LÝ THUYẾT ĐỒ THỊ

THÔNG TIN VỀ GIÁO VIÊN

TT	Họ tên giáo viên	Học hàm	Học vị	Đơn vị công tác (Bộ môn)
1	Ngô Hữu Phúc	GVC	Tiến sỹ	Bộ môn Khoa học máy tính
2	Vi Bảo Ngọc	TG	Thạc sỹ	Bộ môn Khoa học máy tính

- Thời gian, địa điểm làm việc: Bộ môn Khoa học máy tính - Khoa Công nghệ thông tin - Học viện Kỹ thuật Quân sự.
- Địa chỉ liên hệ: Bộ môn Khoa học máy tính - Khoa Công nghệ thông tin - Học viện Kỹ thuật Quân sự.
- Điện thoại, email: ngohuuphuc76@gmail.com
- Các hướng nghiên cứu chính: Xử lý ảnh, Trí tuệ nhân tạo, Nhận dạng mẫu, Tính toán mềm, Xử lý tiếng nói.

THÔNG TIN CHUNG VỀ MÔN HỌC

- Tên học phần: Lý thuyết đồ thị
- Mã học phần:
- Số tín chỉ: 3
- Học phần (bắt buộc hay lựa chọn): tự chọn
- Các học phần tiên quyết: Đại số tuyến tính, Giải tích đại cương, Tin học cơ bản
- Các yêu cầu đối với học phần (nếu có):
- Giờ tín chỉ đối với các hoạt động:
  - Nghe giảng lý thuyết: 30 tiết
  - Làm bài tập trên lớp: 15 tiết
  - Thảo luận: 6 tiết
  - Thực hành, thực tập (ở PTN, nhà máy, thực tập...): 9 tiết
  - Hoạt động theo nhóm:
  - Tự học: 90 tiết
- Khoa/Bộ môn phụ trách học phần, địa chỉ: Bộ môn Khoa học máy tính - Khoa Công nghệ thông tin - Học viện Kỹ thuật Quân sự.

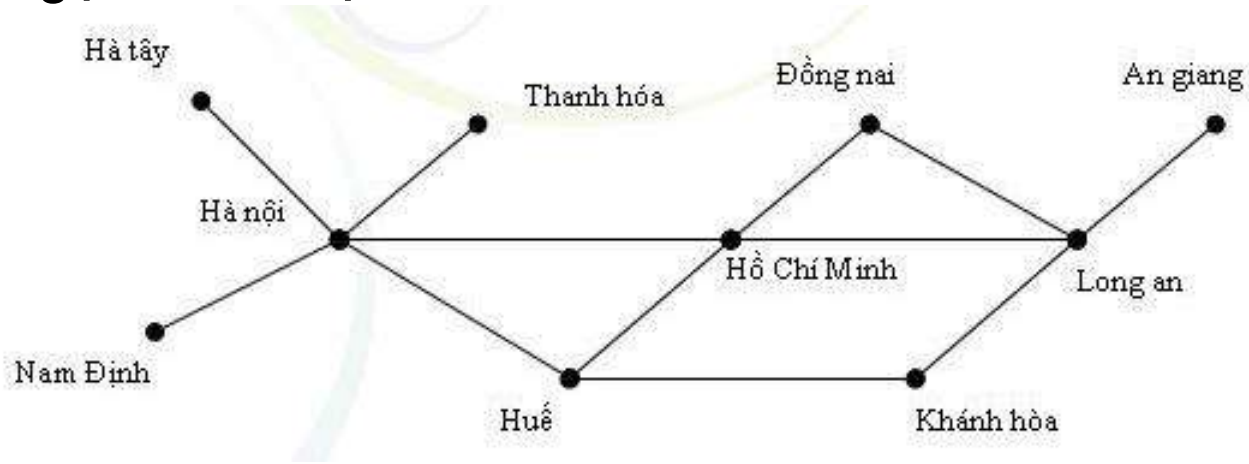
CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 1 KHÁI NIỆM ĐỒ THỊ

• Đồ thị là một cấu trúc rời rạc bao gồm các đỉnh và các cạnh nối các đỉnh này.

• Phân biệt các loại đồ thị khác nhau bởi kiểu và số lượng cạnh nối hai đỉnh nào đó của đồ thị.

Định nghĩa 1 (Đơn đồ thị).

❑ Đơn đồ thị vô hướng G = (V,E) bao gồm V là tập các đỉnh khác rỗng, và E là tập các cặp không có thứ tự gồm hai phần tử khác nhau của V gọi là các cạnh.



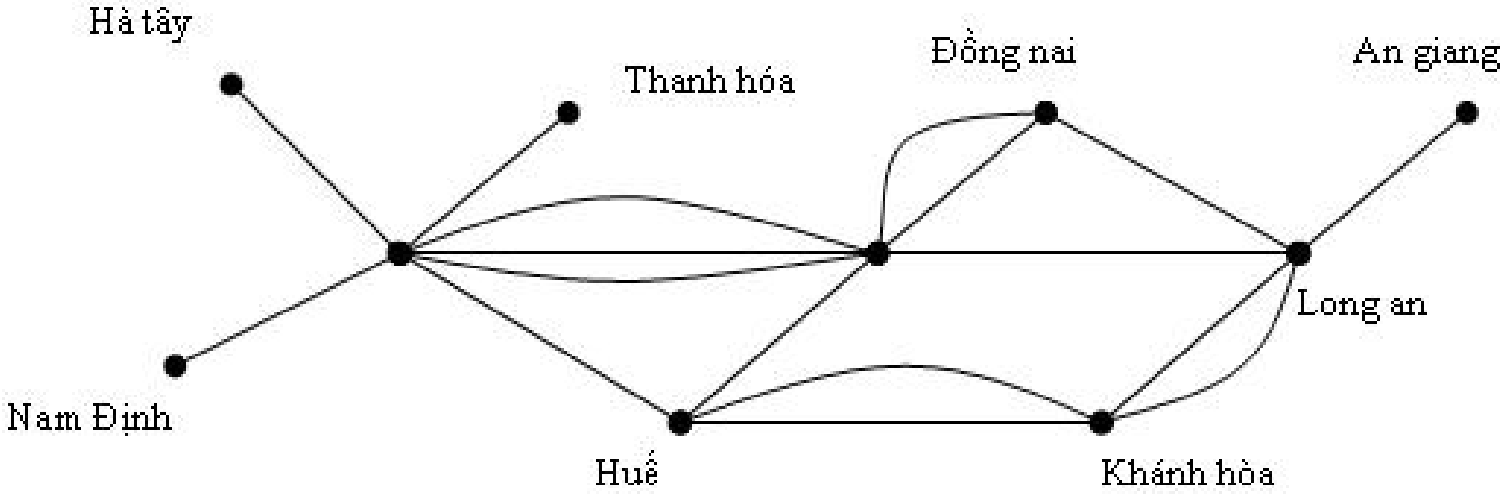
Hình 1. Sơ đồ mạng máy tính đơn kênh thoại.

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 1 KHÁI NIỆM ĐỒ THỊ

Định nghĩa 2 (Đa đồ thị).

❑ Đa đồ thị vô hướng G= (V, E) bao gồm V là tập các đỉnh khác rỗng, và E là tập các cặp không có thứ tự gồm hai phần tử khác nhau của V gọi là các cạnh. Hai cạnh e1 và e2 được gọi là cạnh lặp (bội hay song song) nếu chúng cùng tương ứng với một cặp đỉnh.

❑ Mỗi đơn đồ thị là đa đồ thị, nhưng không phải đa đồ thị nào cũng là đơn đồ thị, vì trong đa đồ thị có thể có hai (hoặc nhiều hơn) cạnh nối một cặp đỉnh nào đó.



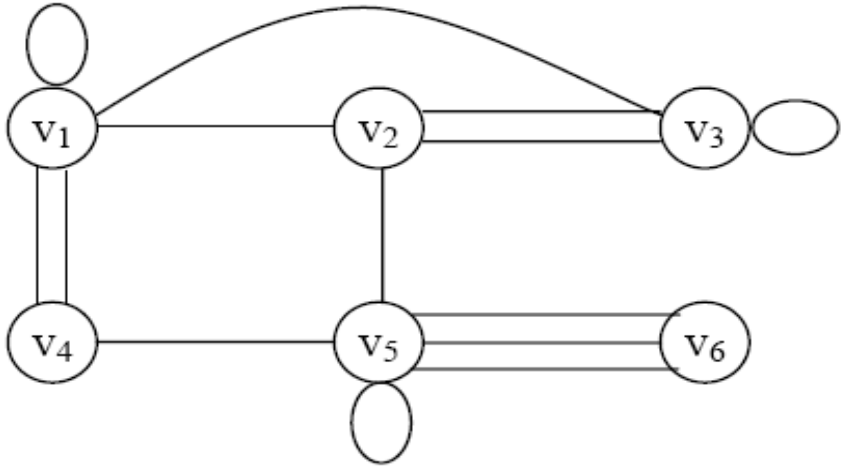
Hình 2. Sơ đồ mạng máy tính đa kênh thoại.

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 1 KHÁI NIỆM ĐỒ THỊ

Định nghĩa 3 (Giả đồ thị).

❑ Giả đồ thị vô hướng G = (V, E) bao gồm V là tập các đỉnh khác rỗng và E là tập các cặp không có thứ tự gồm hai phần tử (không nhất thiết phải khác nhau) của V gọi là cạnh.

Với v Є V, nếu (v,v) Є E thì ta nói có một khuyên tại đỉnh v.

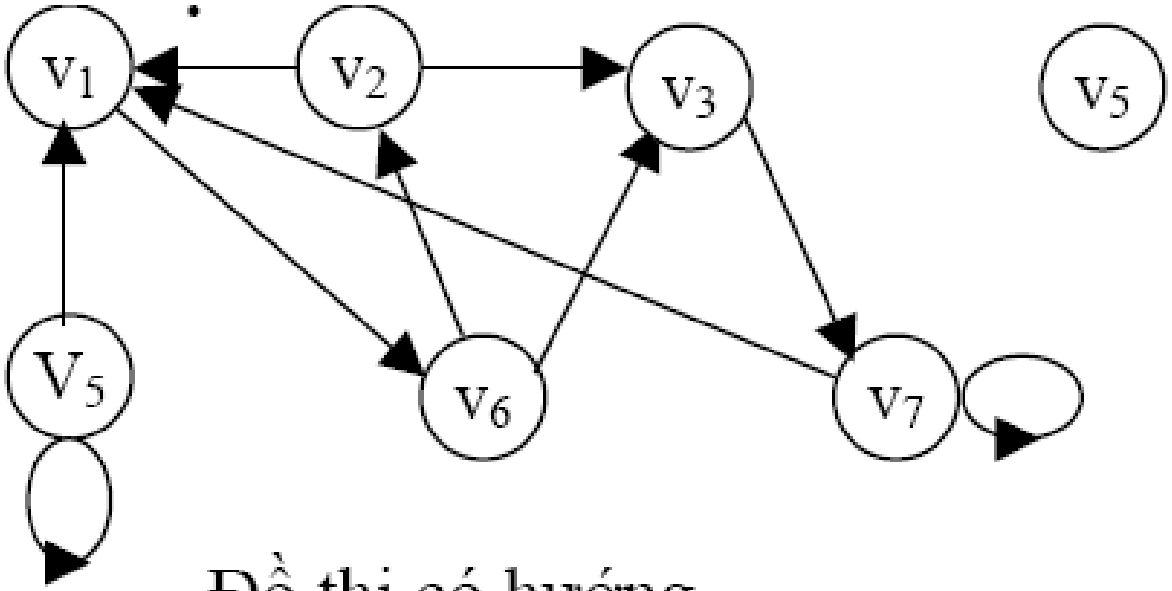


Nhận xét: giả đồ thị là loại đồ thị vô hướng tổng quát nhất vì nó có thể chứa các khuyên và các cạnh lặp. Đa đồ thị là loại đồ thị vô hướng có thể chứa cạnh bội nhưng không thể có các khuyên, còn đơn đồ thị là loại đồ thị vô hướng không chứa cạnh bội hoặc các khuyên.

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 1 KHÁI NIỆM ĐỒ THỊ

<u>Định nghĩa 4</u> (Đơn đồ thị có hướng).

❑ Đơn đồ thị có hướng G = (V, E) bao gồm V là tập các đỉnh khác rỗng và E là tập các cặp có thứ tự gồm hai phần tử khác nhau của V gọi là các cung.

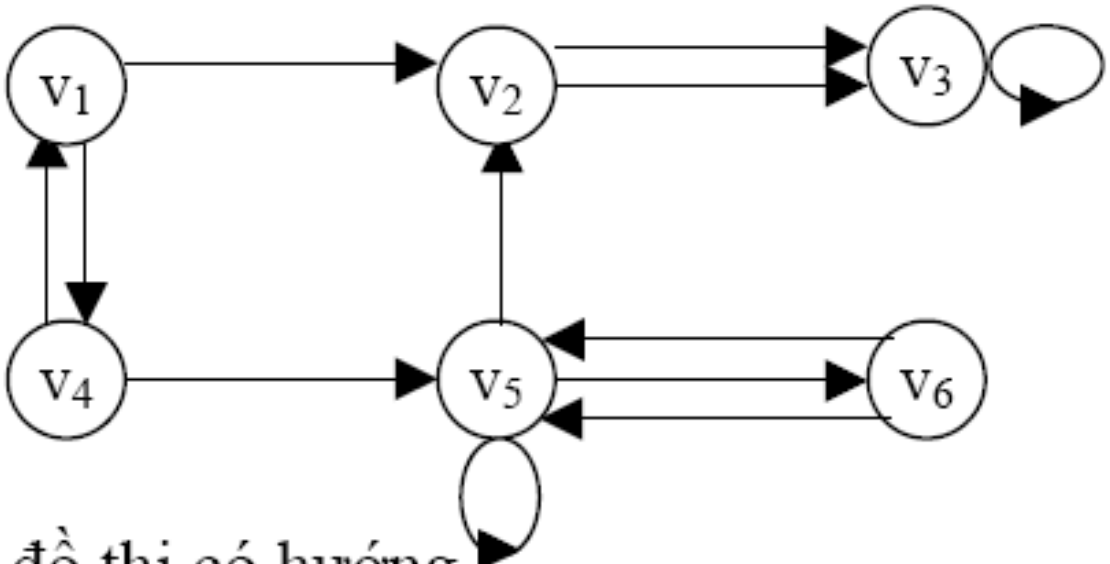


Đồ thị có hướng

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 1 KHÁI NIỆM ĐỒ THỊ

<u>Định nghĩa 5</u> (Đa đồ thị có hướng).

