  $|V|$ , tại mỗi đỉnh  $v$  gán màu  
đầu tiên có sẵn mà chưa được gán cho 1 đỉnh nào liền v

1. Xếp các đỉnh theo thứ tự bất kỳ  $1, 2, \dots, n$
2. Tạo tập  $L_i$  - tập các màu có thể gán cho đỉnh  $i$
3. Bắt đầu tô từ đỉnh 1
4. Với đỉnh  $k \in \{1, \dots, n\}$  tô màu đầu tiên của  $L_k$  cho  $k$
5.  $\forall j > k$  và  $j$  kề k loại bỏ trong  $L_j$  màu đã được tô cho  $k$
6. Giải thuật dừng lại khi tất cả các đỉnh đã được tô

### III. Thuật toán SequentialColor

#### ❖ Ví dụ



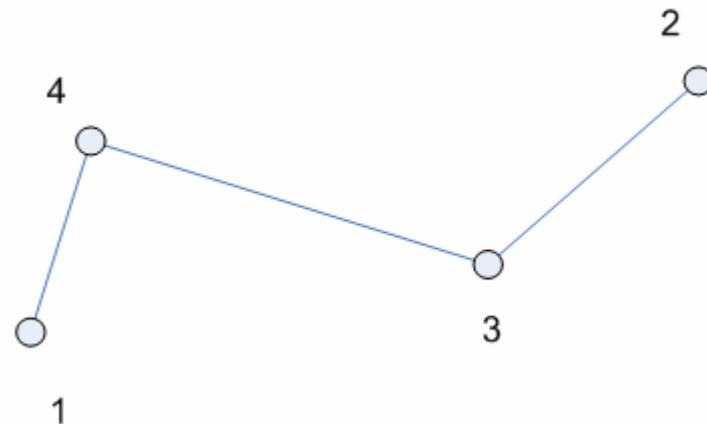
Các màu:  
X: Xanh  
Đ: Đỏ  
T: Tím  
V: Vàng

Thứ tự tô các đỉnh: 1, 2, 3, 4

Các bước	L1	L2	L3	L4	Màu tô
Khởi tạo	X, Đ, T, V				
B1	X	X, Đ, T, V	X, Đ, T, V	Đ, T, V	1 - Xanh
B2		X	Đ, T, V	Đ, T, V	2 - Xanh
B3			Đ	T, V	3 - Đỏ
B4				T	4 - Tím

### III. Thuật toán SequentialColor

#### ❖ Ví dụ



Các màu:  
X: Xanh  
Đ: Đỏ  
T: Tím  
V: Vàng

Thứ tự tô các đỉnh: 4, 3, 2, 1

Các bước	L4	L3	L1	L2	Màu tô
Khởi tạo	X, Đ, T, V				
B1	X	Đ, T, V	Đ, T, V	X, Đ, T, V	4 - Xanh
B2		Đ	Đ, T, V	X, T, V	3 - Đỏ
B3			Đ	X, T, V	1 - Đỏ
B4				X	2 - Xanh

### III. Thuật toán SequentialColor

#### ❖ Nhận xét

- Là dạng thuật toán tham lam  $\rightarrow$  Lời giải tìm được chưa chắc tối ưu
- Độ phức tạp của giải thuật  $O(n^2)$

## IV. Một số bài toán ứng dụng

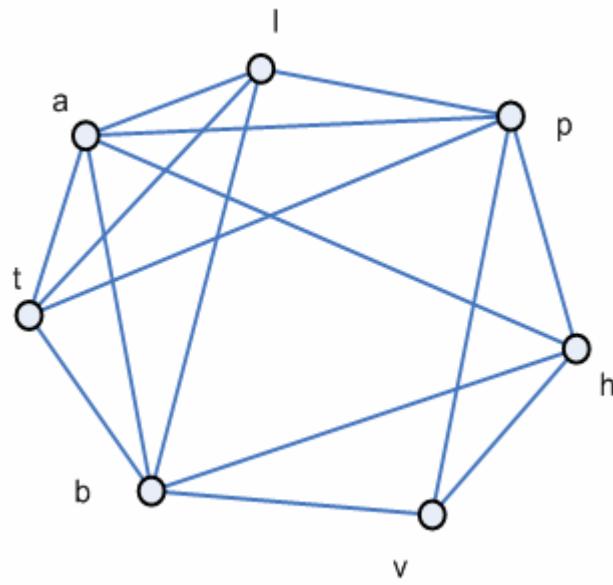
### ❖ Bài toán lập lịch thi

- Lập lịch thi: Hãy lập lịch thi trong trường đại học sao cho không có sinh viên nào có 2 môn thi cùng lúc
  - Các đỉnh : Các môn thi
  - Có 1 cạnh nối 2 đỉnh nếu như có 1 SV thi cả 2 môn này
  - Thời gian thi được thể hiện bởi các màu khác nhau
- Việc lập lịch thi sẽ tương ứng với việc tô màu đồ thị này

# IV. Một số bài toán ứng dụng

## ❖ Bài toán lập lịch thi

- Có 7 môn thi: Toán (t), Anh Văn (a), Lý (l), Pascal (p), Tin học đại cương (h), Tiếng vệt thực hành (v), Visual Basic (b).
- Các cặp môn thi có chung sinh viên là: (t,a), (t, l), (t, p), (t,b),(a,l), (a,p), (a,h), (a,b), (l,p), (l,b), (p,h), (p,v), (h,b), (v,b).



Kết quả tô màu

a	l	p	h	v	b	t
Đỏ	Xanh	Tím	Nâu	Lá cây	Vàng	Đen

Kết quả xếp lịch thi

Đợt thi	Môn thi
1	Anh Văn
2	Lý, Tin học đại cương
3	Pascal, Visual Basic
4	Tiếng vệt thực hành, Toán

## IV. Một số bài toán ứng dụng

### ❖ Bài toán phân chia tần số

- Phân chia tần số: Các kênh truyền hình từ số 2 tới 13 được phân chia cho các đài truyền hình ở Bắc Mỹ sao cho 2 đài ở gần nhau dưới 150 km có 2 kênh khác nhau
- Giải quyết:
  - Mỗi đài phát : 1 đỉnh
  - Hai đài gần nhau dưới 150 km là 2 đỉnh được nối với nhau
  - Việc phân chia kênh: Tô màu đồ thị, trong đó mỗi màu biểu thị một kênh

## Chương 7: Bài toán tìm đường đi ngắn nhất

# Nội dung

- I. Giới thiệu
- II. Thuật toán Ford-Bellman
- III. Thuật toán Dijkstra
- IV. Thuật toán Floyd

# I. Giới thiệu

- ❖ Xét đồ thị có hướng, có trọng số  $G=(V, E)$

$$TrongSo(u, v) = \begin{cases} \infty & , if (u, v) \notin E \\ a(u, v), & if (u, v) \in E \end{cases}$$

Với  $a(u, v) \in \mathbb{R}$

- ❖ Nếu dãy  $v_0, v_1, \dots, v_p$  là 1 đường đi trên  $G$  thì **độ dài** của nó được định nghĩa:

$$DoDai(v_0, v_1, \dots, v_p) = \sum_{i=1}^p a(v_{i-1}, v_i)$$

# I. Giới thiệu

## ❖ Bài toán đường đi ngắn nhất

- Giả sử có nhiều đường đi từ  $v_0$  đến  $v_p$ : Đường đi ngắn nhất là đường đi có tổng trọng số các cung nhỏ nhất.
- Đường đi từ một đỉnh
  - Ford-Bellman
  - Dijkstra
- Đường đi từ một đỉnh
  - Floyd

# Nội dung

- I. Giới thiệu
- II. Thuật toán Ford-Bellman
- III. Thuật toán Dijkstra
- IV. Thuật toán Floyd

## II. Thuật toán Ford-Bellman

- ❖ Thuật toán Ford-Bellman dùng để tìm đường đi ngắn nhất từ một đỉnh s đến tất cả các đỉnh còn lại của đồ thị.
- ❖ Được sử dụng cho đồ thị không có chu trình âm.

Cho đồ thị có hướng, có trọng số  $G=(V, E)$ . Trọng số của các cạnh của  $G$  được tính như sau:

$\text{TrongSo}(u, v) = \infty$  nếu cung  $(u, v) \notin E$ .

$\text{TrongSo}(u, v) = a(u, v)$  nếu cung  $(u, v) \in E$ .

Thuật toán tìm đường đi ngắn nhất  $d(v)$  từ đỉnh s đến v, mọi  $v \in V$ :

+ Xét u  $\in V$ . Nếu  $d(u) + \text{TrongSo}(u, v) < d(v)$  thì ta thay  $d(v) = d(u) + \text{TrongSo}(u, v)$ .

+ Quá trình này sẽ được lặp lại cho đến khi không thể có giá trị  $d(v)$  tốt hơn.

## II. Thuật toán Ford-Bellman

### ❖ Cài đặt thuật toán

#### ■ Đầu vào:

- Đồ thị có hướng  $G=(V,E)$  với  $n$  đỉnh.
- $s \in V$  là đỉnh xuất phát.
- $a[u,v]$ ,  $u,v \in V$  là ma trận trọng số

#### ■ Đầu ra :

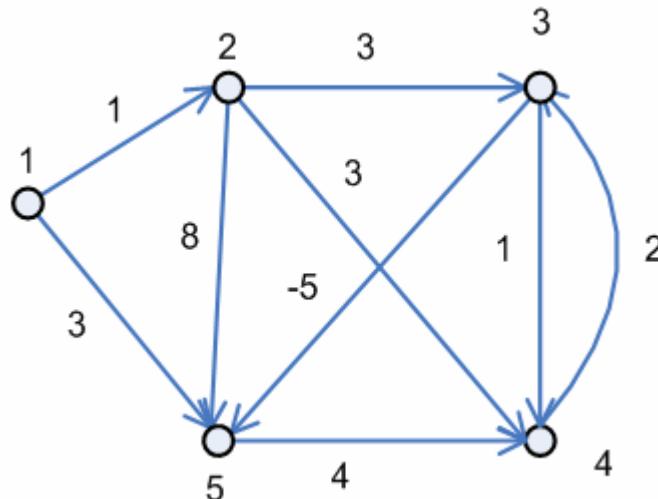
- Khoảng cách từ  $s$  đến tất cả các đỉnh còn lại  $d[v]$ ,  $v \in V$ .
- $\text{Truoc}[v]$ ,  $v \in V$  là đỉnh đi trước  $v$  trong đường đi ngắn nhất từ  $s$  đến  $v$

## II. Thuật toán Ford-Bellman

```
void Ford_Bellman()
{
    for (v ∈ V) /* Khởi tạo d và Truoc */
    {
        d[v] = a[s,v];
        Truoc[v] = s;
    }
    d[s] = 0;
    for (k = 1; k < n-1; k++)
        for (v ∈ V \ {s})
            for (u ∈ V)
                if (d[v] > d[u] + a[u,v] )
                {
                    d[v] = d[u] + a[u,v] ;
                    Truoc[v] = u;
                }
    } /* Độ phức tạp của thuật toán là O(n3) */
```

## II. Thuật toán Ford-Bellman

### ❖ Ví dụ



1	2	3	4	5
1	$\infty$	1	$\infty$	$\infty$
2	$\infty$	$\infty$	3	3
3	$\infty$	$\infty$	$\infty$	1
4	$\infty$	$\infty$	2	$\infty$
5	$\infty$	$\infty$	$\infty$	4

k	d[5], Truoc[5]	d[4], Truoc[4]	d[3], Truoc[3]	d[2], Truoc[2]
1	3, 1	$\infty$ , 1	$\infty$ , 1	1, 1
2	3, 1	4, 2	4, 2	1, 1
3	-1, 3	4, 2	4, 2	1, 1
4	-1, 3	3, 5	4, 2	1, 1
5	-1, 3	3, 5	4, 2	1, 1

# Nội dung

- I. Giới thiệu
- II. Thuật toán Ford-Bellman
- III. Thuật toán Dijkstra
- IV. Thuật toán Floyd

### III. Thuật toán Dijkstra

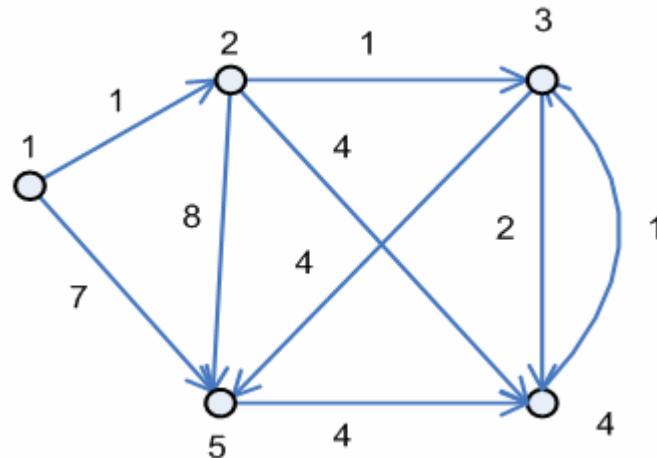
- ❖ Thuật toán Dijkstra dùng để tìm đường đi ngắn nhất từ đỉnh s đến các đỉnh còn lại trong đồ thị.
- ❖ Được sử dụng cho đồ thị không có cung trọng số âm.
- ❖ Thuật toán
  - Đầu vào
    - Đồ thị có hướng  $G=(V,E)$  với  $n$  đỉnh.
    - $s \in V$  là đỉnh xuất phát.
    - $a[u,v]$ ,  $u,v \in V$  là ma trận trọng số
  - Đầu ra
    - Khoảng cách từ s đến tất cả các đỉnh còn lại  $d[v]$ ,  $v \in V$ .
    - $Truoc[v]$ ,  $v \in V$  là đỉnh đi trước v trong đường đi ngắn nhất từ s đến v.