❑ Đa đồ thị có hướng G = (V, E) bao gồm V là tập các đỉnh khác rỗng và E là tập các cặp có thứ tự gồm hai phần tử khác nhau của V gọi là các cung. Hai cung e1, e2 tương ứng với cùng một cặp đỉnh được gọi là cung lặp.



Đa đồ thị có hướng

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 2 BẬC CỦA ĐỈNH

Định nghĩa 1:

❑ Hai đỉnh u và v trong đồ thị (vô hướng) G=(V,E) được gọi là liền kề nếu (u,v) Є E. Nếu e = (u,v) thì e gọi là cạnh liên thuộc với các đỉnh u và v. Cạnh e cũng được gọi là cạnh nối các đỉnh u và v. Các đỉnh u và v gọi là các điểm đầu mút của cạnh e.

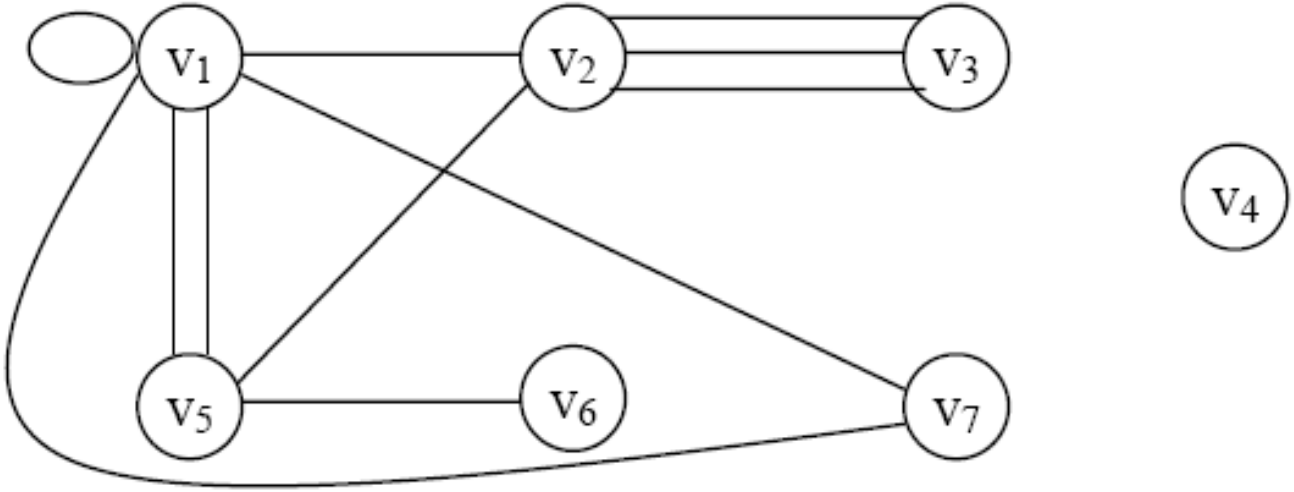
Định nghĩa 2:

❑ Bậc của đỉnh v trong đồ thị G=(V,E), ký hiệu deg(v), là số các cạnh liên thuộc với nó, riêng khuyên tại một đỉnh được tính hai lần cho bậc của nó.

❑ Đỉnh v gọi là đỉnh treo nếu deg(v)=1 và gọi là đỉnh cô lập nếu deg(v)=0.

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 2 BẬC CỦA ĐỈNH

Xét ví dụ:



Ta có: $\deg(v_1)=7$, $\deg(v_2)=5$, $\deg(v_3)=3$, $\deg(v_4)=0$, $\deg(v_5)=4$, $\deg(v_6)=1$, $\deg(v_7)=2$.

Đỉnh v_4 là đỉnh cô lập và đỉnh v_6 là đỉnh treo.

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 2 BẬC CỦA ĐỈNH

Định lý 1. Giả sử G = (V, E) là đồ thị vô hướng với m cạnh. Khi đó tổng bậc của tất cả các đỉnh bằng hai lần số cạnh.

Chứng minh. Rõ ràng mỗi cạnh e = (u, v) được tính một lần trong deg(u) và một lần trong deg(v). Từ đó suy ra tổng tất cả các bậc của các đỉnh bằng hai lần số cạnh.

Hệ quả. Trong đồ thị vô hướng, số đỉnh bậc lẻ (nghĩa là có bậc là số lẻ) là một số chẵn.

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 2 BẬC CỦA ĐỈNH

Chứng minh. Thực vậy, gọi O và U tương ứng là tập đỉnh bậc lẻ và tập đỉnh bậc chẵn của đồ thị.
Ta có:

$$2m = \sum_{v \in U} \deg(v) + \sum_{v \in O} \deg(v)$$

Do deg(v) là chẵn với v là đỉnh trong U nên tổng thứ nhất ở trên là số chẵn.
===> tổng thứ hai (chính là tổng bậc của các đỉnh bậc lẻ) cũng phải là số chẵn,
Do tất cả các số hạng của nó là số lẻ, nên tổng này phải gồm một số chẵn các số hạng. Vì vậy, số đỉnh bậc lẻ phải là số chẵn.

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 2 BẬC CỦA ĐỈNH

Định nghĩa 3.

❑ Nếu e = (u, v) là cung của đồ thị có hướng G thì ta nói hai đỉnh u và v là kề nhau, và nói cung (u, v) nối đỉnh u với đỉnh v hoặc cũng nói cung này là đi ra khỏi đỉnh u và vào đỉnh v. Đỉnh u(v) sẽ được gị là đỉnh đầu (cuối) của cung (u,v).

Định nghĩa 4.

❑ Ta gọi bán bậc ra (bán bậc vào) của đỉnh v trong đồ thị có hướng là số cung của đồ thị đi ra khỏi nó (đi vào nó) và ký hiệu là $deg^+(v)$ ($deg^-(v)$)

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 2 BẬC CỦA ĐỈNH

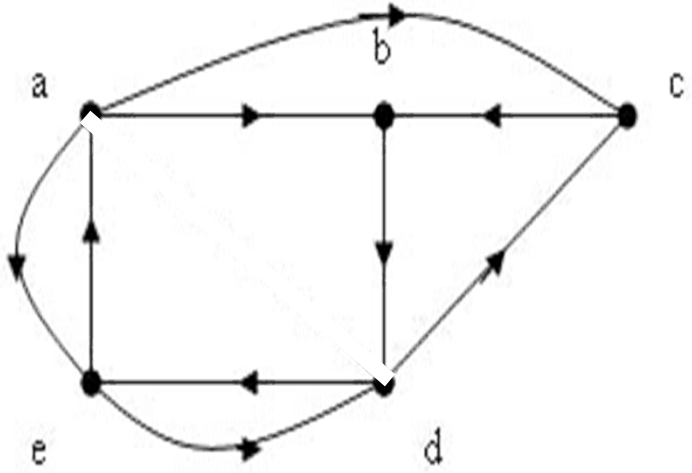
Định lý 2.

Cho G =(V, E) là một đồ thị có hướng. Khi đó:

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

Chứng minh: Kết quả có ngay là vì mỗi cung được tính một lần cho đỉnh đầu và một lần cho đỉnh cuối.

Xét ví du:



Ta có:

$\deg^-(a)=1$, $\deg^-(b)=2$, $\deg^-(c)=2$,
$\deg^-(d)=2$, $\deg^-(e) = 2$.

$\deg^+(a)=3$, $\deg^+(b)=1$, $\deg^+(c)=1$,
$\deg^+(d)=2$, $\deg^+(e)=2$.

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 3 ĐƯỜNG ĐI. CHU TRÌNH. ĐỒ THỊ LIÊN THÔNG

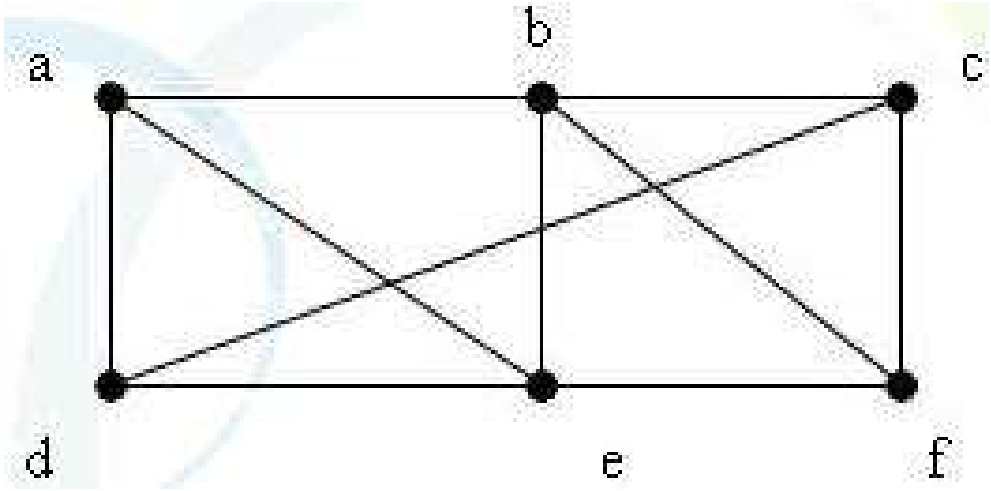
Định nghĩa 1 (Đường đi).

Đường đi độ dài n từ đỉnh u đến đỉnh v, trong đó n là số nguyên dương, trên đồ thị vô hướng G = (V, E) là dãy $x_0, x_1,\ldots, x_{n-1}, x_n$; trong đó $u = x_0$, $v = x_n$, $(x_i , x_{i+1}) \in E$, i = 0, 1, 2,…, n-1.
Đường đi nói trên còn có thể biểu diễn dưới dạng dãy các cạnh: $(x_0, x_1), (x_1, x_2), \ldots, (x_{n-1}, x_n)$

- Đỉnh u gọi là đỉnh đầu, còn đỉnh v gọi là đỉnh cuối của đường đi.
- Đường đi có đỉnh đầu trùng với đỉnh cuối (tức là u = v) được gọi là chu trình.
- Đường đi hay chu trình được gọi là đơn nếu như không có cạnh nào bị lặp lại.

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 3 ĐƯỜNG ĐI. CHU TRÌNH. ĐỒ THỊ LIÊN THÔNG

Ví dụ 1.



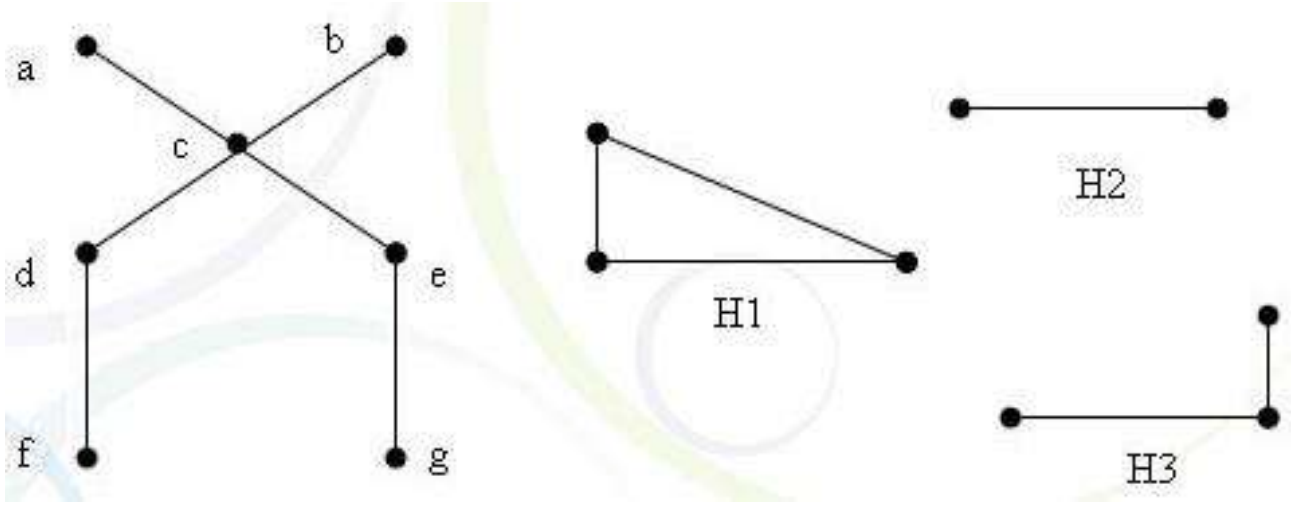
Trên đồ thị vô hướng cho trong hình:

- a, d, c, f, e là đường đi đơn độ dài 4.
- d, e, c, a không là đường đi, do (c,e) không phải là cạnh của đồ thị.
- Dãy b, c, f, e, b là chu trình độ dài 4.
- Đường đi a, b, e, d, a, b có độ dài là 5 không phải là đường đi đơn, do cạnh (a, b) có mặt trong nó 2 lần.

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 3 ĐƯỜNG ĐI. CHU TRÌNH. ĐỒ THỊ LIÊN THÔNG

Định nghĩa 2. (Liên thông)

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



Hình 2. Đồ thị G và H

Đồ thị G là liên thông, còn đồ thị H là không liên thông.

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 3 ĐƯỜNG ĐI. CHU TRÌNH. ĐỒ THỊ LIÊN THÔNG

Định nghĩa 3.

Ta gọi đồ thị con của đồ thị G = (V, E) là đồ thị H = (W, F), trong đó $W \subseteq V$ và $F \subseteq E$.

Trong trường hợp đồ thị là không liên thông, nó sẽ rã ra thành một số đồ thị con liên thông không có đỉnh chung. Những đồ thị con liên thông như vậy ta sẽ gọi là các thành phần liên thông của đồ thị.

Ví dụ 2. Đồ thị H trong hình 2 gồm 3 thành phần liên thông H1, H2, H3.

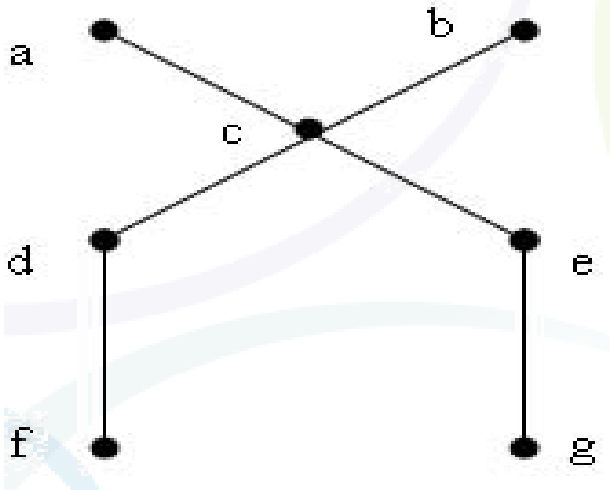
CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 3 ĐƯỜNG ĐI. CHU TRÌNH. ĐỒ THỊ LIÊN THÔNG

Định nghĩa 4.

❑ Đỉnh v được gọi là đỉnh rẽ nhánh nếu việc loại bỏ v cùng với các cạnh liên thuộc với nó khỏi đồ thị làm tăng số thành phần liên thông của đồ thị.

❑ Cạnh e được gọi là cầu nếu việc loại bỏ nó khỏi đồ thị làm tăng số thành phần liên thông của đồ thị.

Ví dụ 3. Trong đồ thị G ở hình 2, đỉnh c, d và e là đỉnh rẽ nhánh, còn các cạnh (c,d) và (c,e) là cầu.



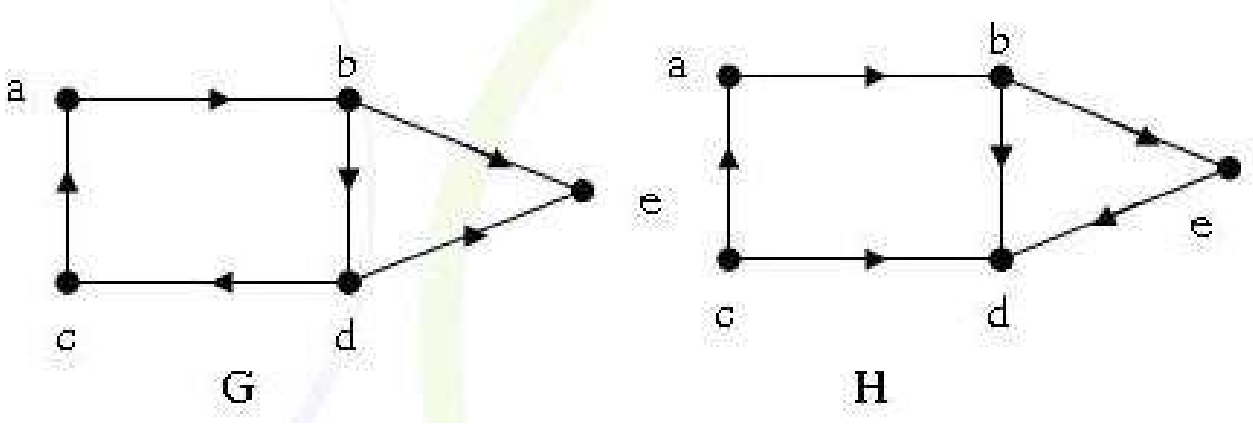
CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 3 ĐƯỜNG ĐI. CHU TRÌNH. ĐỒ THỊ LIÊN THÔNG

Định nghĩa 5.

❑ Đồ thị có hướng G = (V, A) được gọi là liên thông mạnh nếu luôn tìm được đường đi giữa hai đỉnh bất kỳ của nó.

Định nghĩa 6.

❑ Đồ thị có hướng G = (V, A) được gọi là liên thông yếu nếu đồ thị vô hướng tương ứng với nó là vô hướng liên thông.



Hình 3. Đồ thị liên thông mạnh G và đồ thị liên thông yếu H

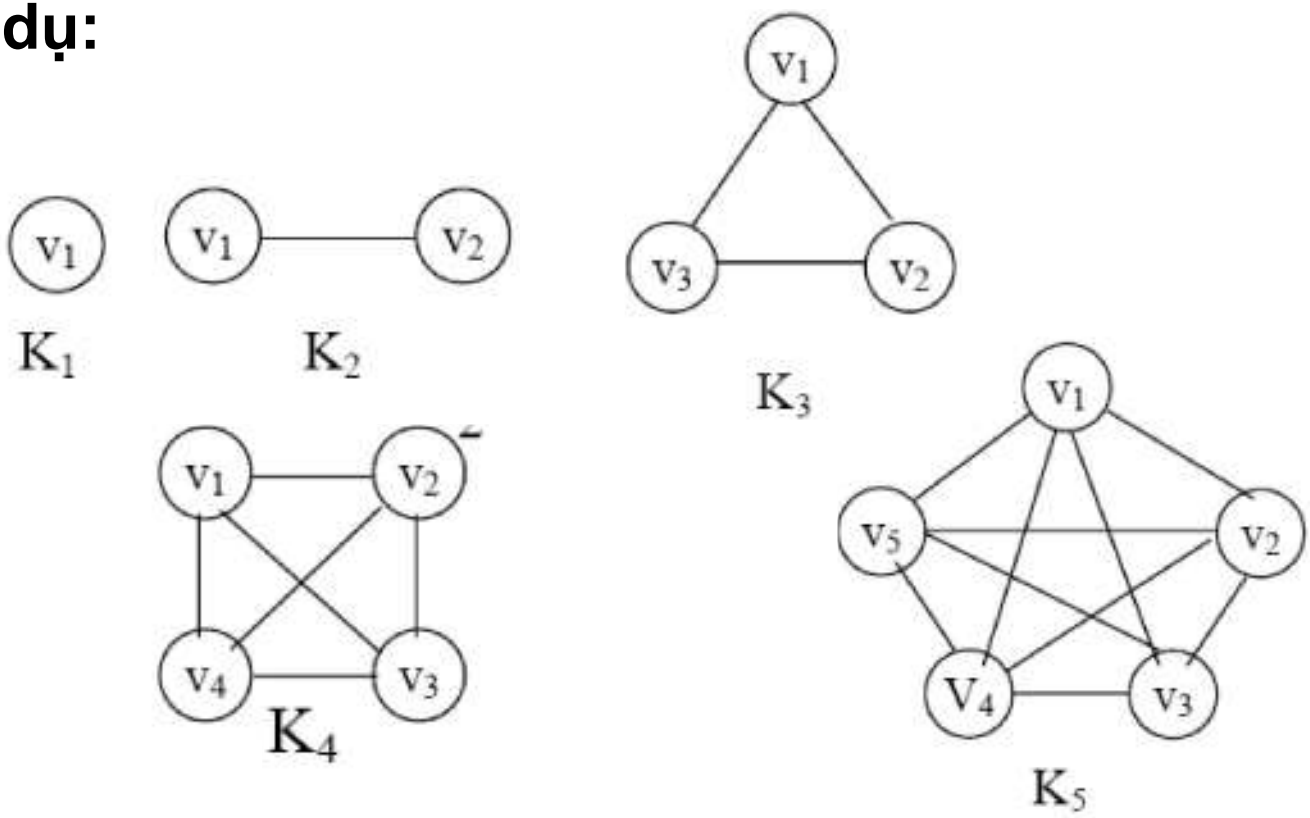
CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 4 ĐƠN ĐỒ THỊ ĐẶC BIỆT

Đồ thị đầy đủ: Đồ thị đầy đủ n đỉnh, ký hiệu là K_n, là đơn đồ thị mà hai đỉnh phân biệt bất kỳ của nó luôn liền kề.

Như vậy, K_n có $n(n-1)/2$ cạnh và

mỗi đỉnh của K_n có bậc là n−1.

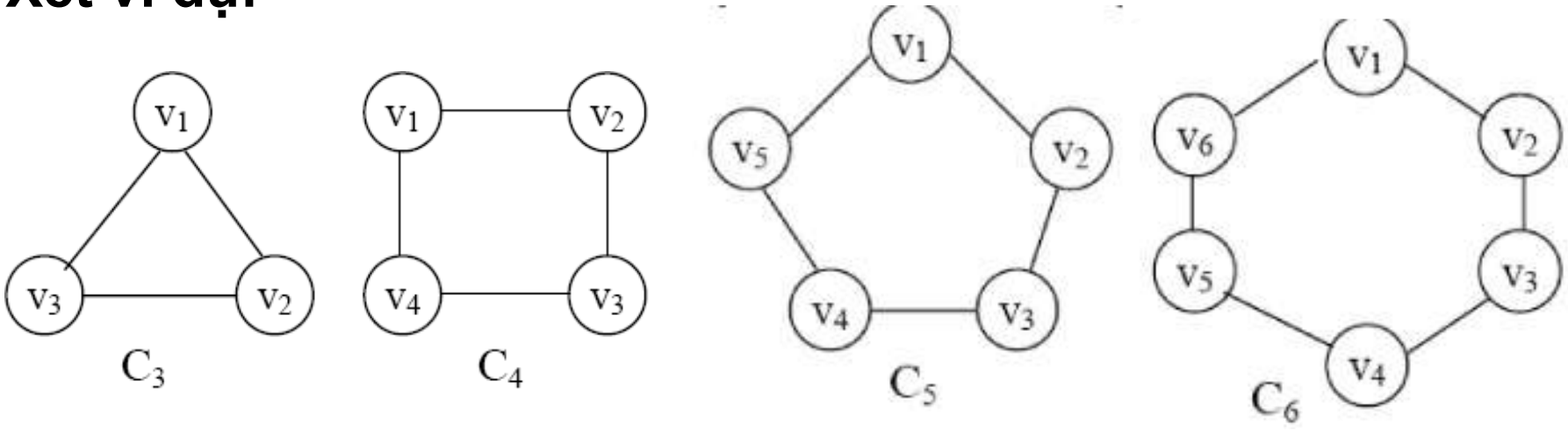
Xét ví dụ:



CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 4 ĐƠN ĐỒ THỊ ĐẶC BIỆT

Đồ thị vòng: Đơn đồ thị n đỉnh v_1, v_2, ..., v_n ($n \geq 3$) và n cạnh (v_1,v_2), (v_2,v_3), ..., (v_{n-1},v_n), (v_n,v_1) được gọi là đồ thị vòng, ký hiệu là C_n. Như vậy, mỗi đỉnh của C_n có bậc là 2.

Xét ví dụ:



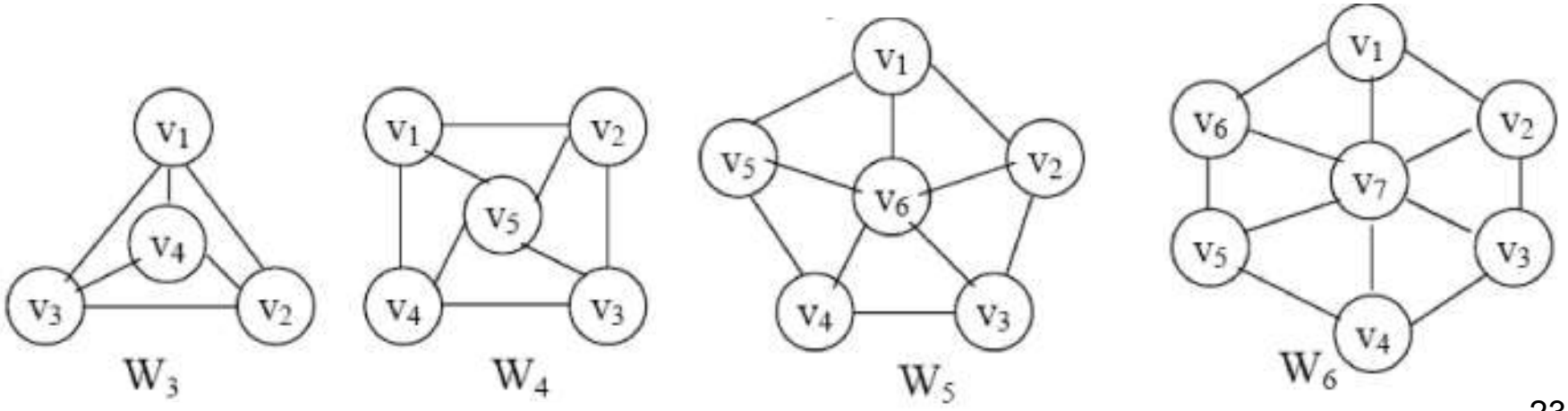
CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 4 ĐƠN ĐỒ THỊ ĐẶC BIỆT

Đồ thị bánh xe: Từ đồ thị vòng C_n, thêm vào đỉnh v_{n+1} và các cạnh (v_{n+1},v_1), (v_{n+1},v_2), ..., (v_{n+1},v_n), ta nhận được đơn đồ thị gọi là đồ thị bánh xe, ký hiệu là W_n.

Như vậy, W_n có n+1 đỉnh, 2n cạnh,

một đỉnh bậc n và n đỉnh bậc 3.

Xét ví dụ:

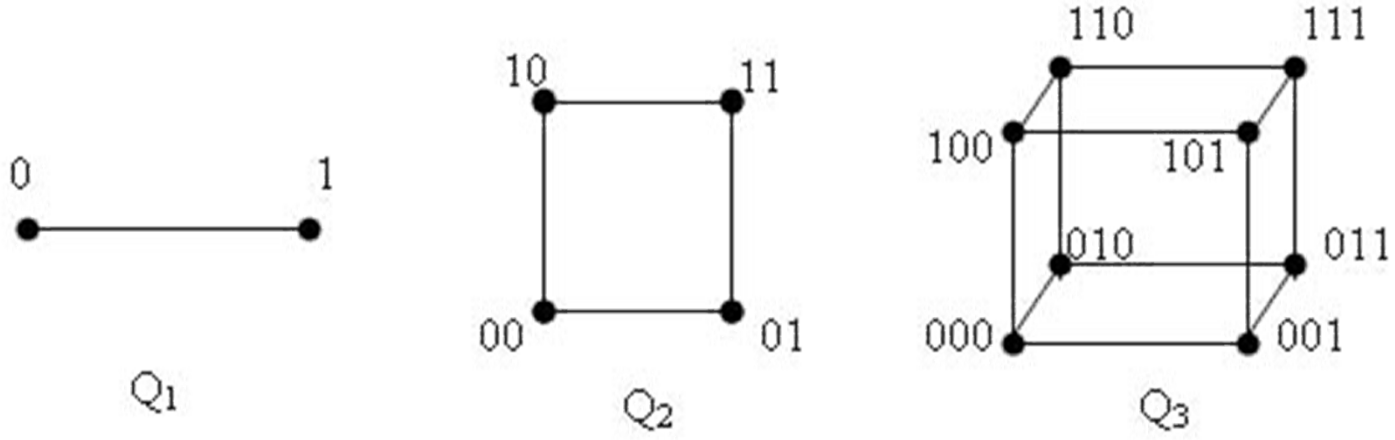


CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 4 ĐƠN ĐỒ THỊ ĐẶC BIỆT

Đồ thị lập phương: Đơn đồ thị 2^n đỉnh, tương ứng với 2^n xâu nhị phân độ dài n và hai đỉnh kề nhau khi và chỉ khi 2 xâu nhị phân tương ứng với hai đỉnh này chỉ khác nhau đúng một bit được gọi là đồ thị lập phương, ký hiệu là Q_n.