### III. Thuật toán Dijkstra

```
void Dijkstra;{
    for (v ∈ V) /* Khởi tạo d và Truoc */ {
        d[v] = a[s,v];
        Truoc[v] = s;
    }
    d[s] = 0; T = V \ {s};
    while (T != ∅) {
        Tìm u ∈ T sao cho d(u) = min { d(z): z ∈ T }
        T = T \ {u}; /* Cố định nhãn của u */
        for (v ∈ T) do
            if (d[v] > d[u] + a[u,v] ) then
            {
                d[v] = d[u] + a[u,v] ;
                Truoc[v] = u;
            }
        }
    } /* Độ phức tạp của thuật toán là O(n2) */
}
```

### III. Thuật toán Dijkstra

#### ❖ Ví dụ



	1	2	3	4	5
1	$\infty$	1	$\infty$	$\infty$	7
2	$\infty$	$\infty$	1	4	8
3	$\infty$	$\infty$	$\infty$	2	4
4	$\infty$	$\infty$	1	$\infty$	$\infty$
5	$\infty$	$\infty$	$\infty$	4	$\infty$

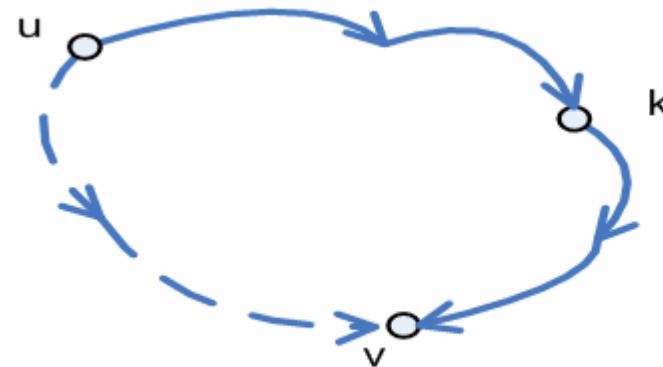
T	Đỉnh 2	Đỉnh 3	Đỉnh 4	Đỉnh 5
2, 3, 4, 5	1, 1	$\infty$ , 1	$\infty$ , 1	7, 1
3, 4, 5		2, 2	5, 2	7, 1
4, 5			4, 3	6, 3
E				6, 3
$\emptyset$	1, 1	2, 2	4, 3	6, 3

# Nội dung

- I. Giới thiệu
- II. Thuật toán Ford-Bellman
- III. Thuật toán Dijkstra
- IV. Thuật toán Floyd

## IV. Thuật toán Floyd

- ❖ Tìm đường đi ngắn nhất giữa tất cả các cặp đỉnh trong đồ thị.
- ❖ Thuật toán
  - Với mọi đỉnh k của đồ thị xét theo thứ tự từ 1 đến n, xét mọi cặp đỉnh u, v. Ta tìm đường đi ngắn nhất từ u đến v theo công thức:
$$a(u, v) = \min (a(u, v), a(u, k) + a(k, v))$$



## IV. Thuật toán Floyd

### ❖ Cài đặt

- Đầu vào
  - Đồ thị cho bởi ma trận trọng số:  $a[i, j]$ ,  $i, j = 1, 2, \dots, n$ .
- Đầu ra: Hai ma trận
  - Ma trận đường đi ngắn nhất giữa các cặp đỉnh:  
 $d[i, j]$ ,  $i, j = \underline{1..n}$ .  
 $d[i, j]$  là độ dài đường đi ngắn nhất từ  $i$  đến  $j$
  - Ma trận ghi nhận đường đi.  
 $p[i, j]$ ,  $i, j = \underline{1..n}$ .  
 $p[i, j]$  ghi nhận đỉnh đi trước đỉnh  $j$  trong đường đi ngắn nhất từ  $i$  đến  $j$ .

## IV. Thuật toán Floyd

```
void Floyd{
```

```
    for (i = 1; i <= n; i++) /* Khởi tạo */
```

```
        for (j = 1; j <= n; j++) {
```

```
            d[i,j] = a[i,j];
```

```
            p[i,j] = i;
```

```
}
```

```
    for (k = 1; k <= n; k++) /* 3 vòng lặp */
```

```
        for (i = 1; i <= n; i++)
```

```
            for (j = 1; j <= n; j++)
```

```
                if (d[i,j] > d[i,k] + d[k,j])
```

```
{
```

```
            d[i,j] = d[i,k] + d[k,j];
```

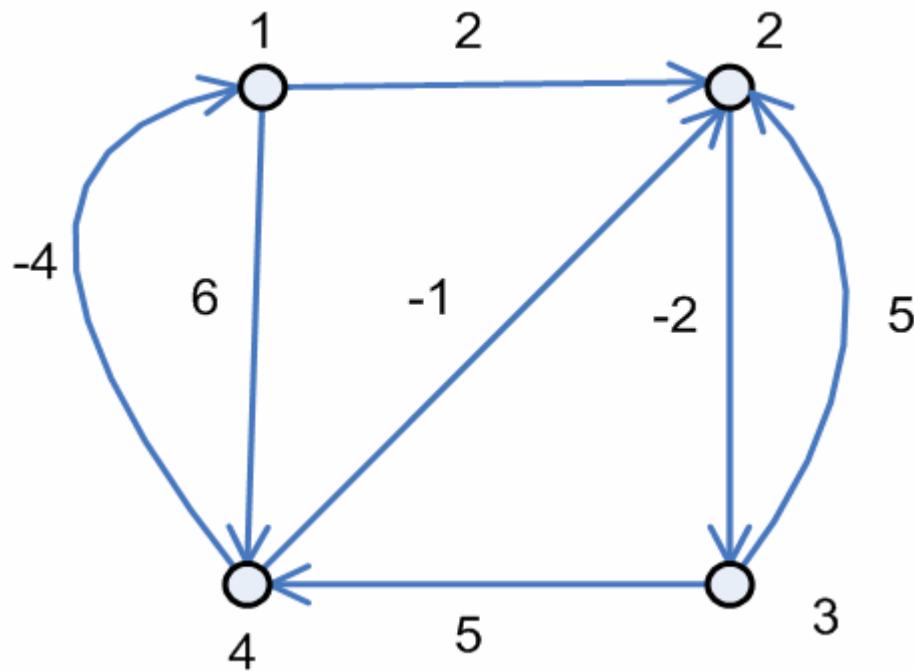
```
            p[i,j] = p[k,j];
```

```
}
```

```
}
```

## IV. Thuật toán Floyd

### ❖ Ví dụ



# IV. Thuật toán Floyd

- Khởi tạo hai ma trận:

D: Ma trận lưu giá trị đường đi.

P: Ma trận lưu vết.

D0				
	1	2	3	4
1	0	2	0	6
2	0	0	-2	0
3	0	5	0	5
4	-4	-1	0	0

P0				
	1	2	3	4
1	0	1	0	1
2	0	0	2	0
3	0	3	0	3
4	4	4	0	0

- k=3

D3				
	1	2	3	4
1	0	2	0	5
2	0	0	-2	3
3	0	5	0	5
4	-4	-2	-4	0

P3				
	1	2	3	4
1	0	1	2	3
2	0	0	2	3
3	0	3	0	3
4	4	1	2	0

Kết quả các bước lặp của thuật toán:

- k=1

D1				
	1	2	3	4
1	0	2	0	6
2	0	0	-2	0
3	0	5	0	5
4	-4	-2	0	0

P1				
	1	2	3	4
1	0	1	0	1
2	0	0	2	0
3	0	3	0	3
4	4	1	0	0

- k=4

D4				
	1	2	3	4
1	0	2	0	5
2	-1	0	-2	3
3	1	5	0	5
4	-4	-2	-4	0

P4				
	1	2	3	4
1	0	1	2	3
2	4	0	2	3
3	4	1	0	3
4	4	1	2	0

- k=2

D2				
	1	2	3	4
1	0	2	0	6
2	0	0	-2	0
3	0	5	0	5
4	-4	-2	-4	0

P2				
	1	2	3	4
1	0	1	2	1
2	0	0	2	0
3	0	3	0	3
4	4	1	2	0

Vậy đường đi ngắn nhất từ đỉnh 1 đến đỉnh 3 là:  $1 \rightarrow 2 \rightarrow 3$ .  
Với trọng số = 0.

## Chương 8: Luồng trong mạng

# Nội dung

- I. Bài toán luồng cực đại
- II. Định lý Ford-Fulkerson
- III. Thuật toán tìm luồng cực đại trong mạng

# I. Bài toán luồng cực đại

## ❖ Mạng

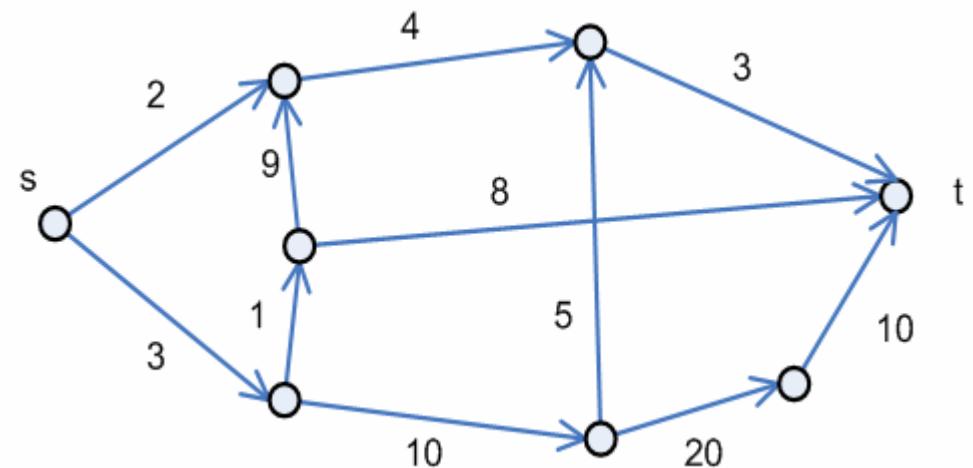
Mạng là một đồ thị có hướng  $G = (V, E)$

- $\exists!$  đỉnh **s** (Điểm phát) mà  $\deg^-(s) = 0$
- $\exists!$  đỉnh **t** (Điểm thu) mà  $\deg^+(t) = 0$
- $\forall$  cung **e** =  $(v, w) \in E$  **được gán** với một số *không âm*  $c(e) = c(v, w) \geq 0$  gọi là **Khả năng thông qua** của cung e.

**s** : Điểm phát

**t** : Điểm thu

Nếu không có cung  $(v, w)$  thì  $c(v, w) = 0$



# I. Bài toán luồng cực đại

## ❖ Luồng trong mạng

- Cho mạng  $G = (V, E)$ , ta gọi **luồng  $f$**  trong mạng  $G$  là một ánh xạ  $f: E \rightarrow \mathbb{R}^*$ , với mọi cung  $e = (v, w) \in E$  được gán với một số không âm  $f(e) = f(v, w) \geq 0$  gọi là **luồng trên cung  $e$** , thỏa mãn các điều kiện sau:
  - Luồng trên mỗi cung  $e \in E$  không vượt quá khả năng thông qua của nó:  $0 \leq f(e) \leq c(e)$
  - Với mọi đỉnh  $v$  không trùng với đỉnh phát  $s$ , và đỉnh thu  $t$ , tổng luồng trên các cung đi vào  $v$  bằng tổng luồng các cung đi ra khỏi  $v$ .

$$Div_f(v) = \sum_{w \in \Gamma^-(v)} f(w, v) - \sum_{w \in \Gamma^+(v)} f(v, w) = 0$$

Với

$$\Gamma^-(v) = \{w \in V \mid (w, v) \in E\}$$

$$\Gamma^+(v) = \{w \in V \mid (v, w) \in E\}$$

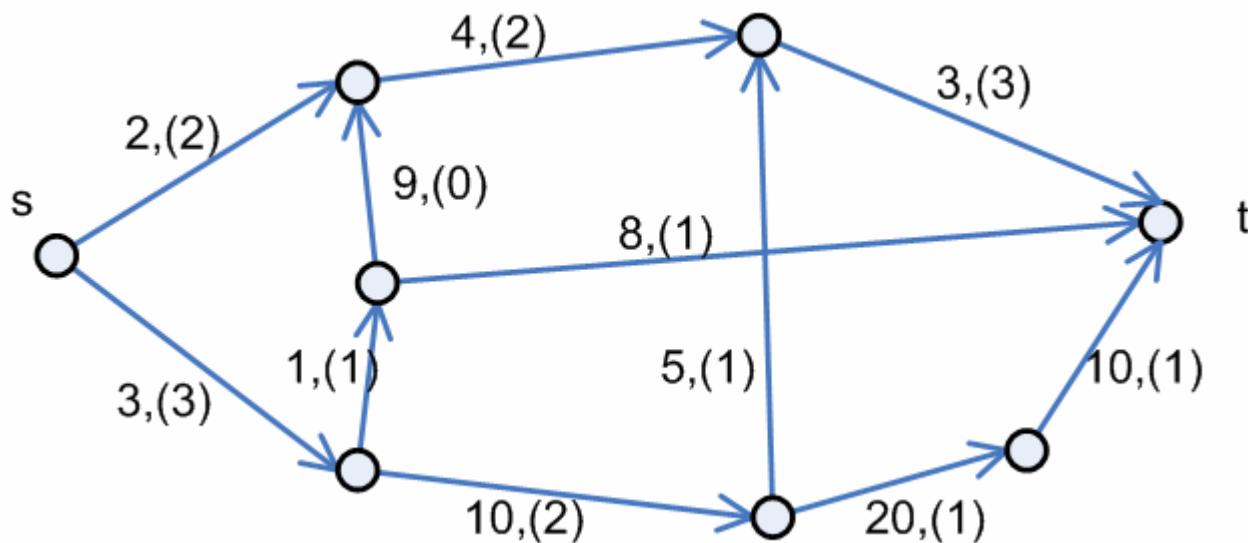
Điều kiện cân  
bằng luồng

# I. Bài toán luồng cực đại

## ❖ Luồng trong mạng

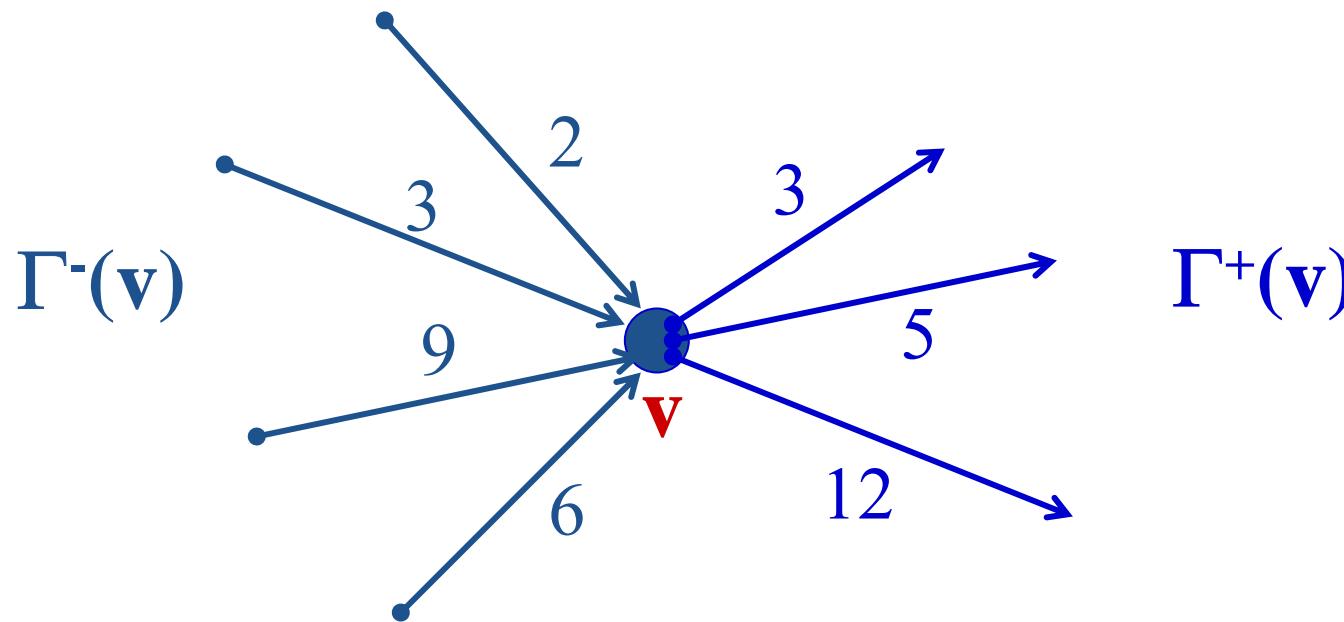
- **Giá trị của luồng  $f$**  là tổng luồng trên các cung đi ra khỏi đỉnh phát (bằng tổng luồng trên các cung đi vào đỉnh thu).