- Như vậy, mỗi đỉnh của Q_n có bậc là n

  số cạnh của Q_n là $n.2^{n-1}$

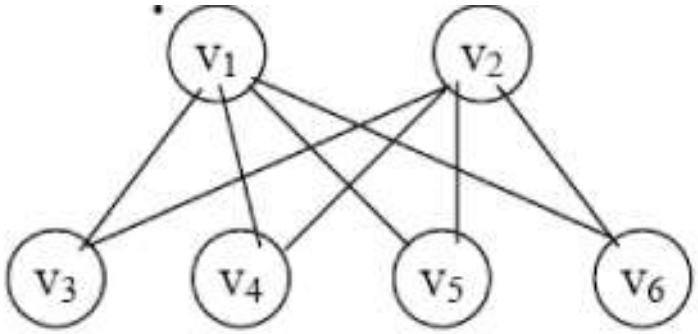


Đồ thị lập phương Q_1, Q_2, Q_3

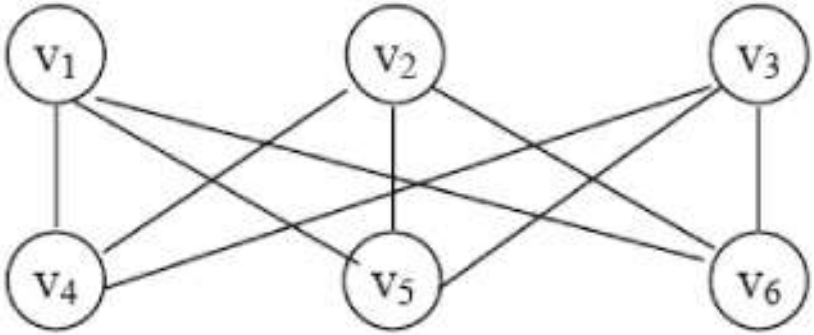
CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 4 ĐƠN ĐỒ THỊ ĐẶC BIỆT

Đồ thị phân đôi (đồ thị hai phe): Đơn đồ thị G=(V,E) sao cho $V=V_1 \cup V_2$, $V_1 \cap V_2=\emptyset$, $V_1 \neq \emptyset$, $V_2 \neq \emptyset$ và mỗi cạnh của G được nối một đỉnh trong V_1 và một đỉnh trong V_2 được gọi là đồ thị phân đôi.

- Nếu đồ thị phân đôi $G=(V_1 \cup V_2, E)$ sao cho với mọi $v_1 \in V_1$, $v_2 \in V_2$, (v1,v2) ϵ E thì G được gọi là đồ thị phân đôi đầy đủ.
- Nếu $|V_1|=m$, $|V_2|=n$ thì đồ thị G được ký hiệu là $K_{m,n}$.
- Như vậy, $K_{m,n}$ có m.n cạnh, các đỉnh của V_1 có bậc n và các đỉnh của V_2 có bậc m.



$K_{2,4}$

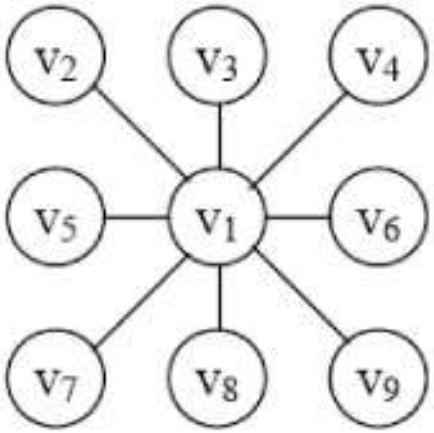


$K_{3,3}$

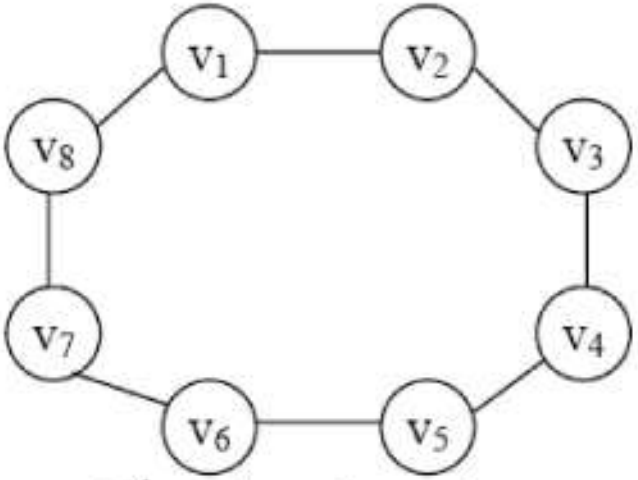
CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 4 ĐƠN ĐỒ THỊ ĐẶC BIỆT

Ứng dụng của các đồ thị đặc biệt:

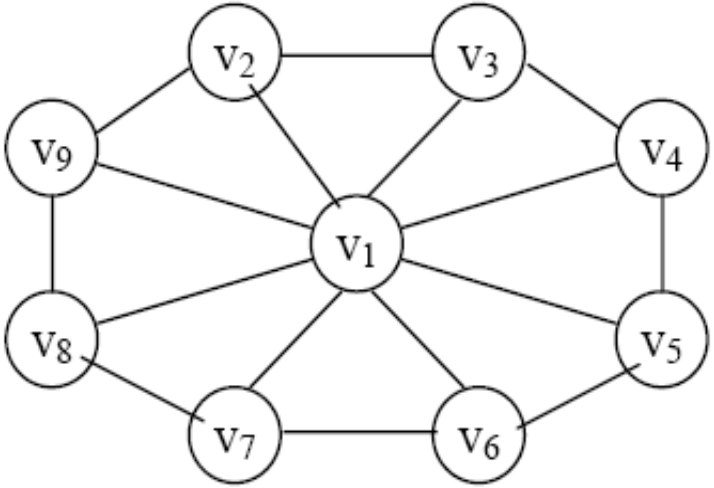
+ Các mạng cục bộ (LAN):



Cấu trúc hình sao



Cấu trúc vòng tròn



Cấu trúc hỗn hợp

Đồ thị phân đôi đầy đủ $K_{1,n}$

Đồ thị vòng C_n

Đồ thị bánh xe W_n

∉

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 5 BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

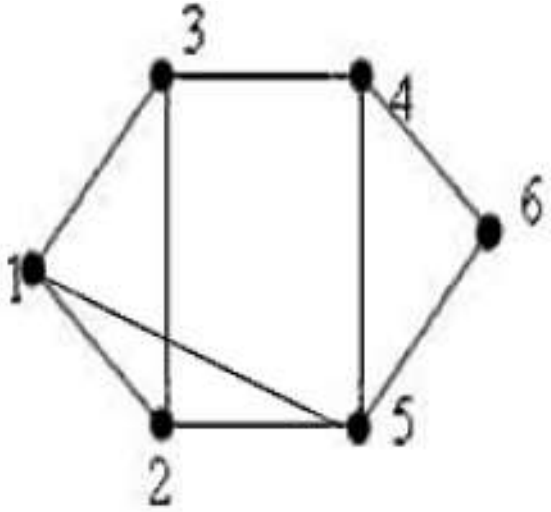
1. Ma trận kề, ma trận trọng số

Xét đơn đồ thị G=(V,E)

Ma trận kề:

Ma trận A={$a_{i,j}$: i,j=1, 2,. . . ,n} với $a_{i,j}$=0, nếu (i,j)∉E và $a_{i,j}$=1, nếu (i,j)ЄE, i, j=1, 2,. . .,n gọi là ma trận kề của đồ thị G.

Ví dụ:



H1. Đồ thị vô hướng G

	1	2	3	4	5	6
1	0	1	1	0	1	0
2	1	0	1	0	1	0
3	1	1	0	1	0	0
4	0	0	1	0	1	1
5	1	1	0	1	0	1
6	0	0	0	1	1	0

Ma trận kề của G

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 5 BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

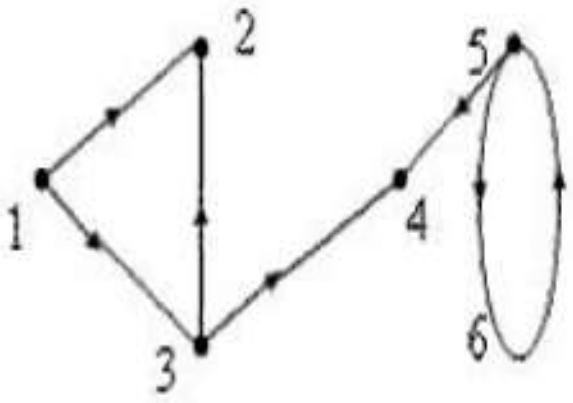
1. Ma trận kề, ma trận trọng số

Xét đơn đồ thị có hướng G=(V,E)

Ma trận kề:

Ma trận A={ai,j : i,j=1, 2,. . . ,n} với $a_{i,j}$=0, nếu (i,j) $\notin$ E và $a_{i,j}$=1, nếu (i,j)ЄE, i, j=1, 2,. . .,n gọi là ma trận kề của đồ thị G.

Ví dụ:



Đồ thị có hướng G_1

	1	2	3	4	5	6
1	0	1	1	0	0	0
2	0	0	0	0	0	0
3	0	1	0	1	0	0
4	0	0	0	0	0	0
5	0	0	0	1	0	1
6	0	0	0	0	1	0

Ma trận kề của G_1

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 5 BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

* Tính chất của ma trận kề của đồ thị vô hướng:

- Tính đối xứng: a[i,j]=a[j,i], i,j=1,2,. . .,n.

- Tổng các phần từ trên dòng i (cột j) bằng bậc của đỉnh i (đỉnh j).

* Tính chất của ma trận kề của đồ thị có hướng:

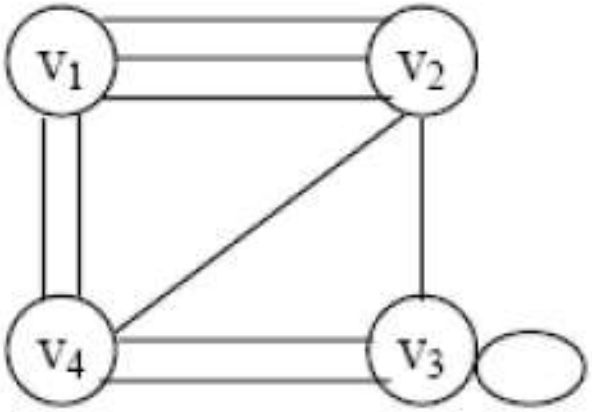
- Không có tính đối xứng

- Tổng các phần từ trên dòng i bằng bán bậc ra của đỉnh i ($deg^+(i)$) và tổng các phần từ trên cột j bằng bán bậc vào của đỉnh j ($deg(^-j)$).

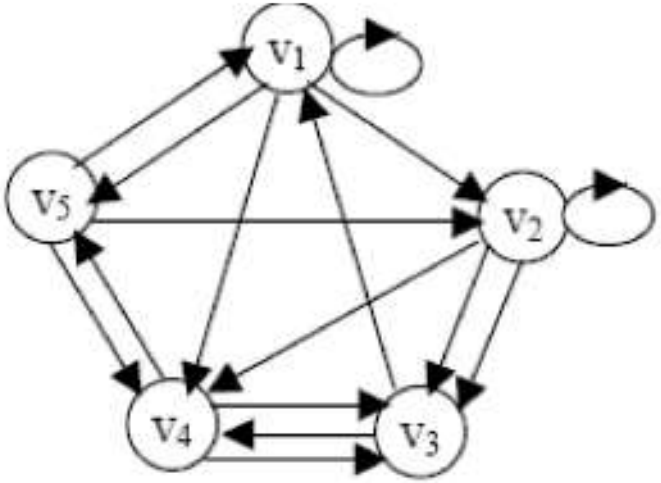
CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 5 BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

1. Ma trận kề, ma trận trọng số

Xét ví dụ 2:



$$\begin{pmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 5 BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

Ma trận trọng số:

Đồ thị có trọng số là đồ thị mà mỗi cạnh (i,j) có một giá trị c(i,j) gọi là trọng số của cạnh.

Để biểu diễn đồ thị ta sử dụng ma trận trọng số C= {c[i,j], i,j=1, 2,. . .,n}

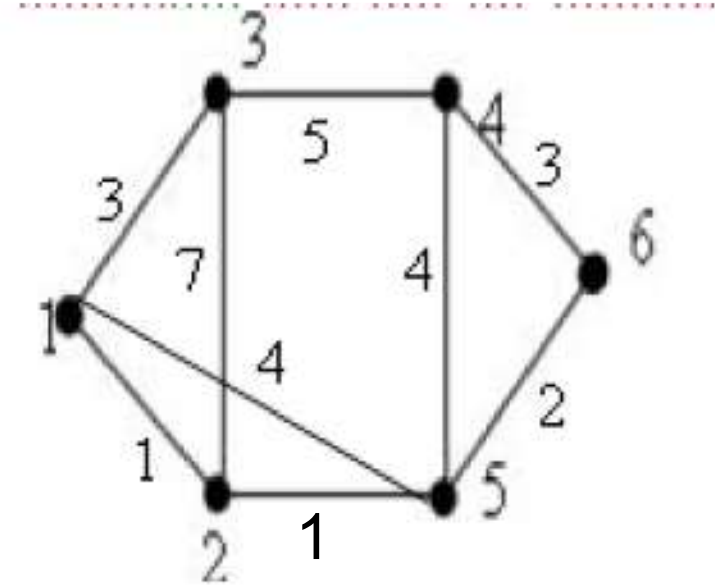
với c[i,j]= c(i,j) nếu (i,j) Є E và c[i,j]=θ nếu (i,j) $\notin$ E

trong đó số θ có thể được đặt bằng một trong các giá trị sau: 0, $+\infty$, $-\infty$.

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 5 BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

Ma trận trọng số:

Ví du:



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

CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 5 BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

Ưu điểm lớn nhất của phương pháp biểu diễn đồ thị bằng ma trận kề (hoặc ma trận trọng số) là để trả lời câu hỏi: Hai đỉnh u,v có kề nhau trên đồ thị hay không, chúng ta chỉ phải thực hiện một phép so sánh.

Nhược điểm lớn nhất của phương pháp này là: không phụ thuộc vào số cạnh của đồ thị, ta luôn phải sử dụng n^2 đơn vị bộ nhớ để lưu trữ ma trận kề của nó.

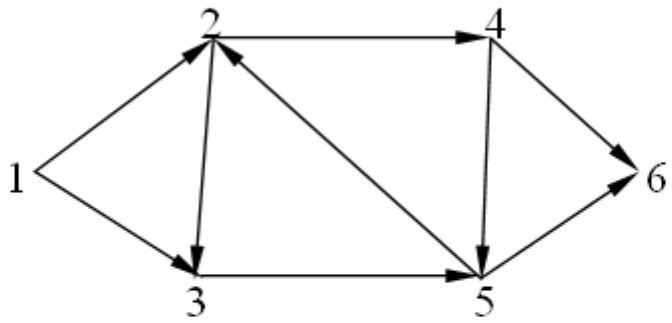
CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 5 BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

2. Ma trận liên thuộc đỉnh-cạnh:

Xét G=(V, E) là đơn đồ thị có hướng. Ma trận liên thuộc đỉnh-cạnh có dạng:

$$a_{ij} = \begin{cases} 1, \text{ nếu i là đỉnh đầu của cung } e_j \\ -1, \text{ nếu i là đỉnh cuối của cung } e_j \\ 0, \text{ nếu i không là đầu mút của cung } e_j \end{cases}$$

Ví dụ: Hình 2



$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{ccccccccc} (1,2) & (1,3) & (2,3) & (2,4) & (3,5) & (4,5) & (4,6) & (5,2) & (5,6) \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \end{pmatrix} \end{array}$$

3. Danh sách cạnh (cung)

+ Trong trường hợp đồ thị thưa (đồ thị có số cạnh m thoả mãn bất đẳng thức: m<6n) biểu diễn đồ thị dưới dạng d/s cạnh.
Lưu trữ danh sách tất cả các cạnh (cung) của đồ thị.
+ Một cạnh (cung) e=(x,y) của đồ thị tương ứng với hai biến Dau[e], Cuoi[e].
+ Để lưu trữ đồ thị ta cần sử dụng 2m đơn vị bộ nhớ.

Nhược điểm: để tìm các đỉnh kề với một đỉnh cho trước phải làm m phép so sánh (khi duyệt qua danh sách tất cả các cạnh của đồ thị).
+ Trong trường hợp đồ thị có trọng số ta cần thêm m đơn vị bộ nhớ để lưu trữ trọng số của các cạnh.

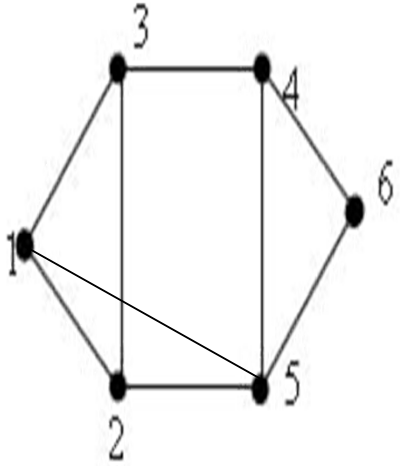
CHƯƠNG I CÁC KHÁI NIỆM CƠ BẢN
BÀI 5 BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

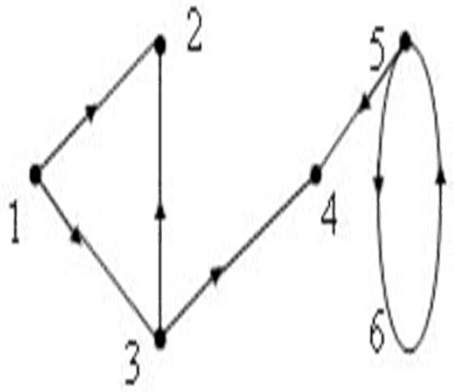
Ví dụ:

D/s cạnh (cung) của G (G_1) (hình 1)

Dau	Cuoi		Dau	Cuoi
1	2		1	2
1	3		1	3
1	5		3	2
2	3		3	4
2	5		5	4
3	4		5	6
4	5		6	5
4	6			
5	6			

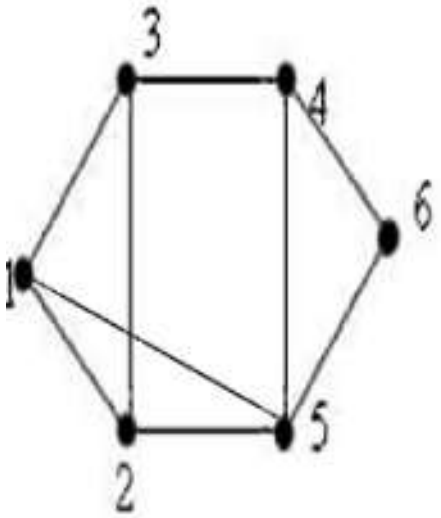
D/s cạnh của G D/s cung của G1

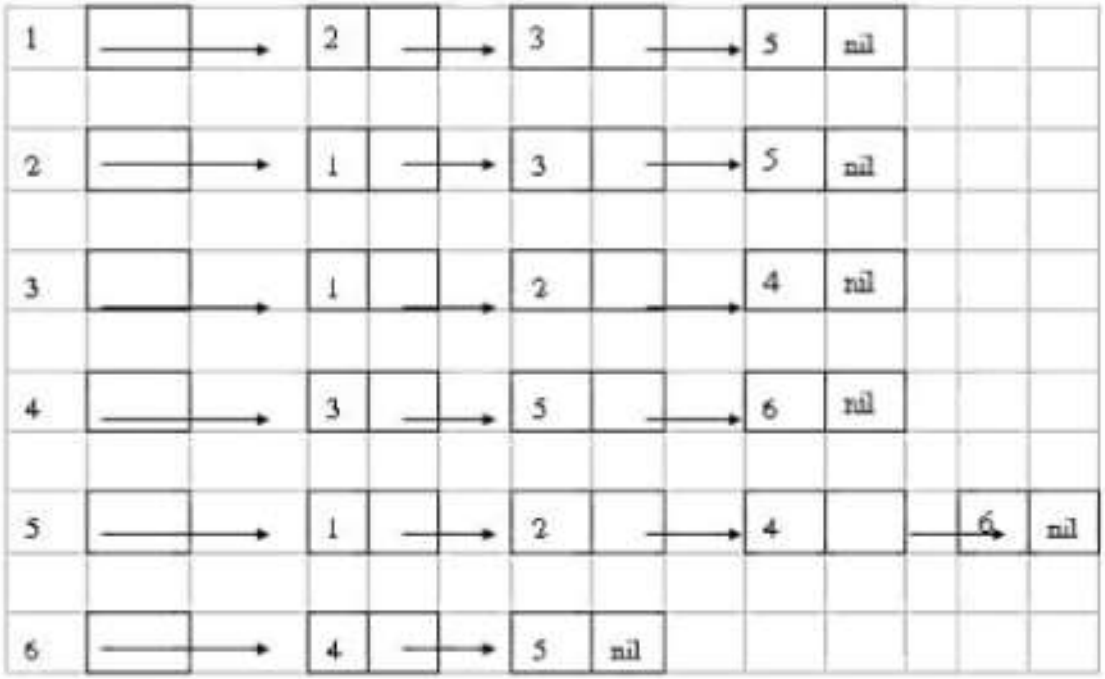




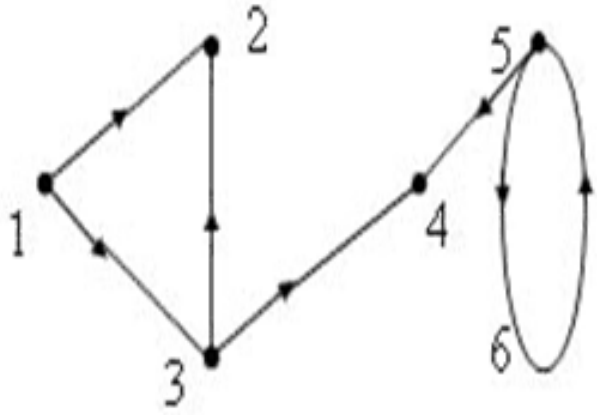
4. Danh sách kề

Với mỗi đỉnh v, ta lưu trữ danh sách các đỉnh kề với v: $Ke(v)=\{u \in V: (v,u) \in E\}$





4. Danh sách kề



Đồ thị có hướng G1

Danh sách kề của G1

Đỉnh đầu

1	→	2	→	3	nil
2	nil				
3	→	2	→	4	nil
4	nil				
5	→	4	→	6	nil
6	→	5	nil		

CHƯƠNG II CÁC THUẬT TOÁN TÌM KIẾM TRÊN ĐỒ THỊ

1. Duyệt đồ thị theo chiều sâu (DFS-Depth First Search)

* Ý tưởng:

- Từ đỉnh v_1 nào đó chưa thăm, thăm v_1, rồi tìm đỉnh v_2 (chưa thăm) kề với v_1, thăm v_2… Thuật toán lặp lại việc thăm cho tới khi tất cả các đỉnh đều được thăm.
- Nếu tại một đỉnh v_i nào đó, không còn đỉnh nào kề với v_i là chưa thăm thì quay trở lại tiếp tục tìm đỉnh kề chưa thăm khác của v_{i-1}.

CHƯƠNG II CÁC THUẬT TOÁN TÌM KIẾM TRÊN ĐỒ THỊ

1. Duyệt đồ thị theo chiều sâu (DFS-Depth First Search)

```
Procedure DFS(v); (*tim kiem theo chieu sau bat dau tu dinh v; cac bien
    Chuaxet, Ke la bien toan cuc*)
Begin
Tham_dinh(v);  Chuaxet[v]:=false;
For u Є Ke(v) do
   If Chuaxet[u] then DFS(u);
End; (*dinh v da duyet xong*)
```

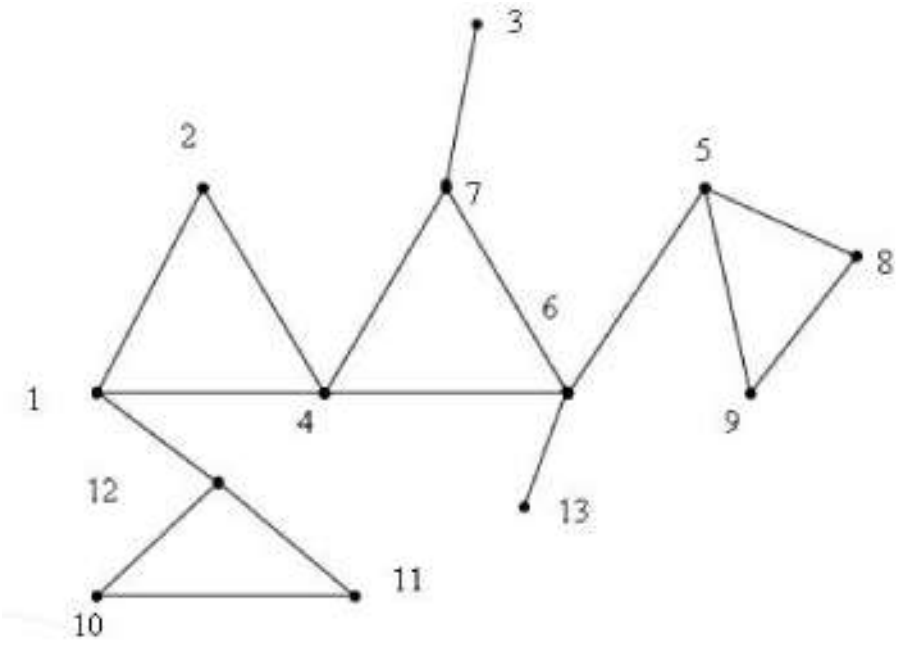
Khi đó, tìm kiếm theo chiều sâu trên đồ thị được thực hiện nhờ thuật toán sau:

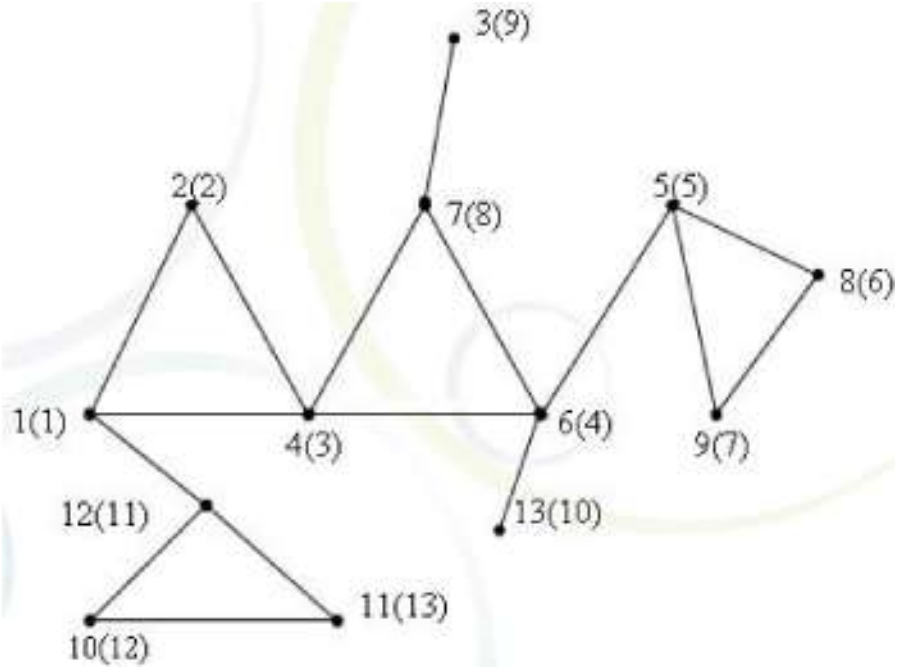
```
Begin
(*Khoi tao tat ca cac dinh cua do thi*)
for v Є V do Chuaxet[v]:=true;
for v Є V do
   If Chuaxet[v] then DFS(v);
End.
```

CHƯƠNG II CÁC THUẬT TOÁN TÌM KIẾM TRÊN ĐỒ THỊ

1. Duyệt đồ thị theo chiều sâu (DFS-Depth First Search)

Ví dụ 1. Xét đồ thị cho trong hình gồm 13 đỉnh, các đỉnh được đánh số từ 1 đến 13 như sau:





CHƯƠNG II CÁC THUẬT TOÁN TÌM KIẾM TRÊN ĐỒ THỊ

2. Duyệt đồ thị theo chiều rộng (BFS-Breadth First Search)

* Ý tưởng:

-Từ đỉnh v nào đó chưa thăm, thăm v, cất tất cả các đỉnh u (chưa thăm) kề với v vào hàng đợi. Lấy từ hàng đợi một đỉnh u, thăm u, rồi lại cất tất cả các đỉnh t (chưa thăm) kề với u vào hàng đợi…

Thuật toán lặp lại việc thăm cho tới khi hàng đợi rỗng.

- Nếu tại một đỉnh x nào đó, không còn đỉnh nào kề với x là chưa thăm thì quay trở lại tiếp tục tìm đỉnh kề chưa thăm khác của y (y là đỉnh trước khi đến x).