$$val(f) = \sum_{w \in \Gamma^+(s)} f(s, w) = \sum_{w \in \Gamma^-(t)} f(w, t)$$



# I. Bài toán luồng cực đại

## ❖ Luồng trong mạng



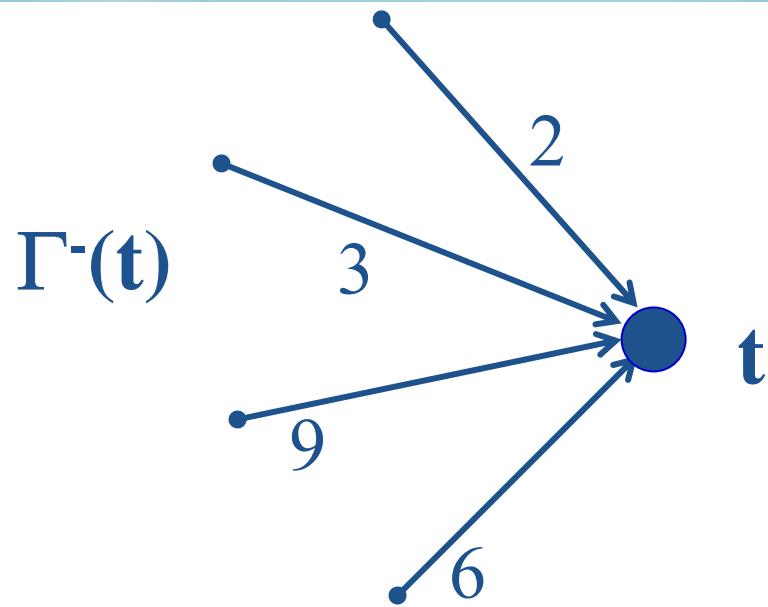
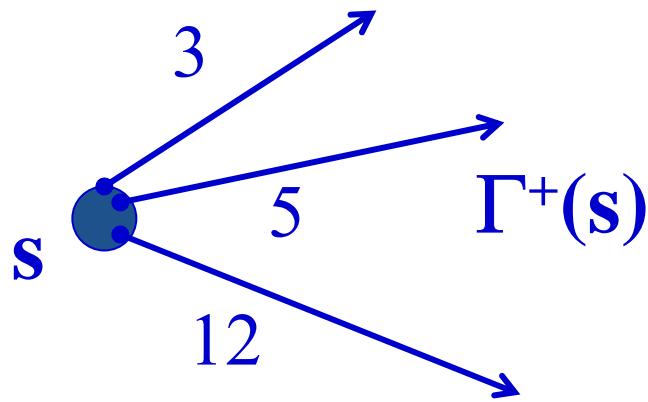
$$\sum_{w \in \Gamma^-(v)} f(w, v) = 2 + 3 + 9 + 6 = 20$$

$$\sum_{w \in \Gamma^+(v)} f(v, w) = 3 + 5 + 12 = 20$$

$$\text{Div}_f(v) = 20 - 20 = 0$$

# I. Bài toán luồng cực đại

## ❖ Luồng trong mạng



$$\sum_{w \in \Gamma^-(t)} f(w, t) = 2 + 3 + 9 + 6 = 20$$

$$\sum_{w \in \Gamma^+(s)} f(s, w) = 3 + 5 + 12 = 20$$

$$\text{val } (f) = 20$$

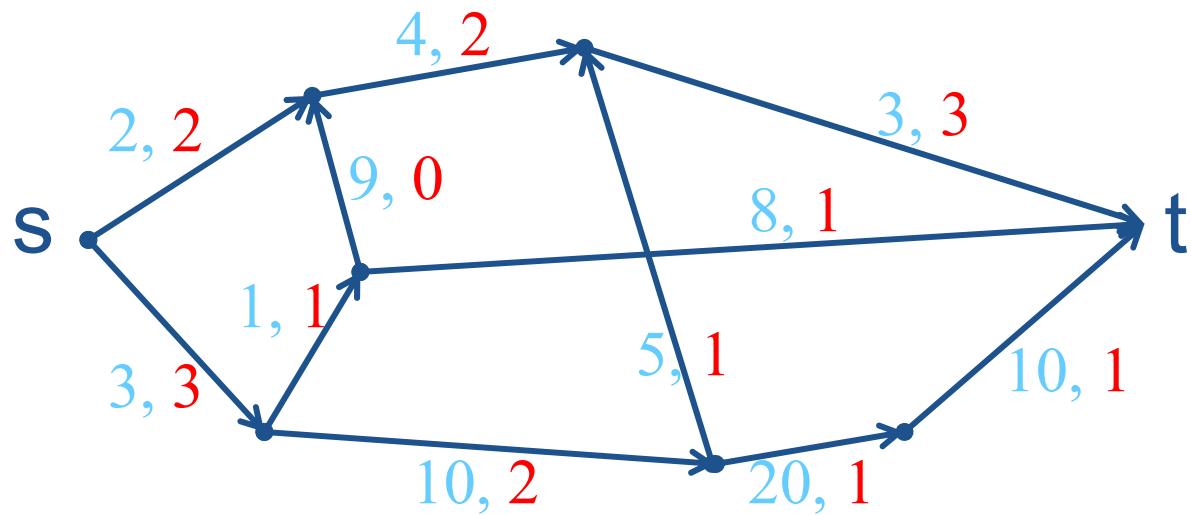
# I. Bài toán luồng cực đại

Các số màu xanh: Khả năng thông qua trên mỗi cung

Các số màu đỏ: Luồng trên mỗi cung

Giá trị của luồng:

$$\text{val}(f) = 5$$



$s$  : Điểm phát  
 $t$  : Điểm thu  
Nếu không có  
cung  $(v, w)$  thì  
 $c(v, w) = 0$

# I. Bài toán luồng cực đại

## ❖ Bài toán luồng cực đại

- Cho mạng  $G = (V, E)$ , hãy tìm luồng  $f$  trong mạng sao cho giá trị luồng là lớn nhất.
- Luồng  $f$  như vậy gọi là **luồng cực đại**

## ❖ Ứng dụng:

- Bài toán lập bản đồ giao thông trong thành phố.
- Bài toán đám cưới vùng quê.

# Nội dung

I.

Bài toán luồng cực đại

II.

Định lý Ford-Fulkerson

III.

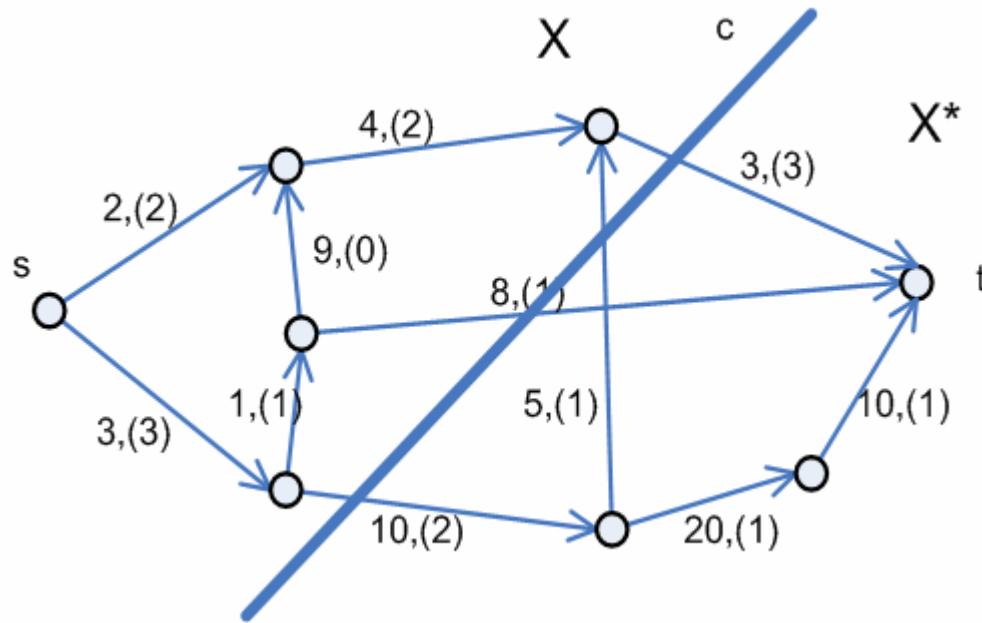
Thuật toán tìm luồng cực đại trong mạng

## II.1. Lát cắt

- Cho mạng  $G = (V, E)$ . **Lát cắt**  $(X, X^*)$  là một phân hoạch tập đỉnh  $V$  của mạng thành hai tập  $X$  và  $X^*$  với điểm phát  $s \in X$  và điểm thu  $t \in X^*$ .
- **Khả năng thông qua** của lát cắt  $(X, X^*)$  là tổng tất cả các khả năng thông qua của các cung  $(v, w)$  có  $v \in X$  và  $w \in X^*$ .
- Lát cắt với **khả năng thông qua** nhỏ nhất được gọi là **lát cắt hẹp nhất**.

## II.1. Lát cắt

### ❖ Lát cắt



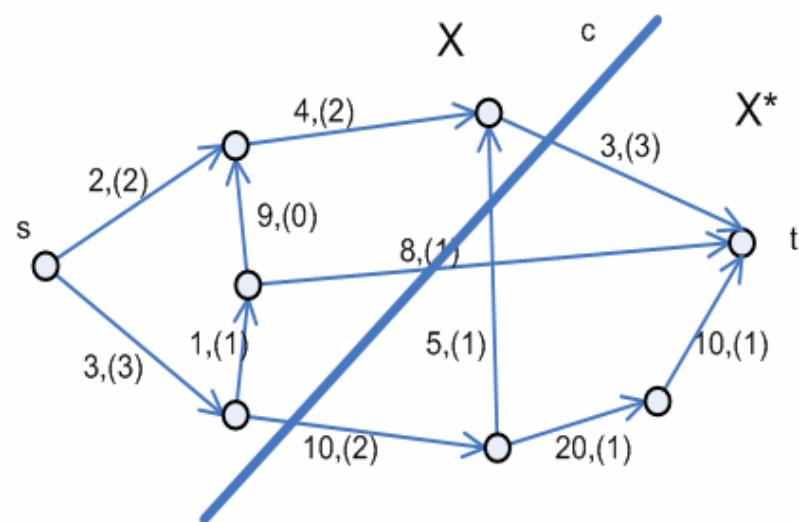
Khả năng thông qua của lát cắt  $(X, X^*)$  là:  $3 + 8 + 10 = 21$ .

## II.2. Luồng và lát cắt

### ❖ Định lý 1

Giá trị của mọi luồng  $f$  trong mạng không lớn hơn khả năng thông qua của lát cắt bất kỳ  $(X, X^*)$ .

$$val(f) \leq c(X, X^*)$$



Khả năng thông qua là 21.  
Giá trị của luồng  $f$ :  $val(f)=5 < 21$ .

## II.2. Luồng và lát cắt

### ❖ Định lý 1

### ❖ Chứng minh

- Với mọi  $v \in V$ , ta cộng các điều kiện cân bằng luồng:

$$\sum_{v \in X} \left( \sum_{w \in \Gamma^-(v)} f(w, v) - \sum_{w \in \Gamma^+(v)} f(v, w) \right) = \text{div}(s) = -\text{val}(f).$$

- Tổng này gồm các số hạng dạng  $f(u, v)$  với dấu + và dấu - mà có ít nhất  $u$  hoặc  $v \in X$ . Nếu cả  $u$  và  $v$  đều  $\in X$  thì  $f(u, v)$  sẽ xuất hiện với dấu + trong  $\text{Div}(v)$  và dấu - trong  $\text{Div}(u)$  nên chung triệt tiêu lẫn nhau. Ta thu được:

$$- \sum_{v \in X, w \in X^*} f(v, w) + \sum_{v \in X^*, w \in X} f(v, w) = -\text{val}(f)$$

$$\Leftrightarrow \text{val}(f) = \sum_{v \in X, w \in X^*} f(v, w) - \sum_{v \in X^*, w \in X} f(v, w) \leq \sum_{v \in X, w \in X^*} c(v, w)$$

$$\Leftrightarrow \text{val}(f) \leq c(X, X^*)$$

(ĐPCM).

## II.2. Luồng và lát cắt

### ❖ Hệ quả

Giá trị luồng cực đại trong mạng không vượt quá khả năng thông qua của lát cắt hẹp nhất trong mạng.

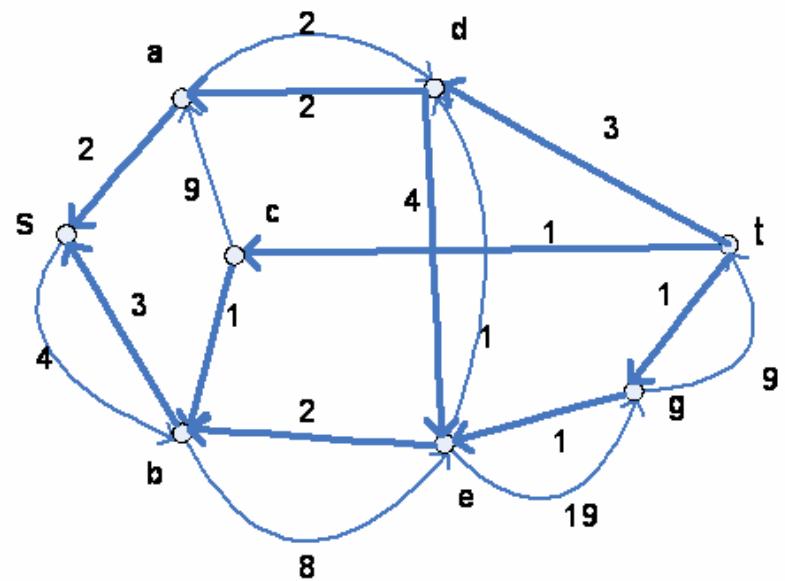
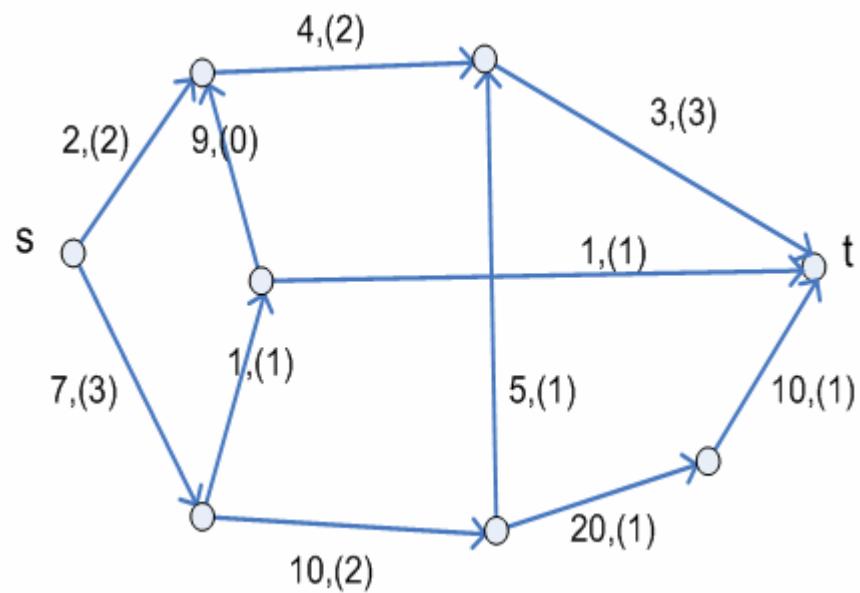
### ❖ Định lý Ford-Fulkerson

*Giá trị luồng cực đại trên mạng đúng bằng khả năng thông qua của lát cắt hẹp nhất.*

## II.3. Đồ thị tăng luồng, đường tăng luồng

- ❖ Giả sử  $f$  là một luồng trong mạng  $G = (V, E)$ . Từ mạng  $G$  ta xây dựng đồ thị có trọng số  $G_f = (V, E_f)$  như sau:
  - Xét các cạnh  $e = (v, w) \in E$ :
    - Nếu  $f(v, w) = 0$  : thêm một cung  $(v, w)$  có trọng số là  $c(v, w)$  vào  $G_f$ .
    - Nếu  $f(v, w) = c(v, w)$  : thêm một cung  $(w, v)$  có trọng số  $c(v, w)$  vào  $G_f$ .
    - Nếu  $0 < f(v, w) < c(v, w)$  : thêm một cung  $(v, w)$  có trọng số  $c(v, w) - f(v, w)$ , và một cung  $(w, v)$  có trọng số  $f(v, w)$  vào  $G_f$  .
  - Các cung của đồng thời cũng là cung của  $G$  được gọi là **cung thuận**, các cung còn lại được gọi là **cung nghịch**. Đồ thị được gọi là **đồ thị tăng luồng**.

### II.3. Đồ thị tăng luồng, đường tăng luồng



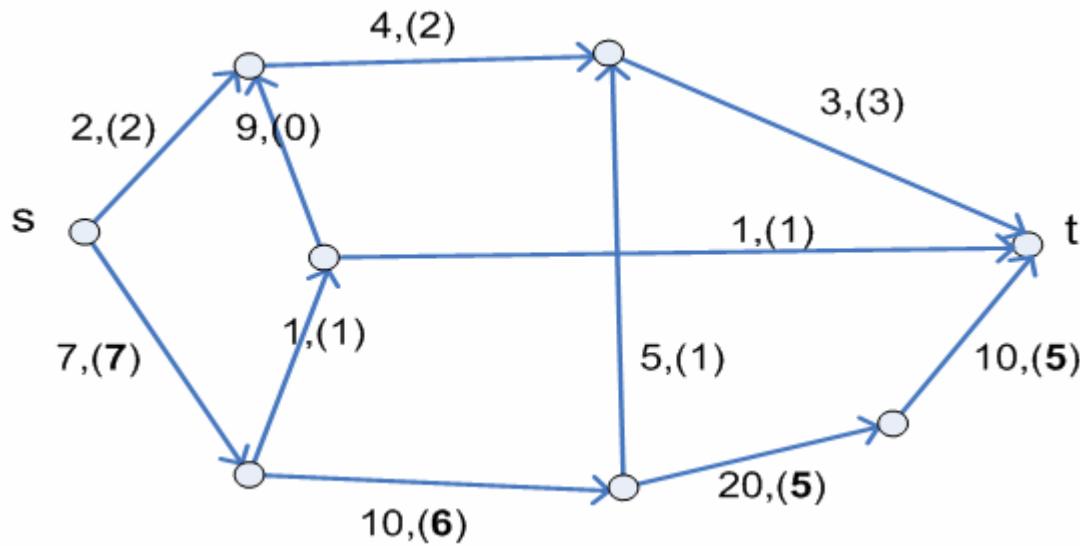
## Mạng G=(V, E)

## Đồ thị tăng luồng $G_f = (V, E_f)$

## II.3. Đồ thị tăng luồng, đường tăng luồng

- ❖ Giả sử  $P = (s, , t)$  là một đường đi từ  $s$  đến  $t$  trên đồ thị tăng luồng. Gọi  $d$  là trọng số nhỏ nhất trong các trọng số của các cung trên đường đi  $P$ . Từ luồng  $f$ , xây dựng luồng  $f'$  trên mạng  $G$  như sau:
  - Nếu  $(v, w) \in P$  là cung thuận thì  $f'(v, w) = f(v, w) + d$ .
  - Nếu  $(v, w) \in P$  là cung nghịch thì  $f'(v, w) = f(v, w) - d$ .
  - Nếu  $(v, w) \notin P$  thì  $f'(v, w) = f(v, w)$ .
- ❖ Khi đó ta được luồng  $f'$  là luồng trong mạng  $G$  và giá trị của luồng  $f'$  tăng thêm  $d$  so với giá trị của luồng  $f$ . Đường đi  $P$  được gọi là **đường tăng luồng**.

## II.3. Đồ thị tăng luồng, đường tăng luồng



## II.3. Đồ thị tăng luồng, đường tăng luồng

### ❖ Định lý 2

- Cho mạng  $G=(V, E)$  và  $f$  là một luồng trong mạng  $G$ . Các mệnh đề sau là tương đương
  - $f$  là luồng cực đại trong mạng.
  - Không tìm được đường tăng luồng  $f$ .
  - $\text{val}(f) = c(X, X^*)$ , với  $(X, X^*)$  là một lát cắt nào đó của mạng.

Chứng minh?

# Nội dung

- I. Bài toán luồng cực đại
- II. Định lý Ford-Fulkerson
- III. Thuật toán tìm luồng cực đại trong mạng

### III. Thuật toán tìm luồng cực đại trong mạng

#### ❖ Qui trình thuật toán Ford-Fulkerson

- Đặt luồng ban đầu bằng 0 (luồng không). Vì một mạng bất kỳ đều có ít nhất một luồng là luồng không.
- Lặp lại hai quá trình tìm đường tăng luồng và tăng luồng cho mạng theo đường tăng luồng đó. Vòng lặp kết thúc khi không tìm được đường tăng luồng nữa.
- Khi đã có luồng cực đại, xây dựng lát cắt hẹp nhất của mạng.

### III. Thuật toán tìm luồng cực đại trong mạng

#### ❖ Thuật toán tìm đường tăng luồng

- Đầu tiên, gán nhãn cho s và đặt nó là chưa xét. Tiếp tục ta gán nhãn cho các đỉnh kề của s và s trở thành đỉnh đã xét. Làm tương tự cho các đỉnh kề với s đã được gán nhãn. Thuật toán dừng lại nếu:
  1. Đỉnh t được gán nhãn. Khi đó ta tìm được đường tăng luồng.
  2. Hoặc t chưa có nhãn mà tất cả các đỉnh có nhãn khác đã được xét. Khi đó luồng đang xét là cực đại, không tìm được đường tăng luồng.

### III. Thuật toán tìm luồng cực đại trong mạng

Bước 1: Đặt  $f(e)=0$ , với mọi cạnh  $e \in E$

Bước 2: Gán nhãn cho s:

$p[s]=[-, \varepsilon(s)]$ ;

$\varepsilon(s)=\infty$ ;

Đặt  $u=s$ ;

Bước 3:

a) Với mọi  $v \in K_{e+}(u)$ , Nếu v chưa có nhãn và  $s(u,v)=c(u,v)f(u,v)>0$  thì:

Đặt  $\varepsilon(v) = \min(\varepsilon(u), s(u,v))$ ;

Gán nhãn  $p[v] = [ +u, \varepsilon(v)]$  ;

Với mọi  $v \in K_{e-}(u)$ , Nếu v chưa có nhãn và  $f(u,v)>0$  thì:

Đặt  $\varepsilon(v) = \min (\varepsilon(u), f(u,v))$ ;

Gán nhãn  $p[v] = [ -u, \varepsilon(v)]$  ;

Bước 4: Nếu t đã có nhãn ( $v == t$ ) Đến Bước 5.

Ngược lại :

Nếu Mọi đỉnh có nhãn đã xét: Đến Bước 6.

Ngược lại: đặt  $u=v$ , Đến Bước 3.

Cuối nếu.

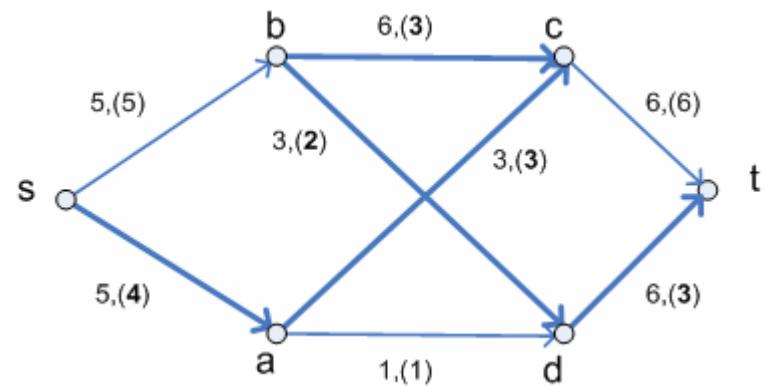
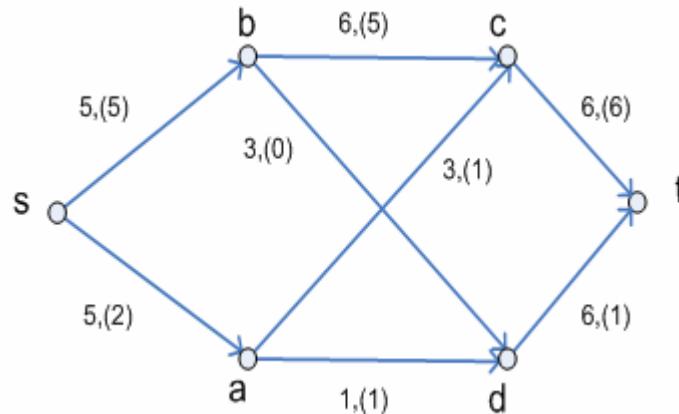
Cuối nếu.

Bước 5: Dùng  $p[t]$  để tìm đường tăng luồng P bằng cách đi ngược từ t đến s. Đặt  $f = f + \varepsilon(t)$   $\forall$  cạnh  $e \in P$ . Đến Bước 2.

Bước 6:  $X = \{\text{Các đỉnh có nhãn đã xét}\}$ ,  $X^* = V \setminus X$ . Lát cắt  $(X, X^*)$  là cực tiểu.

### III. Thuật toán tìm luồng cực đại trong mạng

#### ❖ Ví dụ



- + Gán nhãn:  $s [-, \infty]$ .
  - + Xét  $s$ : cung  $(s, a)$   $s(s, a) = 3 > 0$ :  $\varepsilon(a) = \min(\infty, 3) = 3$ ,  $p[a] = [+s, 3]$ .  
Đỉnh  $b$ : Chưa được gán nhãn.
  - + Xét  $a$ :  $p[c] = [+a, 2]$
  - + Xét  $c$ : cung  $(b, c)$   $f(b, c) = 5 > 0$ ,  $\varepsilon(c) = \min(2, 5) = 2$      $p[b] = [-c, 2]$
  - + Xét  $b$ :  $p[d] = [+b, 2]$ .
  - + Xét  $d$ :  $p[t] = [+d, 2]$ .
- Ta có đường tăng luồng:  
 $t \rightarrow d \rightarrow b \rightarrow c \rightarrow a \rightarrow s$   
Luồng  $f' := f + 2 = 7 + 2 = 9$ .



**LOGIC M NH**

# Contents

1. M nh
2. S t ng ng c a các m nh
3. V ng và l ng t

# 1. M nh

- M nh là m t câu úng ho c sai, ch khÔng th v a úng v a sai.

(*m nh*)

- Hà n i là th ô c a Vi t Nam.
- $1 + 5 = 70$

(*Không ph i m nh*)

- $x + y = z$  (khÔng úng – khÔng sai)
- Bây gi là m y gi ? (câu tr n thu t)

## M nh (cont.)

- Các ch cái s c dùng kí hi u m nh và các bi n : p, q, r, s ... Giá tr chân lý c a m nh là **úng/sai**, kí hi u T (F).
- “Các nh lu t c a t duy” – Geogre Boole (1854) => các m nh ph c h p c t o t m nh hi n có b ng cách dùng các toán t logic.
- nh ngh a 1: Gi s p là m nh .
- Câu “không ph i là p”
- Là 1 m nh khác, c g i là ph nh c a p. kí hi u  $\neg p$  ho c  $\overline{p}$

# Toán tử logic

BẢNG 1. Bảng giá trị chân lý đối với phủ định của một mệnh đề

$p$	$\neg p$
T	F
F	T

- Ví dụ: tìm phím cảm nhận: “Hôm nay là thứ mấy”.
- Giải: “Hôm nay **không phải** là thứ mấy”

Ví d : Hôm nay là th t . Hôm nay tr i m a => “Hôm nay th t và tr i m a” (toán t h i,  $p \wedge q$ )

**ĐỊNH NGHĨA 2.** Giả sử  $p$  và  $q$  là hai mệnh đề. Mệnh đề " $p$  và  $q$ ", được ký hiệu bởi  $p \wedge q$ , là đúng khi cả  $p$  và  $q$  đều đúng, còn sai trong các trường hợp còn lại. Mệnh đề  $p \wedge q$  được gọi là *hội* của  $p$  và  $q$ .

BẢNG 2. Bảng giá trị chân lý đối với hội của hai mệnh đề

$p$	$q$	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

**ĐỊNH NGHĨA 3.** Cho  $p$  và  $q$  là hai mệnh đề. Mệnh đề : " $p$  hoặc  $q$ ", được ký hiệu là  $p \vee q$ , là mệnh đề sai khi cả  $p$  và  $q$  đều sai, và đúng trong các trường hợp còn lại. Mệnh đề  $p \vee q$  được gọi là *tuyến* của  $p$  và  $q$ .

Ví d :

- Món khai v súp ho c salat.
- Các sinh viên ngành CNTT ho c Toán ng d ng có th theo h c h c ph n LT T.