CHƯƠNG II CÁC THUẬT TOÁN TÌM KIẾM TRÊN ĐỒ THỊ

2. Duyệt đồ thị theo chiều rộng (BFS-Breadth First Search)

```
Procedure BFS(v);(*BFS bat dau tu dinh v, cac bien Chuaxet, Ke la bien cuc bo*)
Begin  QUEUE:=Ø; QUEUE⇐ v; (*ket qua nap vao QUEUE*)
   Chuaxet[v]:=false;
   While QUEUE<> Ø do
   Begin
   p ⇐ QUEUE; (*lay p tu QUEUE:*)      Tham_dinh(p);
   For u Є Ke(v) do
   If Chuaxet[u] then
   Begin  QUEUE ⇐ u; Chuaxet[u]:=false; End;
End; End;
```

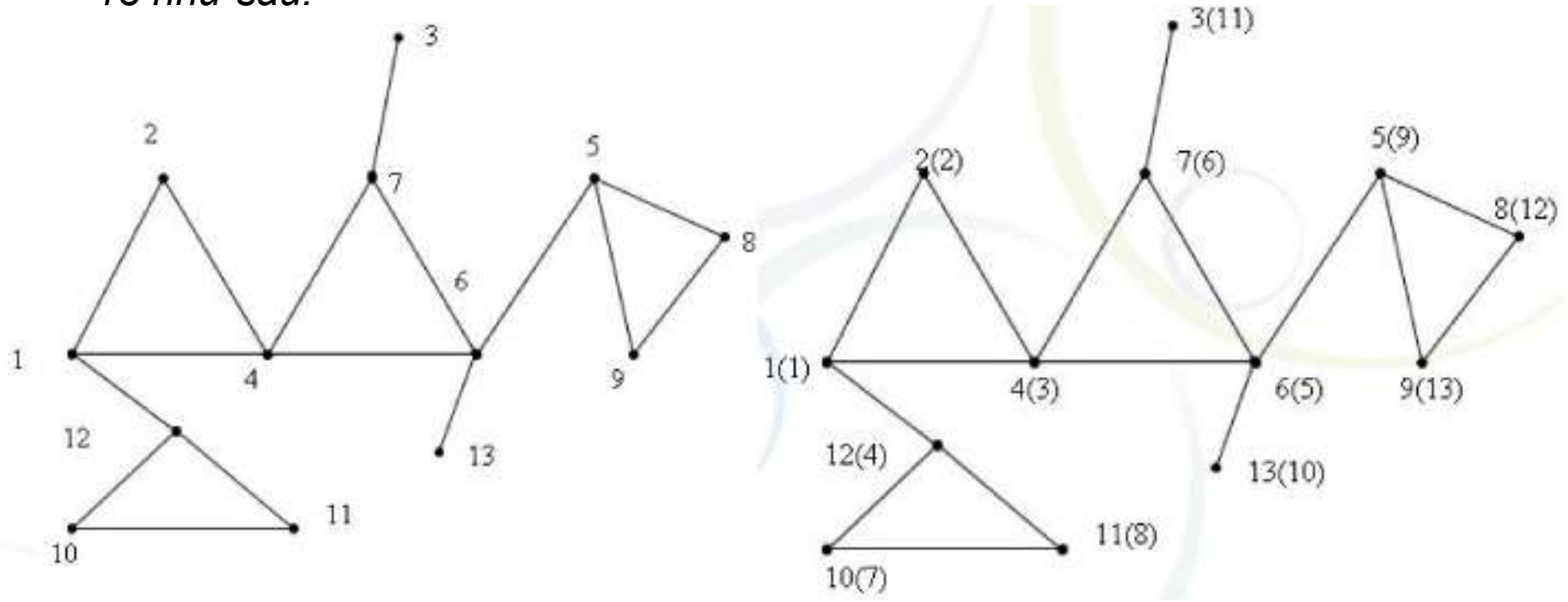
Khi đó, tìm kiếm theo chiều rộng trên đồ thị được thực hiện nhờ thuật toán sau:

```
Begin
   for f Є V do Chuaxet[v]:=true; (*Khoi tao cac dinh cua do thi la chua xet*)
   for v Є V do
   if Chuaxet[v] then BFS(v);
End.
```

CHƯƠNG II CÁC THUẬT TOÁN TÌM KIẾM TRÊN ĐỒ THỊ

2. Duyệt đồ thị theo chiều rộng (BFS-Breadth First Search)

Ví dụ 2. Xét đồ thị cho trong hình gồm 13 đỉnh, các đỉnh được đánh số từ 1 đến 13 như sau:



CHƯƠNG II CÁC THUẬT TOÁN TÌM KIẾM TRÊN ĐỒ THỊ

3. Tìm đường đi và kiểm tra tính liên thông:

a) Bài toán tìm đường đi giữa hai đỉnh:

Giả sử s và t là hai đỉnh nào đó của đồ thị. Hãy tìm đường đi từ s đến t.

* Ý tưởng:

- Gọi thủ tục DFS(s) hoặc (BFS(s)) để thăm tất cả các đỉnh thuộc cùng một thành phần liên thông với s.
- Nếu sau khi thực hiện xong thủ tục mà Chuaxet[t]=true thì không có đường đi từ s đến t,

ngược lại thì có đường đi từ s đến t.

- Để ghi nhận đường đi, ta dùng thêm biến Truoc[v] để ghi nhận đỉnh đi trước đỉnh v trong đường đi từ s đến v.

CHƯƠNG II CÁC THUẬT TOÁN TÌM KIẾM TRÊN ĐỒ THỊ

3. Tìm đường đi và kiểm tra tính liên thông:

a) Bài toán tìm đường đi giữa hai đỉnh:

Khi đó, thủ tục DFS(v) cần sửa câu lệnh if trong nó như sau:

```
If Chuaxet[u] then
Begin
   Truoc[u]:=v;
   DFS(u);
End;
```

Thủ tục BFS(v) cần sửa đổi câu lện if trong nó như sau:

```
If Chuaxet [u] then
Begin
   QUEUE⇐ u;
   Chuaxet[u]:=false;
   Truoc[u]:=p;
End;
```

CHƯƠNG II CÁC THUẬT TOÁN TÌM KIẾM TRÊN ĐỒ THỊ

3. Tìm đường đi và kiểm tra tính liên thông:

b) Tìm các thành phần liên thông của đồ thị:

Hãy cho biết đồ thị gồm bao nhiêu thành phần liên thông và từng thành phần liên thông của nó là gồm những đỉnh nào.

+ Do thủ tục DFS(v) (BFS(s)) cho phép thăm tất cả các đỉnh thuộc cùng một thành phần liên thông với s, nên số thành phần liên thông của đồ thị bằng số lần gọi đến thủ tục này.
+ Vấn đề còn lại là cách ghi nhận các đỉnh trong từng thành phần liên thông.

Ta dùng thêm biến Index[v] để ghi nhận chỉ số của thành phần liên thông chứa đỉnh v, và biến Inconnect để đếm số thành phần liên thông (khởi tạo giá trị 0).

CHƯƠNG II CÁC THUẬT TOÁN TÌM KIẾM TRÊN ĐỒ THỊ

3. Tìm đường đi và kiểm tra tính liên thông:

b) Tìm các thành phần liên thông của đồ thị:

+ Thủ tục Tham_dinh(v) trong các thủ tục DFS(v) và BFS(v) có nhiệm vụ gán: Index[v]:=Inconnect;
+ Câu lệnh if trong các chương trình chính gọi đến các thủ tục này cần được sửa lại như sau:

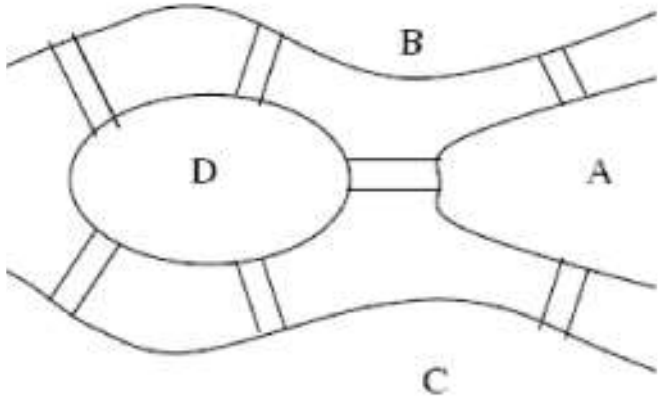
```
Inconnect:=0;
If Chuaxet[v] then
Begin
    Inconnect:=Inconnect+1;
    DFS(v); (*BFS(v)*)
End;
```

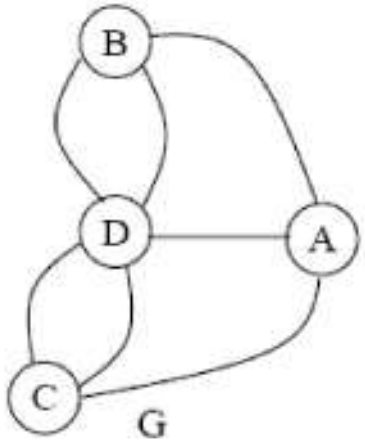
+ Kết thúc vòng lặp thứ hai trong chương trình chính, Inconnect trả về số thành phần liên thông của đồ thị, biến mảng Index[v], v Є V cho phép liệt kê các đỉnh thuộc cùng một thành phần liên thông.

CHƯƠNG III ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

1. Đồ thị EULER:

- 1736 Euler (1707-1783) công bố lời giải "bài toán về các cầu ở Konigsberg".





Bài toán tìm đường đi qua tất cả các cầu, mỗi cầu chỉ qua một lần có thể được phát biểu lại bằng mô hình như sau:

❑ Có tồn tại chu trình đơn trong đa đồ thị G chứa tất cả các cạnh?

CHƯƠNG III ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

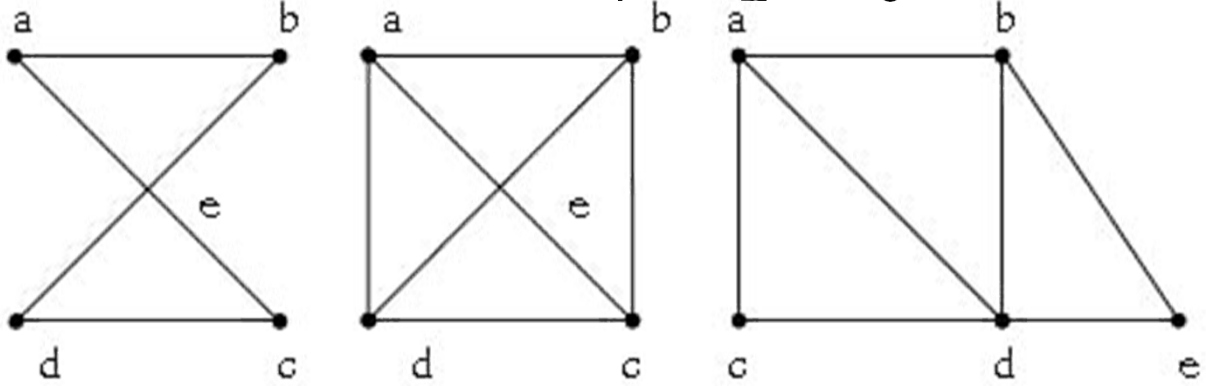
1. Đồ thị EULER:

- Đường đi qua mỗi cạnh của đồ thị đúng một lần được gọi là đường đi Euler.
- Chu trình qua mỗi cạnh của đồ thị đúng một lần được gọi là chu trình Euler.
- Đồ thị được gọi là đồ thị Euler nếu nó có chu trình Euler, và gọi là đồ thị nửa Euler nếu nó có đường đi Euler
- Nhận xét: mọi đồ thị Euler luôn là nửa Euler, nhưng điều ngược lại không luôn đúng.

CHƯƠNG III ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

1. Đồ thị EULER:

- Ví dụ 1: Xét 3 đồ thị G_1, G_2, G_3 bên dưới:



Đồ thị G_1 trong hình là đồ thị Euler vì nó có chu trình Euler a, e, c, d, e, b, a.

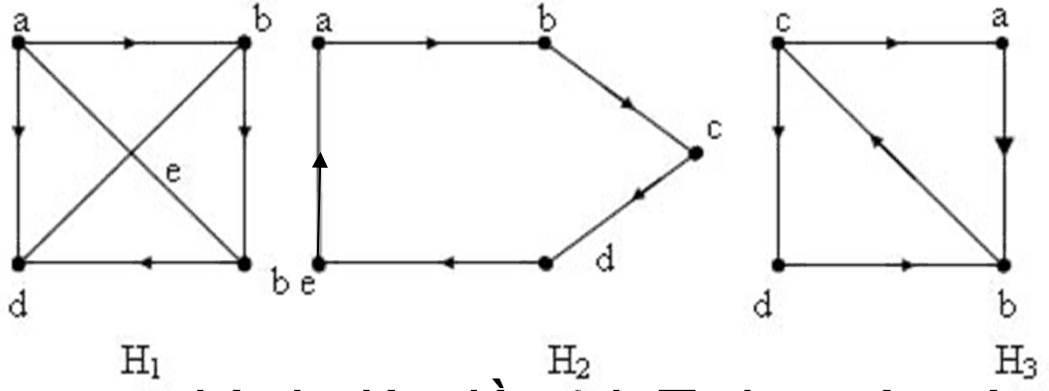
Đồ thị G_3 không có chu trình Euler nhưng nó có đường đi Euler a, c, d, e, b, d, a, b, vì thế G_3 là đồ thị nửa Euler.

Đồ thị G_2 không có chu trình cũng như đường đi Euler.

CHƯƠNG III ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

1. Đồ thị EULER:

- Ví dụ 2: Xét 3 đồ thị H_1, H_2, H_3 bên dưới:



Đồ thị H_2 trong hình là đồ thị Euler vì nó có chu trình Euler a, b, c, d, e, a.

Đồ thị H_3 không có chu trình Euler nhưng nó có đường đi Euler c, d, b, c, a, b, vì thế H_3 là đồ thị nửa Euler.

Đồ thị H_1 không có chu trình cũng như đường đi Euler.

CHƯƠNG III ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

1. Đồ thị EULER:

- **Định lý 1 (Euler):** G là đồ thị vô hướng liên thông. G là đồ thị Euler $\Leftrightarrow$ mọi đỉnh của G đều có bậc chẵn.
- Bổ đề: Nếu bậc của mỗi đỉnh của đồ thị G không nhỏ hơn 2 thì G chứa chu trình.
- Hệ quả: Đồ thị vô hướng liên thông G là nửa Euler khi và chỉ khi nó có không quá 2 đỉnh bậc lẻ.