BÀNG 3. Bảng giá trị chân lý đối với tuyến của hai mệnh đề		
$p$	$q$	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

# Mệnh tuy n lo i

**ĐỊNH NGHĨA 4.** Cho  $p$  và  $q$  là hai mệnh đề.

Mệnh đề *tuyển loại* của  $p$  và  $q$ , được ký hiệu là  $p \oplus q$ , là một mệnh đề chỉ đúng khi một trong  $p$  và  $q$  là đúng và sai trong mọi trường hợp còn lại.

BÀNG 4. Bảng giá trị chân lý đối với phép tuyển loại của hai mệnh đề

$P$	$q$	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

# Mệnh đề kéo theo $p \rightarrow q$

**ĐỊNH NGHĨA 5.** Cho  $p$  và  $q$  là hai mệnh đề.

Mệnh đề kéo theo  $p \rightarrow q$  là một mệnh đề chỉ sai khi  $p$  đúng và  $q$  sai, còn đúng trong mọi trường hợp còn lại.

Trong phép kéo theo nói trên  $p$  được gọi là *giả thiết* còn  $q$  được gọi là *kết luận*.

BÀNG 5. Bảng giá trị chân lý đối với phép  
kéo theo  $p \rightarrow q$

$p$	$q$	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

# Mệnh đề

**ĐỊNH NGHĨA 6.** Cho  $p$  và  $q$  là hai mệnh đề. Mệnh đề *tương đương*  $p \leftrightarrow q$  là mệnh đề chỉ đúng khi  $p$  và  $q$  có cùng giá trị chân lý và sai trong mọi trường hợp còn lại.

BÀNG 6. Bảng giá trị chân lý đối với mệnh đề  
tương đương  $p \leftrightarrow q$

$p$	$q$	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

## Dịch nghĩa câu thông thường

- Tiếng Anh (Viết ...) thường có tính không rõ ràng. Dịch các câu thông thường sang biểu thức logic là làm **mất tính không rõ ràng của nó**.  
nhưng thường có thể xác định giá trị chân lý, thao tác và các quy tắc suy diễn suy luận chúng.
- Ví dụ: “Bịt không cản lái xe máy nếu bùn cao dưới 1.5m trên phiến trên 18 tuổi”.

## G i ý

$q = B$  n c lái xe máy

$r = B$  n cao d i 1.5m

$s = B$  n trên 18 tu i.

Bí u th c logic:

- $(r \wedge \neg s) \rightarrow \neg q$
- Và 1 s cách khác ... t ng ng

# Các phép toán logic và các phép toán BIT

- Binary bit (0, 1) - John Tukey (nhà th ng kê), 1946. 1 = true; 0 = false.

BẢNG 7. Bảng cho các toán tử bit OR, AND và XOR

v	0	1	$\wedge$	0	1	$\oplus$	0	1
0	0	1	0	0	0	0	0	1
1	1	1	1	0	1	1	1	0

**ĐỊNH NGHĨA 7.** Một xâu bit (hoặc xâu nhị phân) là dây không hoặc nhiều bit. Chiều dài của xâu là số các bit trong xâu đó.

**Ví dụ 9.** 101010011 là một xâu bit có chiều dài là 9.

01101 10110

11000 11101

---

11101 11111

∨ OR bit

01000 10100

∧ AND bit

10101 01011

⊕ XOR bit.

## Bài tập

1. Trong các câu dưới đây câu nào là một mệnh đề?
  - a) Boston là thủ phủ của bang Massachusetts.
  - b) Miami là thủ phủ của bang Florida.
  - c)  $2 + 3 = 5$
  - d)  $5 + 7 = 10$
  - e)  $x + 2 = 11$
  - f) Hãy trả lời câu hỏi này.
  - g)  $x + y = y + x$  với mọi cặp số thực  $x$  và  $y$ .

2. Trong các câu sau đây câu nào là một mệnh đề? Xác định giá trị chân lý của các mệnh đề đó?

- a) Không được đi qua.
- b) Bây giờ là mấy giờ?
- c) Không có ruồi đen ở Maine.
- d)  $4 + x = 5$ .
- e)  $x + 1 = 5$  nếu  $x = 1$ .
- f)  $x + y = y + z$  nếu  $x = z$ .

3. Tìm phủ định của các mệnh đề sau :

- a) Hôm nay là thứ năm.
- b) Không có ô nhiễm ở New Jersey.
- c)  $2 + 1 = 3$ .
- d) Mùa hè ở Maine nóng và nắng.

4. Cho  $p$  và  $q$  là hai mệnh đề.

$p$  : Tôi đã mua vé xổ số tuần này.

$q$  : Tôi đã trúng giải độc đắc 1 triệu đô la vào hôm thứ sáu.

Diễn đạt các mệnh đề sau bằng các câu thông thường :

a)  $\neg p$

b)  $p \vee q$

c)  $p \rightarrow q$

d)  $p \wedge q$

e)  $p \Leftrightarrow q$

f)  $\neg p \rightarrow \neg q$

5. Cho  $p$  và  $q$  là hai mệnh đề.

$p$  : Nhiệt độ dưới không.

$q$  : Tuyết rơi..

Dùng  $p$  và  $q$  và các liên từ logic viết các mệnh đề sau :

- a) Nhiệt độ dưới không và tuyết rơi.
- b) Nhiệt độ dưới không nhưng không có tuyết rơi.
- c) Nhiệt độ không dưới không và không có tuyết rơi.
- d) Có tuyết rơi hoặc nhiệt độ dưới không (hoặc cả hai).
- e) Nếu nhiệt độ dưới không thì cũng có tuyết rơi.
- f) Hoặc nhiệt độ dưới không hoặc có tuyết rơi nhưng sẽ không có tuyết rơi nếu nhiệt độ dưới không.
- g) Nhiệt độ dưới không là điều kiện cần và đủ để có tuyết rơi.

6. Cho  $p$ ,  $q$  và  $r$  là những mệnh đề :

$p$  : Bạn bị cúm.

$q$  : Bạn thi trượt kỳ thi cuối khoá.

$r$  : Bạn được lên lớp.

Hãy diễn đạt những mệnh đề sau thành những câu thông thường.

a)  $p \rightarrow q$

b)  $\neg q \leftrightarrow r$

c)  $q \rightarrow \neg r$

d)  $p \vee q \vee r$

e)  $(p \rightarrow \neg r) \vee (q \rightarrow \neg r)$

f)  $(p \wedge q) \vee (\neg q \wedge r)$ .

7. Cho  $p$  và  $q$  là hai mệnh đề

$p$  : Bạn lái xe với tốc độ trên 65 dặm/h.

$q$  : Bạn bị phạt vì vượt quá tốc độ cho phép.

Hãy viết các mệnh đề sau bằng cách dùng  $p$  và  $q$  và các liên từ logic.

- a) Bạn không lái xe với tốc độ trên 65 dặm/h.
- b) Bạn lái xe với tốc độ trên 65 dặm/h, nhưng bạn không bị phạt vì vượt quá tốc độ cho phép.
- c) Bạn sẽ bị phạt vì vượt quá tốc độ cho phép nếu bạn lái xe với tốc độ trên 65 dặm/h.
- d) Nếu bạn không lái xe với tốc độ trên 65 dặm/h thì bạn sẽ không bị phạt vì vượt quá tốc độ cho phép.

8. Cho  $p$ ,  $q$  và  $r$  là các mệnh đề

$p$  : Bạn nhận được điểm giỏi trong kỳ thi cuối khoá.

$q$  : Bạn làm hết các bài tập trong quyển sách này.

$r$  : Bạn sẽ được công nhận là giỏi ở lớp này.

Hãy dùng  $p, q$  và  $r$  cùng với các liên từ logic để viết các mệnh đề sau:

- a) Bạn được công nhận là giỏi ở lớp này, nhưng bạn không làm hết các bài tập ở quyển sách này.
- b) Bạn nhận được điểm giỏi ở kỳ thi cuối khoá, bạn làm hết các bài tập trong quyển sách này và bạn được công nhận là giỏi ở lớp này.
- c) Để được công nhận là giỏi ở lớp này bạn cần phải được điểm giỏi ở kỳ thi cuối khoá.
- d) Bạn nhận được điểm giỏi ở kỳ thi cuối khoá, nhưng bạn không làm hết các bài tập ở quyển sách này, tuy nhiên bạn vẫn được công nhận là giỏi ở lớp này.
- e) Nhận được điểm giỏi ở kỳ thi cuối khoá và làm hết những bài tập ở quyển sách này là đủ để bạn được công nhận là giỏi ở lớp này.
- f) Bạn sẽ được công nhận là giỏi ở lớp này, nếu và chỉ nếu bạn làm hết các bài tập trong quyển sách này hoặc nhận được điểm giỏi ở kỳ thi cuối khoá.

9. Đối với các câu sau đây, hãy cho biết các câu đó sẽ có ý nghĩa nào nếu liên từ hoặc ở đây có sắc thái nghĩa bao hàm (tức là tuyển) so với liên từ hoặc có sắc thái nghĩa loại trừ? Theo bạn trong hai nghĩa đó, nghĩa nào là nghĩa hàm định?
- a) Để theo học môn toán học rời rạc, bạn cần phải đã học giải tích hoặc một khoá tin học.
  - b) Khi bạn mua một chiếc xe mới của hãng Acme Motor bạn sẽ được bớt lại 2000 USD tiền mặt hoặc được nợ 2% giá trị chiếc xe.
  - c) Bữa ăn tối gồm hai món ở cột A hoặc ba món ở cột B.
  - d) Trường sẽ đóng cửa nếu tuyết rơi dày hơn 2m hoặc gió lạnh dưới -100.

10. Một nhà thám hiểm bị một nhóm người ăn thịt người bắt cóc. Có hai loại người ăn thịt người : loại luôn luôn nói thật và loại luôn luôn nói dối. Họ sẽ nướng sống nhà thám hiểm nếu Ông không xác định được một người nào đó trong họ là luôn luôn nói dối hay nói thật. Ông được phép hỏi người đó chỉ một câu hỏi.
- Hãy giải thích tại sao câu hỏi "Anh là người nói dối?" không mang lại kết quả?
  - Tìm câu hỏi mà nhà thám hiểm đã dùng để xác định người ăn thịt người đó là luôn luôn nói dối hay nói thật.

11. Hãy viết những câu sau dưới dạng "nếu  $p$  thì  $q$ "

(Gợi ý : Tham khảo các cách thường dùng để diễn đạt phép kéo theo đã được liệt kê trong Tiết này.

- a) Có tuyết rơi mỗi khi có gió Đông Bắc.
  - b) Các cây táo sẽ nở hoa nếu trời ấm kéo dài một tuần.
  - c) Việc đội Pistons dành chức vô địch ngũ ý ràng họ đã đánh bại đội Lakers.
  - d) Cần phải đi 8 dặm nữa mới tới được đỉnh núi Long.
  - e) Để được phong giáo sư, nổi tiếng thế giới là đủ.
  - f) Nếu bạn cho xe chạy hơn 400 dặm, bạn sẽ cần phải mua xăng.
  - g) Giấy bảo hành của bạn còn hiệu lực nếu bạn đã mua chiếc đầu CD của bạn ít hơn 90 ngày trước đây.
-

12. Viết các mệnh đề sau dưới dạng " $p$  nếu và chỉ nếu  $q$ " trong ngôn ngữ thông thường.
- a) Để nhận được điểm giỏi trong khoá học này cần và đủ là phải học giải được các bài tập của toán học rời rạc.
  - b) Nếu bạn đọc háo mỗi ngày bạn sẽ thạo tin tức và ngược lại.
  - c) Trời mưa nếu là ngày cuối tuần và là ngày cuối tuần nếu trời mưa.
  - d) Bạn có thể nhìn thấy lão phù thuỷ nếu lão không ở trong đó và lão phù thuỷ không ở trong đó nếu bạn nhìn thấy lão.

13. Phát biểu mệnh đề đảo và phản đảo của các mệnh đề kéo theo sau :

- a) Nếu hôm nay tuyết rơi, ngày mai tôi sẽ đi trượt tuyết.
- b) Tôi tới lớp mỗi khi sắp có kỳ thi.
- c) Một số nguyên dương là số nguyên tố nếu nó không có một ước số nào khác 1 và chính nó.

14. Phát biểu mệnh đề đảo và phản đảo của các mệnh đề kéo theo sau :

- a) Nếu đêm nay có tuyết rơi, tôi sẽ ở nhà.
- b) Tôi đều đi ra bãi tắm bất cứ ngày nào trời nắng.
- c) Khi tôi ở lại muộn, cần phải để tôi ngủ đến trưa.

15. Lập bảng giá trị chân lý đối với các mệnh đề phức hợp sau :

a)  $p \wedge \neg p$

b)  $p \vee \neg p$

c)  $(p \vee \neg q) \rightarrow q$

d)  $(p \vee q) \rightarrow (p \wedge q)$

e)  $(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$

f)  $(p \rightarrow q) \rightarrow (q \rightarrow p).$

16. Lập bảng giá trị chân lý cho các mệnh đề phức hợp sau :

a)  $p \oplus q$

b)  $p \oplus \neg p$

c)  $p \oplus \neg q$

d)  $\neg p \oplus \neg q$

e)  $(p \oplus q) \vee (p \oplus \neg q)$

f)  $(p \oplus q) \wedge (p \oplus \neg q)$

**17. Lập bảng giá trị chân lý cho các mệnh đề phức hợp sau**

a)  $p \rightarrow \neg q$

h)  $\neg p \leftrightarrow q$

c)  $(p \rightarrow q) \vee (\neg p \rightarrow q)$

d)  $(p \rightarrow q) \wedge (\neg p \rightarrow q)$

e)  $(p \leftrightarrow q) \vee (\neg p \leftrightarrow q)$

f)  $(\neg p \leftrightarrow \neg q) \leftrightarrow (p \leftrightarrow q)$

**18. Lập bảng giá trị chân lý cho các mệnh đề phức hợp sau :**

a)  $(p \vee q) \vee \neg r$

b)  $(p \vee q) \wedge r$

c)  $(p \wedge q) \vee r$

d)  $(p \wedge q) \wedge r$

e)  $(p \vee q) \wedge \neg r$

f)  $(p \wedge q) \vee \neg r$

19. Lập bảng giá trị chân lý cho các mệnh đề phức hợp sau :

a)  $p \rightarrow (\neg q \vee r)$

h)  $\neg p \rightarrow (q \rightarrow r)$

c)  $(p \rightarrow q) \vee (\neg p \rightarrow r)$

d)  $(p \rightarrow q) \wedge (\neg p \rightarrow r)$

e)  $(p \leftrightarrow q) \vee (\neg q \leftrightarrow r)$

f)  $(\neg p \leftrightarrow \neg q) \leftrightarrow (q \leftrightarrow r)$

20. Xác định giá trị của  $x$  sau mỗi khi gặp câu lệnh dưới đây trong một chương trình máy tính, nếu trước khi tới câu lệnh đó  $x = 1$ .

a) **if**  $1 + 2 = 3$  **then**  $x := x + 1$ .

b) **if**  $(1 + 1 = 3)$  OR  $(2 + 2 = 3)$  **then**  $x := x + 1$ .

c) **if**  $(2 + 3 = 5)$  AND  $(3 + 4 = 7)$  **then**  $x := x + 1$ .

d) **if**  $(1 + 1 = 2)$  XOR  $(1 + 2 = 3)$  **then**  $x := x + 1$ .

e) **if**  $x < 2$  **then**  $x := x + 1$ .

**21.** Tìm các OR bit, AND bit và XOR bit của các cặp xâu bit sau :

- a) 10 11110 ; 01 00001
- b) 111 10000 ; 101 01010
- c) 00011 10001 ; 10010 01000
- d) 11111 11111 ; 00000 00000

**22.** Xác định các biểu thức sau :

- a) 11000  $\wedge$  (01011  $\vee$  11011)
- b) (01111  $\wedge$  10101)  $\vee$  01000
- c) (01010  $\oplus$  11011)  $\oplus$  01000
- d) (11011  $\vee$  01010)  $\wedge$  (10001  $\vee$  11011)

**Logic mờ** được sử dụng trong trí tuệ nhân tạo. Trong logic mờ, giá trị chân lý của một mệnh đề là một số nằm giữa 0 và 1. Một mệnh đề với giá trị chân lý 0 là sai, và giá trị chân lý 1 là đúng. Còn giá trị chân lý nằm giữa 0 và 1 chỉ ra mức độ thay đổi của chân lý. Ví dụ, giá trị chân lý 0,8 có thể được gán cho mệnh đề "Fred hạnh phúc" vì phần lớn thời gian Fred sống hạnh phúc, giá trị chân lý 0,4 có thể được gán cho mệnh đề "John hạnh phúc" vì John hạnh phúc gần một nửa thời gian.

23. Giá trị chân lý của phủ định một mệnh đề trong logic mờ là hiệu của 1 và giá trị chân lý của mệnh đề đó. Hãy xác định giá trị chân lý của mệnh đề "Fred không hạnh phúc" và "John không hạnh phúc".
24. Giá trị chân lý của hợp hai mệnh đề trong logic mờ là giá trị chân lý nhỏ nhất của hai mệnh đề đó. Hãy xác định giá trị chân lý của các mệnh đề sau "Fred và John đều hạnh phúc" và "Cả Fred và John đều không hạnh phúc".

**ĐỊNH NGHĨA** 1. Một mệnh đề phức hợp mà luôn luôn đúng bất kể các giá trị chân lý của các mệnh đề thành phần của nó được gọi là *hằng đúng* (tautology). Một mệnh đề mà luôn luôn sai được gọi là *mâu thuẫn*. Cuối cùng, một mệnh đề không phải là hằng đúng, cũng không phải là mâu thuẫn được gọi là *tiếp liên* (contingency).

**Ví dụ** 1. Chúng ta có thể xây dựng các ví dụ về các mệnh đề hằng đúng và mâu thuẫn bằng cách chỉ dùng một mệnh đề. Hãy xét bảng giá trị chân lý của  $p \vee \neg p$  và  $p \wedge \neg p$  cho trong Bảng 1. Vì  $p \vee \neg p$  là luôn luôn đúng vậy nó là hằng đúng. Vì  $p \wedge \neg p$  là luôn luôn sai, nên nó là mâu thuẫn.



**BẢNG 1. Ví dụ về mệnh đề hằng đúng và mệnh đề mâu thuẫn**

$p$	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
T	F	T	F
F	T	T	F

## 2. T ửng logic

- Các m nh ư ph c h p luôn luôn có cùng giá tr chân lý c g i là t ửng logic.
- **nh ngh a** 1. các m nh ư p và q c g i là t ửng logic n u p $\leftrightarrow$ q là h ng úng.
- Kí hi u:  $p \Leftrightarrow q$  ch p và q là t ửng logic.
- M t cách xác nh hai m nh có t ửng hay không là dùng b ằng giá tr chân lý.

**BẢNG 2. Bảng chân lý đối với  $\neg(p \vee q)$  và  $\neg p \wedge \neg q$**

$p$	$q$	$p \vee q$	$\neg(p \vee q)$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$
T	T	T	F	F	F	F
T	F	T	F	F	T	F
F	T	T	F	T	F	F
F	F	F	T	T	T	T

**Ví dụ 2.** Chứng minh rằng  $\neg(p \vee q)$  và  $\neg p \wedge \neg q$  là **tương đương logic**. Sự tương đương này là một trong số các luật De Morgan đối với các mệnh đề, các luật này được gọi theo tên nhà toán học Anh Augustus De Morgan, giữa thế kỷ 19.

**BÀNG 3. Bảng giá trị chân lý đối với  $\neg p \vee q$  và  $p \rightarrow q$**

$p$	$q$	$\neg p$	$\neg p \vee q$	$p \rightarrow q$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	F	T

**Ví dụ 3.** Chứng minh rằng  $p \rightarrow q$  và  $\neg p \vee q$  là tương đương logic.

*Giải :* Chúng ta lập bảng chân lý cho các mệnh đề này trong Bảng 3. Vì các giá trị chân lý của  $\neg p \vee q$  và  $p \rightarrow q$  phù hợp nhau, nên các mệnh đề này là tương đương logic.



**Ví dụ 4.** Chứng minh rằng các mệnh đề  $p \vee (q \wedge r)$  và  $(p \vee q) \wedge (p \vee r)$  là tương đương logic. Đây là luật phân bố của phép tuyển đổi với phép hợp.

**BẢNG 4. Một cách chứng minh  $p \vee (q \wedge r)$  và  $(p \vee q) \wedge (p \vee r)$  là tương đương logic**

$p$	$q$	$r$	$q \wedge r$	$p \vee (q \wedge r)$	$p \vee q$	$p \vee r$	$(p \vee q) \wedge (p \vee r)$
T	T	T	T	T	T	T	T
T	T	F	F	T	T	T	T
T	F	T	F	T	T	T	T
T	F	F	F	T	T	T	T
F	T	T	F	T	T	T	T
F	T	F	F	F	T	F	F
F	F	T	F	F	F	T	F
F	F	F	F	F	F	F	F

BÀNG 5. Các tương đương logic

TƯƠNG ĐƯƠNG	TÊN GỌI
$p \wedge T \Leftrightarrow p$ $p \vee F \Leftrightarrow p$	Luật đồng nhất
$p \vee T \Leftrightarrow T$ $p \wedge F \Leftrightarrow F$	Luật ruột
$p \vee p \Leftrightarrow p$ $P \wedge p \Leftrightarrow p$	Luật Kíy đẳng

$$\neg(\neg p) \Leftrightarrow p$$

Luật phủ định kép

$$p \vee q \Leftrightarrow q \vee p$$

Luật giao hoán

$$p \wedge q \Leftrightarrow q \wedge p$$

$$(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$$

Luật kết hợp

$$(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$$

$$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$$

Luật phân phối

$$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$$

$$\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$$

Luật De Morgan

$$\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$$

### BẢNG 8. Một số tương đương tiện ích

$$p \vee \neg p \Leftrightarrow T$$

$$p \wedge \neg p \Leftrightarrow F$$

$$(p \rightarrow q) \Leftrightarrow (\neg p \vee q)$$

**Ví dụ 5.** Chứng minh rằng  $\neg(p \vee (\neg p \wedge q))$  và  $\neg p \wedge \neg q$  là tương đương logic.

$$\begin{aligned}\neg(p \vee (\neg p \wedge q)) &\Leftrightarrow \neg p \wedge \neg(\neg p \wedge q) && \text{theo Luật De Morgan thứ hai} \\&\Leftrightarrow \neg p \wedge [\neg(\neg p) \vee \neg q] && \text{theo Luật De Morgan thứ nhất} \\&\Leftrightarrow \neg p \wedge [p \vee \neg q] && \text{theo Luật phủ định kép.} \\&\Leftrightarrow (\neg p \wedge p) \vee (\neg p \wedge \neg q) && \text{theo Luật phân phối.} \\&\Leftrightarrow \mathbf{F} \vee (\neg p \wedge \neg q) && \text{Vì } \neg p \wedge p \Leftrightarrow \mathbf{F} \\&\Leftrightarrow (\neg p \wedge \neg q) \vee \mathbf{F} && \text{theo Luật giao hoán đối với phép tuyển} \\&\Leftrightarrow \neg p \wedge \neg q && \text{theo Luật đồng nhất đối với } \mathbf{F}\end{aligned}$$

Vậy  $\neg(p \vee (\neg p \wedge q))$  và  $\neg p \wedge \neg q$  là tương đương logic.

## Ví dụ 6. Chứng minh rằng $(p \wedge q) \rightarrow (p \vee q)$ là hằng đúng.

*Giải :* Để chứng minh một mệnh đề là hằng đúng, ta sẽ dùng các tương đương logic để chứng tỏ rằng nó tương đương logic với T (Chú ý : Điều này cũng có thể làm được bằng cách lập bảng chân lý).

$$\begin{aligned}(p \wedge q) \rightarrow (p \vee q) &\Leftrightarrow \neg(p \wedge q) \vee (p \vee q) && \text{theo Ví dụ 3} \\&\Leftrightarrow (\neg p \vee \neg q) \vee (p \vee q) && \text{theo Luật De Morgan thứ nhất} \\&\Leftrightarrow (\neg p \vee p) \vee (\neg q \vee q) && \text{theo Luật kết hợp và giao hoán} \\&\Leftrightarrow T \vee T && \text{đổi với phép tuyển} \\&\Leftrightarrow T && \text{theo Ví dụ 1 và Luật giao hoán} \\&&& \text{đổi với phép tuyển} \\&&& \text{theo Luật nuốt.}\end{aligned}$$

## Bài tập

1. Dùng bảng chân lý để chứng minh các tương đương sau :

a)  $p \wedge T \Leftrightarrow p$

b)  $p \vee F \Leftrightarrow p$

c)  $p \wedge F \Leftrightarrow F$

d)  $p \vee T \Leftrightarrow T$

e)  $p \vee p \Leftrightarrow p$

f)  $p \wedge p \Leftrightarrow p$

2. Chứng minh rằng  $\neg(\neg p)$  và  $p$  là tương đương logic.

3. Dùng bảng chân lý để chứng minh luật giao hoán :

a)  $p \vee q \Leftrightarrow q \vee p$

b)  $p \wedge q \Leftrightarrow q \wedge p$

4. Dùng bảng chân lý để chứng minh luật kết hợp :

a)  $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$

b)  $(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$

5. Dùng bảng chân lý để chứng minh luật phân phối :

$$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$$

6. Dùng bảng chân lý để chứng minh tương đương :

$$\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$$

7. Dùng bảng chân lý chứng minh các mệnh đề kéo theo sau là hàng đúng :

- a)  $(p \wedge q) \rightarrow p$
- b)  $p \rightarrow (p \vee q)$
- c)  $\neg p \rightarrow (p \rightarrow q)$
- d)  $(p \wedge q) \rightarrow (p \rightarrow q)$
- e)  $\neg(p \rightarrow q) \rightarrow p$
- f)  $\neg(p \rightarrow q) \rightarrow \neg q$

9. Làm lại Bài tập 7 nhưng không dùng các bảng chân lý

12. Xác định xem  $(\neg p \wedge (p \rightarrow q)) \rightarrow \neg q$  có phải là hằng đúng không?
13. Cũng hỏi như trên với mệnh đề  $(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$
14. Chứng minh rằng  $p \leftrightarrow q$  và  $(p \wedge q) \vee (\neg p \wedge \neg q)$  là tương đương.
15. Chứng minh rằng  $(p \rightarrow q) \rightarrow r$  và  $p \rightarrow (q \rightarrow r)$  là không tương đương.
16. Chứng minh rằng  $p \rightarrow q$  và  $\neg q \rightarrow \neg p$  là tương đương.
17. Chứng minh rằng  $\neg p \leftrightarrow q$  và  $p \leftrightarrow \neg q$  là tương đương.
18. Chứng minh rằng  $\neg(p \oplus q)$  và  $p \leftrightarrow q$  là tương đương.
19. Chứng minh rằng  $\neg(p \leftrightarrow q)$  và  $\neg p \leftrightarrow q$  là tương đương.

**Đối ngẫu** của một mệnh đề phức hợp chỉ chứa các toán tử logic  $\vee$ ,  $\wedge$  và  $\neg$  là một mệnh đề nhận được bằng cách thay mỗi  $\vee$  bằng  $\wedge$ , mỗi  $\wedge$  bằng  $\vee$ , mỗi T bằng F và mỗi F bằng T. Đối ngẫu của s được ký hiệu là  $s^*$ .

20. Tlm đối ngẫu của các mệnh đề sau :

- a)  $p \wedge \neg q \wedge \neg r$
- b)  $(p \wedge q \wedge r) \vee s$
- c)  $(p \vee \mathbf{F}) \wedge (q \vee \mathbf{T})$

Các bài tập dưới đây liên quan đến các toán tử logic NAND và NOR. Mệnh đề  $p$  NAND  $q$  là đúng khi  $p$  hoặc  $q$  hoặc cả hai đều sai ; và nó là sai khi cả  $p$  và  $q$  đều đúng. Mệnh đề  $p$  NOR  $q$  là đúng khi cả  $p$  và  $q$  đều sai ; và sai trong các trường hợp còn lại. Các mệnh đề  $p$  NAND  $q$  và  $p$  NOR  $q$  được ký hiệu tương ứng là  $p \mid q$  và  $p \downarrow q$ .

30. Lập bảng giá trị chân lý cho toán tử logic NAND.

31. Chứng minh rằng  $p \mid q$  là tương đương logic với  $\neg(p \wedge q)$ .

32. Lập bảng giá trị chân lý của toán tử logic NOR.

### 3. VĂNG VÀ LÝ TƯ

#### MỞ ĐẦU

Các câu có liên quan đến các biến như :

$$"x > 3"; \quad "x = y + 3" \quad \text{và} \quad "x + y = z"$$

rất thường gặp trong các khẳng định toán học và trong các chương trình máy tính. Các câu này không đúng cũng không sai chừng nào các biến còn chưa được cho những giá trị xác định. Trong tiết này chúng ta sẽ xem xét các cách tạo ra những mệnh đề từ những câu như vậy.

Câu " $x$  lớn hơn 3" có hai bộ phận. Bộ phận thứ nhất, tức là biến  $x$ , là chủ ngữ của câu. Bộ phận thứ hai " $\text{lớn hơn } 3$ " - là **vị ngữ**, nó cho biết một tính chất mà chủ ngữ có thể có. Chúng ta có thể ký hiệu câu " $x$  lớn hơn 3" là  $P(x)$ , với  $P$  ký hiệu vị ngữ " $\text{lớn hơn } 3$ " và  $x$  là biến. Người ta cũng nói câu  $P(x)$  là giá trị của **hàm mệnh đề**  $P$  tại  $x$ . Một khi biến  $x$  được gán cho một giá trị, thì câu  $P(x)$  sẽ có giá trị chân lý. Ta hãy xét các ví dụ sau.