CHƯƠNG III ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

1. Đồ thị EULER:

- Thuật toán Flor:

Xuất phát từ một đỉnh u nào đó của G ta đi theo các cạnh của nó một cách tuỳ ý chỉ cần tuân thủ 2 qui tắc sau:

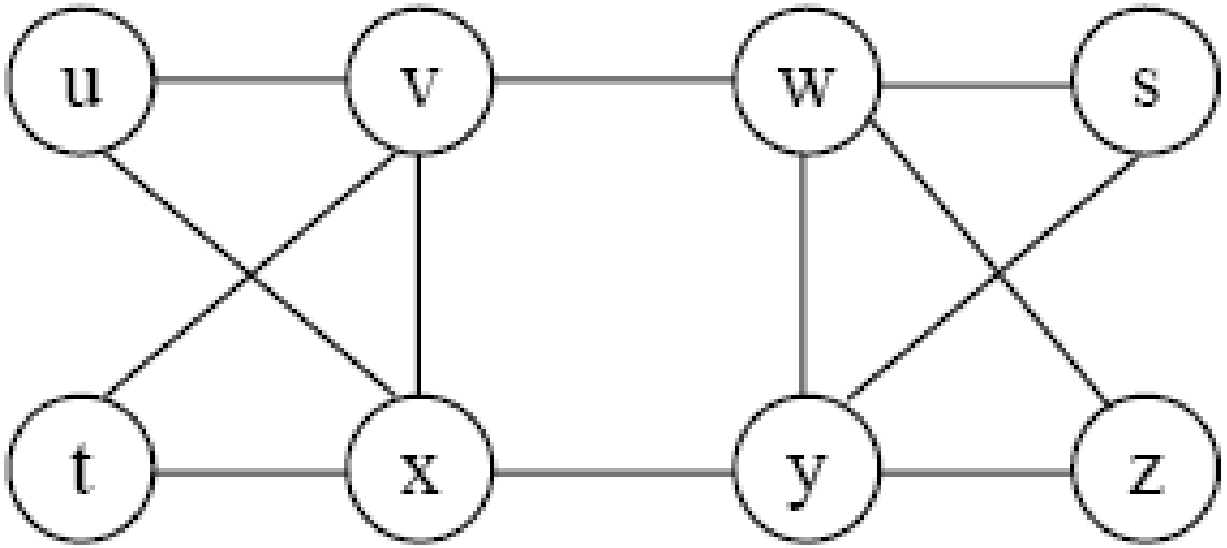
(1) Xoá bỏ cạnh đã đi qua đồng thời xoá bỏ đỉnh cô lập tạo thành.

(2) Ở mỗi bước ta chỉ đi qua cầu khi không còn cách lựa chọn nào khác.

CHƯƠNG III ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

1.Đồ thị EULER:

Xét ví dụ: Tìm chu trình Euler trong đồ thị:



❑ Định lý 2. G có hướng liên thông mạnh G là đồ thị Euler $\Leftrightarrow \deg^+(v) = \deg^-(v), \forall v \in V$.

CHƯƠNG III ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

1. Đồ thị EULER:

- Bài toán người phát thư Trung Hoa (Guan 1960):

Một NV đi từ Sở BĐ, qua một số đường phố để phát thư, rồi quay về Sở. Phải đi qua các đường theo trình tự nào để đường đi là ngắn nhất?

Xét bài toán: Cho đồ thị liên thông G. Một chu trình qua mọi cạnh của G gọi là một hành trình trong G. Hãy tìm hành trình ngắn nhất (qua ít cạnh nhất).

Nếu G là đồ thị Euler thì chu trình Euler trong G là hành trình ngắn nhất cần tìm.

Xét trường hợp G có một số đỉnh bậc lẻ. Khi đó, mọi hành trình trong G phải đi qua ít nhất hai lần một số cạnh nào đó.

CHƯƠNG III ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

1. Đồ thị EULER:

- Bài toán người phát thư Trung Hoa (Guan 1960):

Định lý (Gooodman và Hedetniemi, 1973). Nếu G là một đồ thị liên thông có q cạnh thì hành trình ngắn nhất trong G có chiều dài q + m(G),

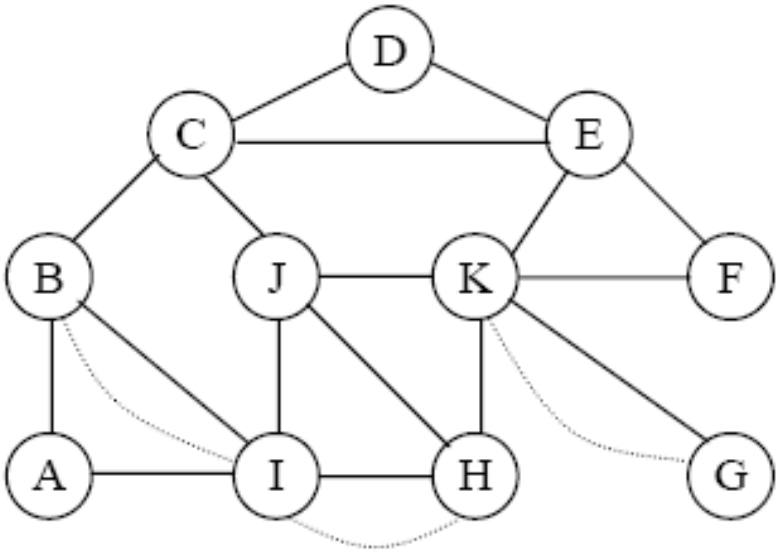
trong đó m(G) là số cạnh mà hành trình đi qua hai lần và được xác định như sau:

Gọi $V_0(G)$ là tập hợp các đỉnh bậc lẻ (2k đỉnh) của G. Ta phân 2k phần tử của G thành k cặp, mỗi tập hợp k cặp gọi là một phân hoạch cặp của $V_0(G)$.

Gọi độ dài đường đi ngắn nhất từ u đến v là khoảng cách d(u,v). Đối với mọi phân hoạch cặp P_i, tính khoảng cách giữa hai đỉnh trong từng cặp, tính tổng d(Pi).

Số m(G) bằng cực tiểu của các d(Pi): m(G)=min d(Pi).

1. Đồ thị EULER:



G_T

$V_O(G)=\{B, G, H, K\}$
tập hợp các phân hoạch cặp là:
$P=\{P_1, P_2, P_3\}$, trong đó:
$P_1 = \{(B, G), (H, K)\} \rightarrow d(P_1) = d(B, G)+d(H, K) = 4+1 = 5$
$P_2 = \{(B, H), (G, K)\} \rightarrow d(P_2) = d(B, H)+d(G, K) = 2+1 = 3,$
$P_3 = \{(B, K), (G, H)\} \rightarrow d(P_3) = d(B, K)+d(G, H) = 3+2 = 5.$
$m(G) = \min(d(P_1), d(P_2), d(P_3)) = 3.$
G_T có được từ G bằng cách thêm vào 3 cạnh: (B, I), (I, H), (G, K) và G_T là đồ thị Euler.
Vậy hành trình ngắn nhất cần tìm là đi theo chu trình Euler trong G_T:
A, B, C, D, E, F, K, G, K, E, C, J, K, H, J, I, H, I, B, I, A.

CHƯƠNG III ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

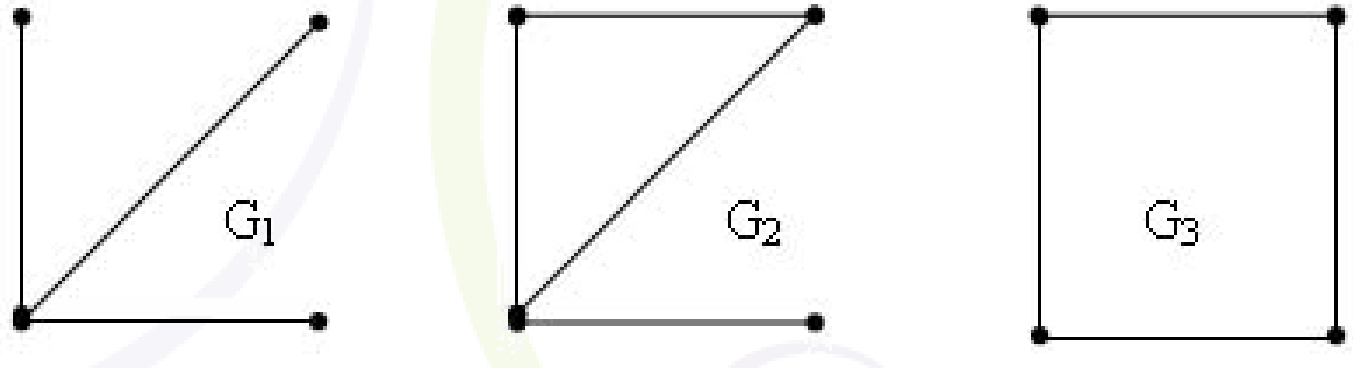
2. Đồ thị HAMILTON

- Đường đi qua tất cả các đỉnh của đồ thị mỗi đỉnh đúng một lần được gọi là đường đi Hamilton.
- Chu trình bắt đầu từ một đỉnh v nào đó qua tất cả các đỉnh còn lại mỗi đỉnh đúng một lần rồi quay trở về v được gọi là chu trình Hamilton.
- Đồ thị G được gọi là đồ thị Hamilton nếu nó chứa chu trình Hamilton và gọi là đồ thị nửa Hamilton nếu nó có đường đi Hamilton.

CHƯƠNG III ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

2. Đồ thị HAMILTON

Ví dụ 3. Trong hình: Đồ thị G3 là Hamilton, G2 là nửa Hamilton còn G1 không là nửa Hamilton.



Cho đến nay việc tìm một tiêu chuẩn nhận biết đồ thị Hamilton vẫn còn là mở.

Phần lớn các phát biểu đều có dạng "nếu G có số cạnh đủ lớn thì G là Hamilton"

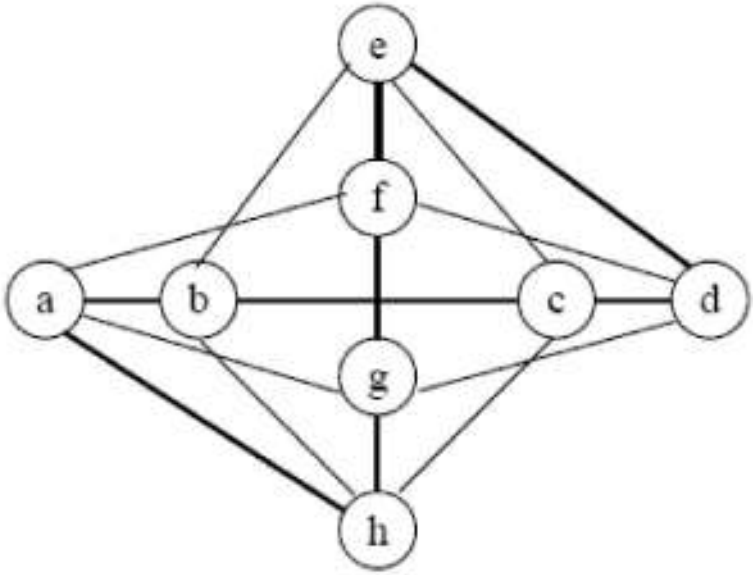
CHƯƠNG III ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

2. Đồ thị HAMILTON

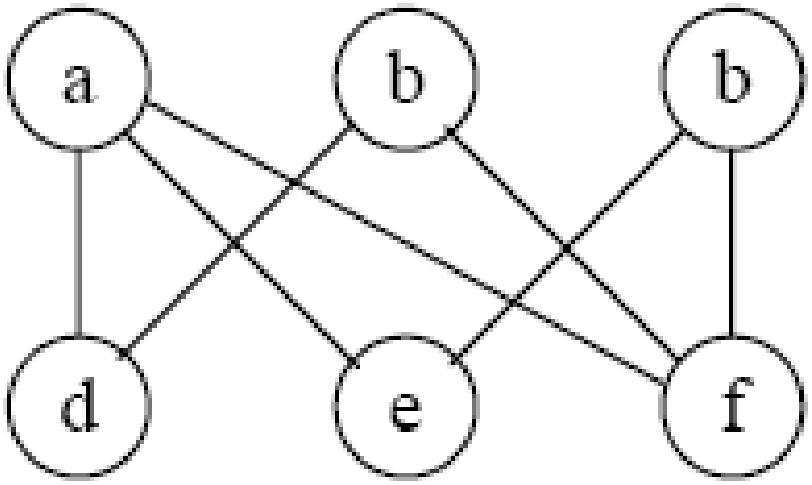
- Định lý 1 (Dirak 1952). Đơn đồ thị vô hướng G với n>2 đỉnh, mỗi đỉnh có bậc không nhỏ hơn n/2 là đồ thị Hamilton.
- Định lý 2 Nếu G là đồ thị phân đôi với hai tập đỉnh là V_1, V_2 có số đỉnh cùng bằng n ($n \geq 2$) và bậc của mỗi đỉnh lớn hơn n/2 thì G là một đồ thị Hamilton.
- Định lý 3. Giả sử G là đồ có hướng liên thông với n đỉnh. Nếu $deg^+(v) \geq n/2$, $deg^-(v) \geq n/2$, $\forall v$ thì G là Hamilton.

CHƯƠNG III ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

2. Đồ thị HAMILTON



Đồ thị G này có 8 đỉnh, đỉnh nào cũng có bậc 4, nên G là đồ thị Hamilton.