**Ví dụ 1.** Cho  $P(x)$  là ký hiệu của câu " $x > 3$ ". Xác định giá trị chân lý của  $P(4)$  và  $P(2)$ .

*Giải :* Mệnh đề  $P(4)$  nhận được khi thay  $x = 4$  vào câu " $x > 3$ ". Do đó  $P(4)$  – tức là câu " $4 > 3$ " – là đúng. Tuy nhiên  $P(2)$  – tức là câu " $2 > 3$ " – lại là sai.

Chúng ta cũng thường gặp những câu có nhiều biến hơn. Ví dụ, xét câu " $x = y + 3$ ". Chúng ta sẽ ký hiệu câu này là  $Q(x, y)$ , trong đó  $x, y$  là các biến và  $Q$  là vị ngữ. Khi các biến  $x$  và  $y$  được gán cho một giá trị xác định, câu  $Q(x,y)$  sẽ có giá trị chân lý.

**Ví dụ 2.** Cho  $Q(x,y)$  là ký hiệu của câu " $x = y + 3$ ". Xác định giá trị chân lý của các mệnh đề  $Q(1,2)$  và  $Q(3,0)$ .

*Giải :* Để nhận được  $Q(1,2)$  ta đặt  $x = 1$  và  $y = 2$  vào câu  $Q(x,y)$ . Do đó,  $Q(1,2)$  là mệnh đề " $1 = 2 + 3$ ", nó là sai. Câu  $Q(3,0)$  là mệnh đề " $3 = 0 + 3$ ", nó là đúng.

**Ví dụ 3.** Xác định giá trị chân lý của các mệnh đề  $R(1,2,3)$  và  $R(0,0,1)$ .

*Giai :* Mệnh đề  $R(1,2,3)$  nhận được bằng cách đặt  $x = 1$ ,  $y = 2$  và  $z = 3$  vào câu  $R(x,y,z)$ .

Ta thấy rằng  $R(1,2,3)$  chính là mệnh đề " $1 + 2 = 3$ ", nó là đúng. Trong khi đó,  $R(0,0,1)$  là mệnh đề " $0 + 0 = 1$ ", là sai.

**Ví dụ 4.** Xét câu :

```
if x > 0 then x := x + 1
```

Khi gặp câu này trong chương trình, giá trị của biến  $x$  ở điểm đó trong quá trình thực hiện chương trình sẽ được đặt vào  $P(x)$ , tức là đặt vào câu " $x > 0$ ". Nếu  $P(x)$  là đúng đối với giá trị này của  $x$ , thì lệnh gán  $x := x + 1$  sẽ được thực hiện và giá trị của  $x$  sẽ tăng lên 1. Nếu  $P(x)$  là sai đối với giá trị đó của  $x$ , thì lệnh gán sẽ không được thực hiện và giá trị  $x$  không thay đổi.

# L NG T

Khi tất cả các biến trong một hàm mệnh đề đều được gán cho giá trị xác định, thì mệnh đề tạo thành sẽ có giá trị chân lý. Tuy nhiên, còn có một cách quan trọng khác để biến các hàm mệnh đề thành các mệnh đề, mà người ta gọi là **sự lượng hóa**. Ta sẽ xét ở đây hai loại lượng hóa (còn gọi là các lượng từ - ND), đó là lượng từ phổ dụng (cũng quen gọi là lượng từ "với mọi" - ND) và lượng từ tồn tại.

Có nhiều phát biểu toán học khẳng định rằng một tính chất nào đó đúng với mọi giá trị của biến trong một miền đặc biệt nào đó. Miền này được gọi là **không gian** hay **vũ trụ biện luận** (dưới đây ta sẽ gọi tắt là không gian - ND). Một câu như vậy sẽ được diễn đạt bằng lượng từ "với mọi". Lượng từ "với mọi" của một mệnh đề tạo nên một mệnh đề, mệnh đề này là đúng nếu và chỉ nếu  $P(x)$  là đúng với mọi giá trị của  $x$  trong không gian. Không gian sẽ chỉ rõ các giá trị khả dĩ của biến  $x$ .

**ĐỊNH NGHĨA 3.** *Lượng từ "với mọi" của  $P(x)$  là mệnh đề " $P(x)$  đúng với mọi giá trị của  $x$  trong không gian".*

Lượng từ "với mọi" của  $P(x)$  được ký hiệu là :  $\forall x P(x)$

Mệnh đề  $\forall x P(x)$  cũng được diễn đạt như :

"Đối với mọi  $x$   $P(x)$ "

### Ví dụ 5. Diễn đạt câu

"Tất cả sinh viên ở lớp này đều đã học giải tích" như một lượng từ "với mọi".

*Giải :* Cho  $P(x)$  là ký hiệu của câu :

" $x$  đã học giải tích"

Khi đó câu "Tất cả sinh viên ở lớp này đều đã học giải tích" có thể được viết như  $\forall x P(x)$ , ở đây không gian gồm tất cả các sinh viên trong lớp đó.

Câu trên cũng có thể được diễn đạt như sau :

$$\forall x (S(x) \longrightarrow P(x))$$

ở đây  $S(x)$  là câu :

" $x$  ở lớp này".

$P(x)$  vẫn như trước và không bao giờ là tập hợp tất cả sinh viên.



Ví dụ 5 cho thấy thường có nhiều cách để thể hiện một lượng tử.

**Ví dụ 6.** Cho  $P(x)$  là hàm mệnh đề " $x + 1 > x$ ". Xác định giá trị chân lý của lượng tử  $\forall x P(x)$ , ở đây không gian là tập hợp các số thực.

*Giải :* Vì  $P(x)$  đúng với mọi số thực  $x$ , nên lượng tử  $\forall x P(x)$  là đúng.



**Ví dụ 7.** Cho  $Q(x)$  là câu " $x < 2$ ". Xác định giá trị chân lý của lượng tử  $\forall x P(x)$  với không gian là tập hợp các số thực.

*Giải :*  $Q(x)$  là không đúng với mọi số thực  $x$ , vì, ví dụ,  $Q(3)$  là sai. Do đó,  $\forall x Q(x)$  là sai.

**Ví dụ 8.** Xác định giá trị của  $\forall x P(x)$ , với  $P(x)$  là câu " $x^2 < 10$ " và không gian bao gồm các số nguyên dương không vượt quá 4.

*Giải :* Câu  $\forall x P(x)$  giống như là phép hội

$$P(1) \wedge P(2) \wedge P(3) \wedge P(4)$$

vì không gian ở đây gồm các số nguyên 1,2,3 và 4.

Vì  $P(4)$  – tức là mệnh đề " $4^2 < 10$ " – là sai, suy ra  $\forall x P(x)$  là sai.



**ĐỊNH NGHĨA 3.** Lượng từ tồn tại của  $P(x)$  là mệnh đề "Tồn tại một phần tử  $x$  trong không gian sao cho  $P(x)$  là đúng".

Lượng từ tồn tại của  $P(x)$  được ký hiệu là :  $\exists x P(x)$

Lượng từ tồn tại  $\exists x P(x)$  cũng được diễn đạt như sau :

"Tồn tại một  $x$  sao cho  $P(x)$ "

"Tồn tại ít nhất một  $x$  sao cho  $P(x)$ "

hay "Đối với một  $x$  nào đó  $P(x)$ ".

**Ví dụ 9.** Cho  $P(x)$  là câu " $x > 3$ ". Tìm giá trị chân lý của  $\exists x P(x)$  với không gian là tập hợp các số thực.

*Gidi* : Vì " $x > 3$ " là đúng, chẳng hạn với  $x = 4$ , nên lượng từ tồn tại của  $P(x)$ ,  $\exists x P(x)$ , là đúng.



**Ví dụ 10.** Cho  $Q(x)$  là câu " $x = x + 1$ ". Tìm giá trị chân lý của lượng từ  $\exists x P(x)$ , với không gian là tập hợp các số thực.

*Gidi* : Vì  $Q(x)$  là sai đối với mọi số thực  $x$ , nên lượng từ tồn tại của  $Q(x)$ ,  $\exists x Q(x)$ , là sai.

**BÀNG 1. Các lượng từ**

MÊNH ĐỀ	KHI NÀO ĐÚNG ?	KHI NÀO SAI?
$\forall x P(x)$	P(x) đúng với mọi x	Có một giá trị của x để P(x) sai
$\exists x P(x)$	Có một giá trị của x để P(x) đúng	P(x) sai với mọi x

**Ví dụ 11.** Xác định giá trị chân lý của  $\exists x P(x)$ , với P(x) là câu " $x^2 > 10$ " và không gian gồm các số nguyên dương không lớn hơn 4.

*Giai :* Vì không gian là {1, 2, 3, 4}, mệnh đề  $\exists x P(x)$  giống hệt như phép tuyển :

$$P(1) \vee P(2) \vee P(3) \vee P(4)$$

Vì  $P(4)$  – tức là câu " $4^2 > 10$ " – là đúng nên suy ra  $\exists x P(x)$  là đúng. ■

## Dуч các câu thông th ng thành bi u th c Logic

- Trong ph n 1 mô t d ch các câu thông th ng thành các bi u th c logic ch a nhi u m nh và các liên t logic.
- Trong ph n này s bi u di n c t p h p r ng l n h n các câu thông th ng thành các bi u th c logic. M c ích lo i i nh ng i u mù m , ch a rõ ràng và làm cho ta có th dùng các câu ó suy lu n c.
- Các ví d sau cho th y các toán t logic và l ng t dùng di n t các câu thông th ng, t ng t nh lo i câu th ng g p trong các phát bi u toán h c, trong l p trình logic và trí tu nhân t o.

**Ví dụ 12.** Biểu diễn câu "Mọi người đều có chính xác một người bạn tốt nhất" thành một biểu thức logic.

*Giải:* Giả sử  $B(x,y)$  là câu "y là bạn tốt nhất của x". Để dịch câu trong ví dụ, cần chú ý câu  $B(x,y)$  muốn nói rằng đối với mỗi một cá nhân x có một cá nhân khác là y sao cho y là bạn tốt nhất của x, và nếu z là một cá nhân khác y thì z không phải là bạn tốt nhất của x. Do đó, câu trong ví dụ có thể dịch thành :

$$\forall x \exists y \forall z [B(x,y) \wedge ((z \neq y) \rightarrow \neg B(x,z))]$$



# CÁC VÍ DỤ CỦA LEWIS CARROL

- Lewis Carroll (bút danh C.L.Dodgson) tác giả của “Alice trong xứ sở kỳ lân” và là một công trình logic kinh điển.

**Ví dụ 14.** Xét các câu sau. Hai câu đầu được gọi là *tiền đề* và câu thứ ba được gọi là *kết luận*. Toàn bộ tập hợp ba câu này được gọi là một *suy lí*.

"Tất cả sư tử đều hung dữ"

"Một số sư tử không uống cà phê"

"Một số sinh vật hung dữ không uống cà phê".

$P(x)$  :  $x$  là s t

$Q(x)$  :  $x$  hung d

$R(x)$  :  $x$  u ng cafe

Ta có thể biểu diễn các câu đó như sau :

$$\forall x \cdot (P(x) \rightarrow Q(x))$$

$$\exists x \cdot (P(x) \wedge \neg R(x))$$

$$\exists x \cdot (Q(x) \wedge \neg R(x))$$

**Ví dụ 15.** Xét các câu sau, trong đó ba câu đầu là tiên đề và câu thứ tư là kết luận đúng.

"Tất cả chim ruồi đều có màu sắc sờ"

"Không có con chim lớn nào sống bằng mật ong"

"Các chim không sống bằng mật ong đều có màu xám"

"Chim ruồi là nhỏ".

Gọi  $P(x)$ ,  $Q(x)$ ,  $R(x)$  và  $S(x)$  là các câu " $x$  là chim ruồi" ; " $x$  là lớn", " $x$  sống bằng mật ong", và " $x$  có màu sắc sờ", tương ứng. Giả sử rằng không gian là tất cả các loại chim, hãy diễn đạt các câu trong suy lí trên bằng cách dùng  $P(x)$ ,  $Q(x)$ ,  $R(x)$ ,  $S(x)$  và các lượng tử.

*Giai :* Ta có thể biểu diễn các câu trong suy lí trên như sau :

$$\forall x (P(x) \rightarrow S(x))$$

$$\neg \exists x (Q(x) \wedge R(x))$$

$$\forall x (\neg R(x) \rightarrow \neg S(x))$$

$$\forall x (P(x) \rightarrow \neg Q(x))$$

## CÁC BI N RÀNG BU

Khi một lượng từ được dùng đối với biến  $x$  hoặc khi chúng ta gán một giá trị cho biến đó, chúng ta nói rằng sự thâm nhập này của biến là bị ràng buộc. Sự thâm nhập của một biến không bị ràng buộc hoặc không được đặt bằng một giá trị đặc biệt nào đó được gọi là tự do. Tất cả các biến thâm nhập trong các hàm mệnh đề đều phải bị ràng buộc để biến nó thành một mệnh đề. Điều này được làm bằng cách dùng các lượng từ phổ dụng và tồn tại kết hợp với việc gán giá trị.

**Ví dụ 16.** Cho  $P(x,y)$  là câu " $x + y = y + x$ ". Xác định giá trị chân lý của các lượng từ  $\forall x \forall y P(x,y)$ .

*Giải :* Lượng từ

$$\forall x \forall y P(x,y)$$

là ký hiệu của mệnh đề :

"Với mọi số thực  $x$  và với mọi số thực  $y$ ,  
 $x + y = y + x$  là đúng".

Vì  $P(x,y)$  đúng với mọi số thực  $x$  và  $y$ , nên mệnh đề  $\forall x \forall y P(x,y)$  là đúng

**Ví dụ 17.** Cho  $Q(x,y)$  là câu " $x + y = 0$ ". Xác định giá trị chân lý của các lượng từ  $\exists y \forall x Q(x,y)$  và  $\forall x \exists y Q(x,y)$ .

*Giải :* Lượng từ

$$\exists y \forall x Q(x,y)$$

là ký hiệu của mệnh đề :

"Tồn tại một số thực  $y$  sao cho với mọi số thực  $x$ ,  
 $Q(x,y)$  là đúng".

Bất kể số  $y$  được chọn là bao nhiêu, chỉ có một giá trị của  $x$  thoả mãn  $x + y = 0$ . Vì không có một số thực  $y$  sao cho  $x + y = 0$  đúng với mọi số thực  $x$ , nên mệnh đề  $\exists y \forall x Q(x,y)$  là sai.

Lượng từ

$$\forall x \exists y Q(x,y)$$

là ký hiệu của câu

"Với mọi số thực  $x$ , tồn tại một số thực  $y$  sao cho  $Q(x,y)$  là đúng".

Với số thực  $x$  đã cho, luôn có một số thực  $y$  sao cho  $x + y = 0$ , cụ thể là  $y = -x$ . Từ đó suy ra mệnh đề  $\forall x \exists y Q(x,y)$  là đúng.

# Các l ạng t hai bi n

MÊNH ĐỀ	KHI NÀO ĐÚNG?	KHI NÀO SAI ?
$\forall x \forall y P(x,y)$ $\forall y \forall x P(x,y)$	$P(x,y)$ đúng với mọi cặp $(x,y)$	Có một cặp $(x,y)$ đối với nó $P(x,y)$ là sai
$\forall x \exists y P(x,y)$	Với mọi $x$ , có một $y$ sao cho $P(x,y)$ là đúng	Có một $x$ sao cho $P(x,y)$ là sai với mọi $y$
$\exists x \forall y P(x,y)$	Có một $x$ sao cho $P(x,y)$ đúng với mọi $y$	Với mọi $x$ có một $y$ sao cho $P(x,y)$ là sai
$\exists x \exists y P(x,y)$ $\exists y \exists x P(x,y)$	Có một cặp $(x, y)$ sao cho $P(x,y)$ là đúng	$P(x,y)$ là sai đối với mọi cặp $(x,y)$

**Ví dụ 18.** Cho  $Q(x,y,z)$  là câu " $x + y = z$ ". Xác định giá trị chân lý của  $\forall x \forall y \exists z Q(x,y,z)$  và  $\exists z \forall x \forall y Q(x,y,z)$ .

*Giai :* Giả sử  $x,y$  đã được gán giá trị. Khi đó tồn tại một giá trị  $z$  sao cho  $x + y = z$ . Vì vậy, lượng từ :

$$\forall x \forall y \exists z Q(x,y,z)$$

là ký hiệu của câu :

"Đối với mọi số thực  $x$  và mọi số thực  $y$ , tồn tại một số thực  $z$  sao cho  $x + y = z$ " là đúng. Thứ tự của các lượng từ ở đây là quan trọng, vì lượng từ :

$$\exists z \forall x \forall y Q(x,y,z)$$

là ký hiệu của câu :

"Có một số thực  $z$  sao cho với mọi số thực  $x$  và mọi số thực  $y$  đẳng thức  $x + y = z$  là đúng" lại là một mệnh đề sai, vì không có một giá trị nào của  $z$  lại thoả mãn phương trình  $x + y = z$  với mọi giá trị của  $x$  và  $y$ .

**Ví dụ 20.** Diễn đạt định nghĩa giới hạn bằng cách dùng các lượng từ

$$\lim_{x \rightarrow a} f(x) = L$$

là : "Với mọi số thực  $\varepsilon > 0$  tồn tại một số thực  $\delta > 0$  sao cho  $|f(x) - L| < \varepsilon$  khi  $0 < |x - a| < \delta$ ".

Định nghĩa này của giới hạn có thể được diễn đạt bằng cách dùng các lượng từ như sau :

$$\forall \varepsilon \exists \delta \forall x (0 < |x - a| < \delta \rightarrow |f(x) - L| < \varepsilon)$$

ở đây không gian đối với các biến  $\delta$  và  $\varepsilon$  là tập các số thực dương, còn đối với  $x$  là tập các số thực.

**BẢNG 3. Phù định các lượng tử**

PHÙ ĐỊNH	MỆNH ĐỀ TƯƠNG ĐƯƠNG	KHI NÀO PHÙ ĐỊNH LÀ ĐÚNG?	KHI NÀO SAI?
$\neg \exists x P(x)$	$\forall x \neg P(x)$	$P(x)$ sai với mọi $x$	Có một $x$ để $P(x)$ là đúng.
$\neg \forall x P(x)$	$\exists x \neg P(x)$	Có một $x$ để $P(x)$ là sai	$P(x)$ đúng với mọi $x$

## BÀI TẬP

1. Cho  $P(x)$  là câu " $x \leq 4$ ". Xác định giá trị chân lý của các mệnh đề sau:
  - a)  $P(0)$  ;
  - b)  $P(4)$  ;
  - c)  $P(6)$
2. Cho  $P(x)$  là câu "từ  $x$  chứa chữ cái  $a$ ". Xác định giá trị chân lý của các mệnh đề sau :
  - a)  $P(\text{orange})$
  - b)  $P(\text{lemon})$
  - c)  $P(\text{true})$
  - d)  $P(\text{false})$
3. Cho  $Q(x,y)$  là câu " $x$  là thủ phủ của  $y$ ". Xác định giá trị chân lý của các mệnh đề sau :
  - a)  $Q(\text{Denver}, \text{Colorado})$
  - b)  $Q(\text{Detroit}, \text{Michigan})$
  - c)  $Q(\text{Massachusetts}, \text{Boston})$
  - d)  $Q(\text{New York}, \text{New York})$

4. Cho biết giá trị của  $x$  sau khi lệnh `if P(x) then x := 1` được thực hiện, biết rằng  $P(x)$  là câu " $x > 1$ " và giá trị của  $x$  khi tới câu lệnh này là :

a)  $x = 0$

b)  $x = 1$

c)  $x = 2$

5. Cho  $P(x)$  là câu " $x$  học ở lớp hơn 5 giờ mỗi ngày trong tuần", ở đây không gian là tập hợp các sinh viên. Hãy diễn đạt các lượng từ sau thành câu thông thường :

a)  $\exists x \ P(x)$

b)  $\forall x \ P(x)$

c)  $\exists x \ \neg P(x)$

d)  $\forall x \ \neg P(x)$

6. Cho  $P(x,y)$  là câu " $x$  đã học môn  $y$ ", với không gian của  $x$  là tập hợp tất cả sinh viên trong lớp bạn, và không gian của  $y$  là tập hợp tất cả các môn tin học ở trường bạn. Hãy diễn đạt các lượng từ sau thành câu thông thường :

a)  $\exists x \ \exists y \ P(x,y),$

b)  $\exists x \ \forall y \ P(x,y)$

c)  $\forall x \ \exists y \ P(x,y),$

d)  $\exists y \ \forall x \ P(x,y)$

e)  $\forall y \ \exists x \ P(x,y),$

f)  $\forall x \ \forall y \ P(x,y).$

7. Cho  $P(x)$  là câu "x nói được tiếng Nga" và  $Q(x)$  là câu "x biết ngôn ngữ C++". Hãy diễn đạt các câu sau bằng cách dùng  $P(x)$ ,  $Q(x)$ , các lượng từ và các liên từ logic. Cho không gian đối với các lượng từ là tập hợp tất cả sinh viên ở trường bạn.
- Có một sinh viên ở trường bạn nói được tiếng Nga và biết C++.
  - Có một sinh viên ở trường bạn nói được tiếng Nga nhưng không biết C++.
  - Mọi sinh viên ở trường bạn đều nói được tiếng Nga hoặc biết C++.
16. Dùng các lượng từ diễn đạt phủ định của các mệnh đề sau, rồi dịch các phủ định đó ra các câu thông thường.
- Mọi sinh viên ở lớp này đều thích môn toán.
  - Có một sinh viên trong lớp này chưa bao giờ nhìn thấy một chiếc máy tính.
  - Có một sinh viên ở lớp này đã học tất cả các môn toán được dạy ở trường này.
  - Có một sinh viên ở lớp này đã ở ít nhất một phòng trong tất cả các tòa nhà ở ký túc xá.

Các Bài tập 17 - 20 dựa trên các câu hỏi lấy từ cuốn sách Logic ký hiệu của Lewis Carroll.

17. Cho  $P(x)$ ,  $Q(x)$  và  $R(x)$  là các câu "x là giáo sư", "x là kẻ ngu dốt" và "x là kẻ vô tích sự", tương ứng. Bằng cách dùng các lượng từ, các liên từ logic cùng với  $P(x)$ ,  $Q(x)$  và  $R(x)$  diễn đạt các câu sau với không gian là tập hợp toàn thể loài người.
- Không có giáo sư nào là kẻ ngu dốt.
  - Mọi kẻ ngu dốt đều là vô tích sự.
  - Không có giáo sư nào là vô tích sự.
21. Chứng tỏ rằng các câu  $\neg \exists x \forall y P(x,y)$  và  $\forall x \exists y \neg P(x,y)$  có cùng giá trị chân lý.
22. Chứng tỏ rằng  $\forall x (P(x) \wedge Q(x))$  và  $\forall x P(x) \wedge \forall x Q(x)$  có cùng giá trị chân lý.
23. Chứng tỏ rằng  $\exists x (P(x) \vee Q(x))$  và  $\exists x P(x) \vee \exists x Q(x)$  có cùng giá trị chân lý.

19. Cho  $P(x)$ ,  $Q(x)$ ,  $R(x)$  và  $S(x)$  tương ứng là các câu "x là một đứa bé", "x là logic", "x có khả năng cai quản một con cá sấu" và "x bị coi thường". Giả sử rằng không gian là tập hợp tất cả mọi người. Hãy dùng các lượng từ, các liên từ logic cùng với  $P(x)$ ,  $Q(x)$ ,  $R(x)$  và  $S(x)$  để diễn đạt các câu sau :
- Những đứa bé là không logic.
  - Không ai bị coi thường nếu cai quản được cá sấu.
  - Những người không logic bị coi thường.
  - Những đứa bé không cai quản được cá sấu.
20. Cho  $P(x)$ ,  $Q(x)$ ,  $R(x)$  và  $S(x)$  tương ứng là các câu sau : "x là một con vịt", "x là một trong số gia cầm của tôi", "x là một viên sĩ quan" và "x săn lùng khiêu vũ". Dùng các lượng từ, các liên từ logic cùng với  $P(x)$ ,  $Q(x)$ ,  $R(x)$  và  $S(x)$  để diễn đạt các câu sau :
- Không có con vịt nào săn lùng khiêu vũ cả.
  - Không có viên sĩ quan nào từ chối khiêu vũ.
  - Toàn bộ đàn gia cầm của tôi đều là vịt.
  - Đàn gia cầm của tôi không phải là các sĩ quan.

# BÀI TẬP – TH

1. G là m tinh thô, vô hổng có số nh  $N > 3$ .  
Chứng minh  $G$  có chia 2 số cùng bội.
2. Số  $G$  có ứng 2 số bội 1. Chứng minh tồn tại mảng dây chuyền nối hai số 0 và i nhau.
3. Xét số  $G$  số, vô hổng gồm  $N$  số,  $M$  số nh và  $P$  thành phần liên thông.
  - a. Chứng minh:  $M \leq (N-P)(N-P+1)/2$ ,  
suy ra nếu  $M > (N-1)(N-2)/2$  thì  $G$  liên thông.
  - a. Số tinh số có 10 số, 37 số nh thì có chia số liên thông hay không?

# BÀI TẬP

4. 

th  $G$  có  $n$ , vô hướng  $g \leq m \leq N$  và  $d(x) \geq (N-1)/2$  với mọi  $x$ . Chứng minh  $G$  liên thông.
5. 

th vô hướng  $G$  liên thông  $g \leq m \leq N$  có  $n$  nhánh. Chứng minh số cạnh của  $G \geq N-1$ .
6. Xét th  $G$  vô hướng. Giả sử  $x$  là nhánh có bắc nhánh tách  $G$ . Giả sử  $d(x) \geq k \geq 2$  với  $k$  nguyên dương. Chứng minh  $G$  chia thành chu trình số có chí số dài lẻ  $n$  hay bằng  $k+1$ .

# BÀI TẬP

7. Cho  $G$  là một đồ thị vô hướng liên thông. Giả sử  $C_1$  và  $C_2$  là 2 dây chuyền simple trong  $G$  có số cung nhau nhau nhất. Chứng minh  $C_1$  và  $C_2$  có cung chung.
8.  $G$  là một đồ thị vô hướng không khuyên và  $d(x) \geq 3$  với mỗi  $x$ . Chứng minh  $G$  có chu trình viễn cảnh ngắn.

# TREE

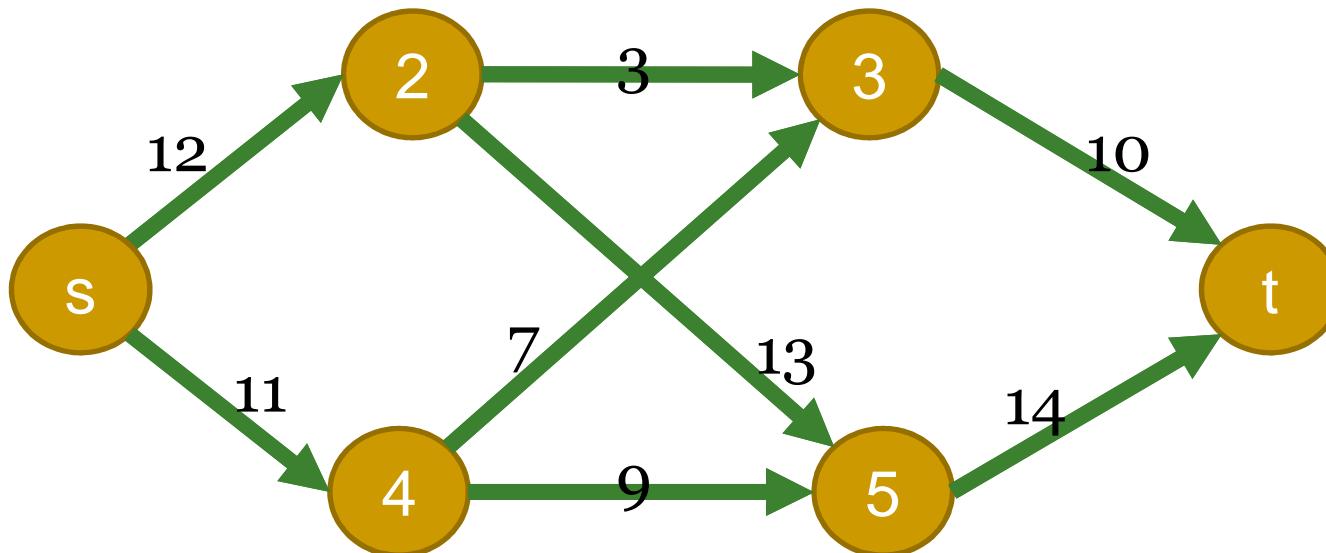
1. Chứng minh các性质 của cây
2. Xác định số lượng cây tối đa có thể dùng CÂY, CHƯƠNG TRÌNH SẮP, ...
3. Chứng minh tính đúngしさ của các giải thuật PRIM, KRUSKAL

# BÀI TẬP – NGHỀ

1. Chứng minh nguyên lý Bellman
2. Chứng minh tính đúng của các thuật toán Dijkstra, Floyd, Bellman
3. Cài đặt thuật toán xác định chu trình Euler
4. Xác định các “nét” cần thiết K nét.

# Bài tập – luồng trong mạng

1. Tìm luồng cung cấp cho mạng sau:



- 
2. Hãy nêu giới phát giới quy tắc và liên thông các nh.
  3. \*Hãy nêu giới pháp tìm cách path cover các tiêu.
  4. Chứng minh rằng mỗi đường có cách đi trên một đồ thị  $G = (V, E)$  luôn có thể xác định sau mỗi dây tia  $|E|$  quá trình tìm đường tăng luồng.
  5. Chứng minh với mỗi cặp nhau, về bờ kia, ta luôn có  $c_f(u, v) + c_f(v, u) = c(u, v) + c(v, u)$ . Vì  $c_f$  là trọng số của cung trên đường tăng luồng.
-