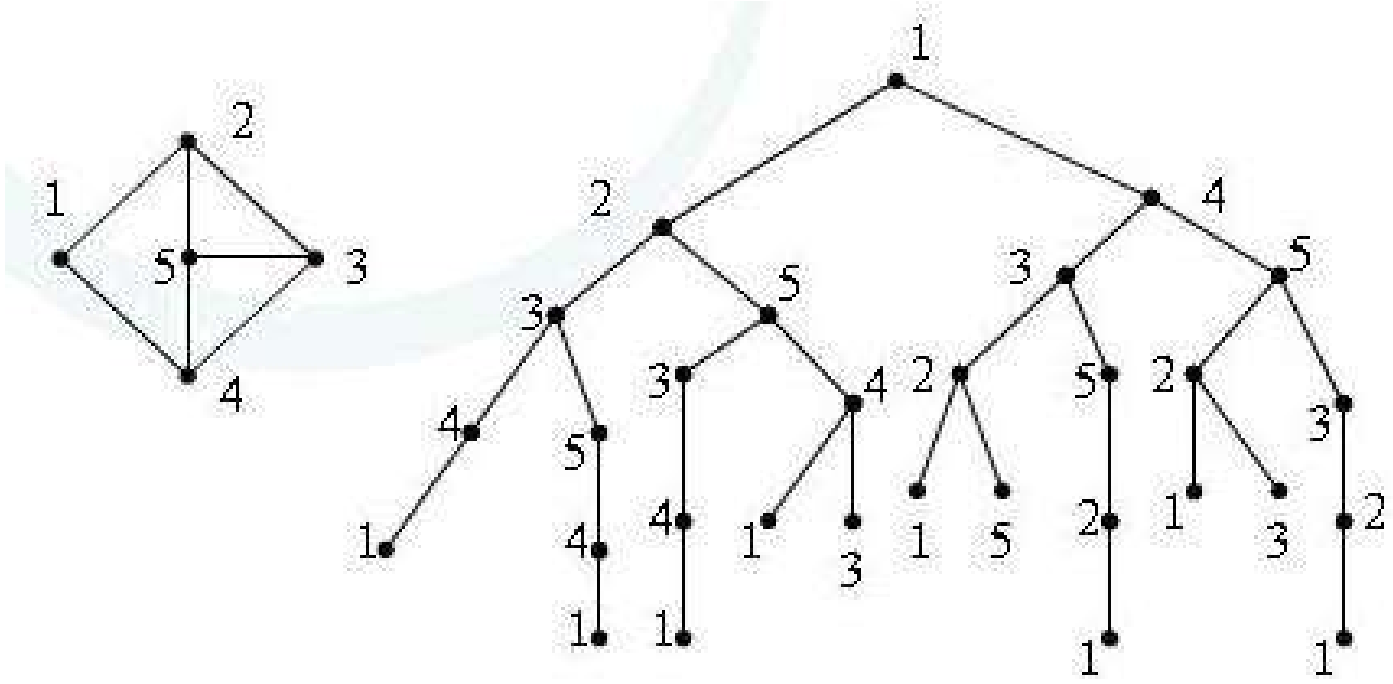


Đồ thị phân đôi này có bậc của mỗi đỉnh bằng 2 hoặc 3 (> 3/2), nên nó là đồ thị Hamilton.

CHƯƠNG III ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

2. Đồ thị HAMILTON

Ví dụ 4. Hình dưới đây mô tả cây tìm kiếm tất cả các chu trình Hamilton của đồ thị.



2. Đồ thị HAMILTON

Bài toán sắp xếp chỗ ngồi:

Có n đại biểu đến dự hội nghị. Mỗi ngày họp một lần ngồi quanh một bàn tròn. Hỏi phải bố trí bao nhiêu ngày và bố trí như thế nào sao cho trong mỗi ngày, mỗi người có hai người kế bên là bạn mới. Lưu ý rằng n người đều muốn làm quen với nhau.

Xét đồ thị gồm n đỉnh, mỗi đỉnh ứng với mỗi người dự hội nghị, hai đỉnh kề nhau khi hai đại biểu tương ứng muốn làm quen với nhau. Như vậy, ta có đồ thị đầy đủ K_n.

CHƯƠNG III ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

2. Đồ thị HAMILTON

Mỗi chu trình Hamilton là một cách sắp xếp như yêu cầu của bài toán. Bái toán trở thành tìm các chu trình Hamilton phân biệt của đồ thị đầy đủ K_n (hai chu trình Hamilton gọi là phân biệt nếu chúng không có cạnh chung).

Định lý: Đồ thị đầy đủ K_n với n lẻ và $n \geq 3$ có đúng $(n-1)/2$ chu trình Hamilton phân biệt.

2. Đồ thị HAMILTON

Giải bài toán sắp xếp chỗ ngồi với n=11.

Có (11−1)/2=5 cách sắp xếp chỗ ngồi phân biệt như sau:

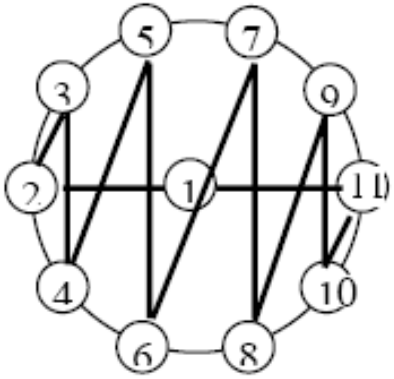
1 2 3 4 5 6 7 8 9 10 11 1

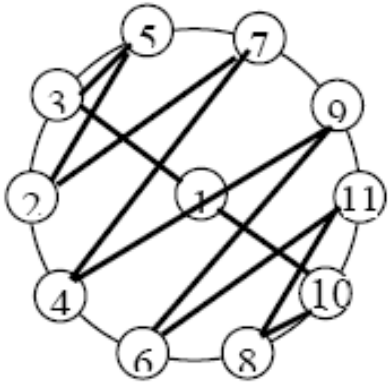
1 3 5 2 7 4 9 6 11 8 10 1

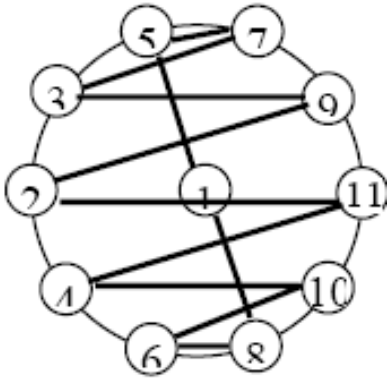
1 5 7 3 9 2 11 4 10 6 8 1

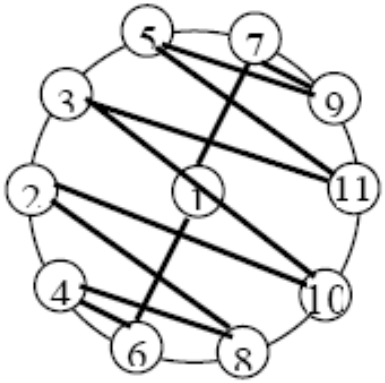
1 7 9 5 11 3 10 2 8 4 6 1

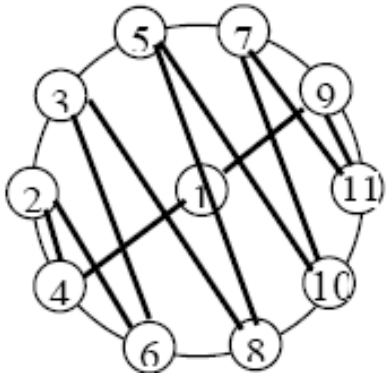
1 9 11 7 10 5 8 3 6 2 4 1











CHƯƠNG IV CÂY VÀ CÂY KHUNG CỦA ĐỒ THỊ

1. Định nghĩa và các tính chất cơ bản

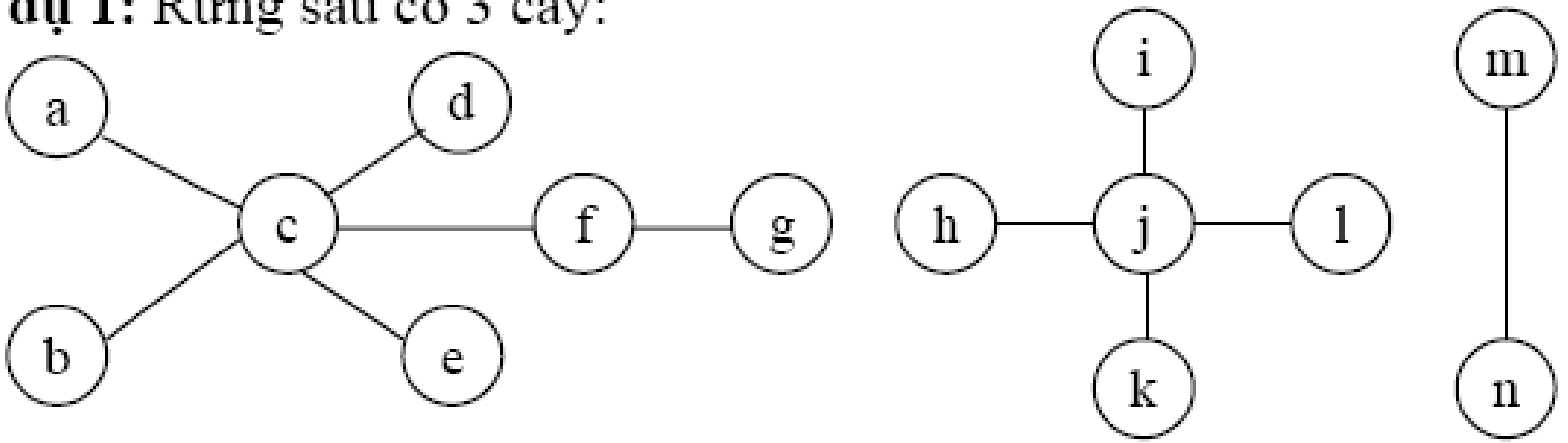
Định nghĩa:

Cây là một đồ thị vô hướng liên thông, không chứa chu trình và có ít nhất hai đỉnh.

Một đồ thị vô hướng không chứa chu trình và có ít nhất hai đỉnh gọi là một rừng.

Trong một rừng, mỗi thành phần liên thông là một cây.

Thí dụ 1: Rừng sau có 3 cây:



CHƯƠNG IV CÂY VÀ CÂY KHUNG CỦA ĐỒ THỊ

1. Định nghĩa và các tính chất cơ bản:

Định lý: Cho T là một đồ thị có $n \geq 2$ đỉnh. Các điều sau là tương đương:

1) T là một cây.
2) T liên thông và có $n-1$ cạnh.
3) T không chứa chu trình và có $n-1$ cạnh.
4) T liên thông và mỗi cạnh là cầu.
5) Giữa hai đỉnh phân biệt bất kỳ của T luôn có duy nhất một đường đi đơn.
6) T không chứa chu trình nhưng khi thêm một cạnh mới thì có được một chu trình duy nhất.

CHƯƠNG IV CÂY VÀ CÂY KHUNG CỦA ĐỒ THỊ

2. Cây khung và bài toán tìm cây khung nhỏ nhất:

Định nghĩa: Trong đồ thị liên thông G, nếu ta loại bỏ cạnh nằm trên chu trình nào đó thì ta sẽ được đồ thị vẫn là liên thông. Nếu cứ loại bỏ các cạnh ở các chu trình khác cho đến khi nào đồ thị không còn chu trình (vẫn liên thông) thì ta thu được một cây nối các đỉnh của G. Cây đó gọi là cây khung hay cây bao trùm của đồ thị G.

CHƯƠNG IV CÂY VÀ CÂY KHUNG CỦA ĐỒ THỊ

2. Cây khung và bài toán tìm cây khung nhỏ nhất:

Bài toán được phát biểu như sau:

Cho G=(V,E) là đồ thị vô hướng liên thông có trọng số, mỗi cạnh $e \in E$ có trọng số $m(e) \geq 0$. Giả sử $T=(V_T,E_T)$ là cây khung của đồ thị G ($V_T=V$). Ta gọi độ dài m(T) của cây khung T là tổng trọng số của các cạnh của nó:

$$m(T) = \sum_{e \in E_T} m(e)$$

CHƯƠNG IV CÂY VÀ CÂY KHUNG CỦA ĐỒ THỊ

2. Cây khung và bài toán tìm cây khung nhỏ nhất:

Bài toán đặt ra là trong số tất cả các cây khung của đồ thị G, hãy tìm cây khung có độ dài nhỏ nhất.

Cây khung như vậy được gọi là cây khung nhỏ nhất của đồ thị.

Bài toán được gọi là "***bài toán tìm cây khung nhỏ nhất***".

- Hai mô hình thực tế tiêu biểu:
  - * *Bài toán xây dựng hệ thống đường sắt.*
  - * *Bài toán nối mạng máy tính.*

CHƯƠNG IV CÂY VÀ CÂY KHUNG CỦA ĐỒ THỊ

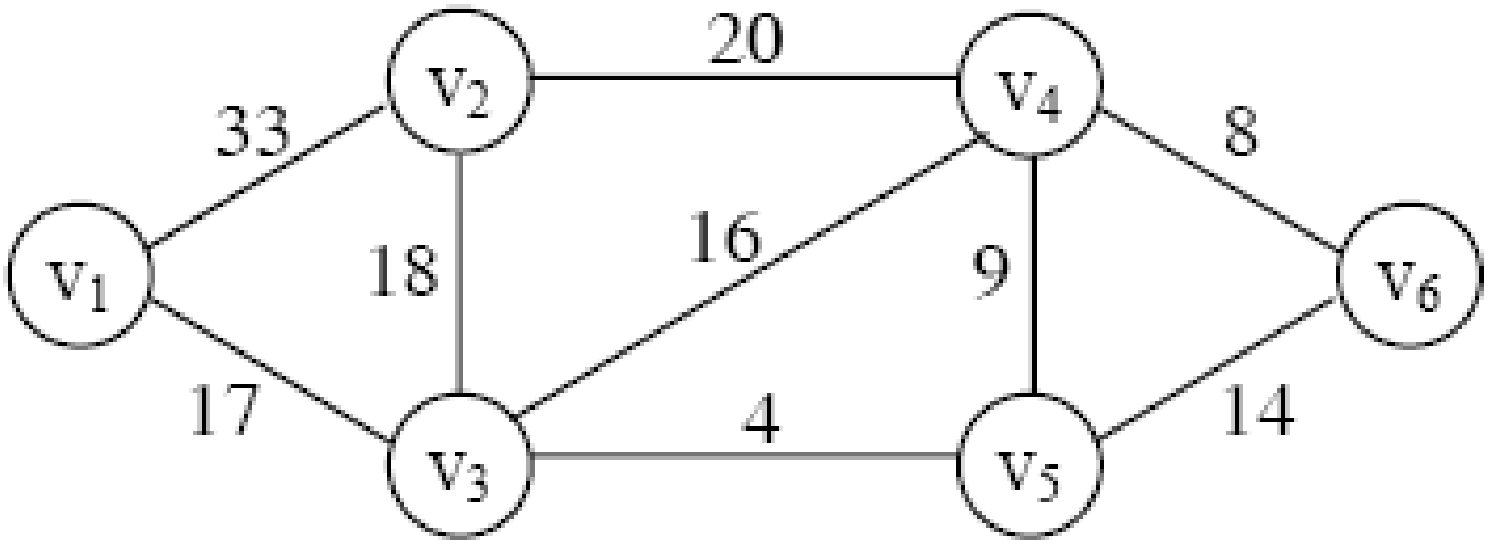
3. Thuật toán Kruskal: Thuật toán sẽ xây dựng tập cạnh E_T của cây khung nhỏ nhất $T=(V_T,E_T)$ theo từng bước:

1. Bắt đầu từ đồ thị rỗng T có n đỉnh.
2. Sắp xếp các cạnh của G theo thứ tự không giảm của trọng số.
3. Bắt đầu từ cạnh đầu tiên của dãy này, thêm dần các cạnh của dãy đã được xếp vào T theo nguyên tắc cạnh thêm vào không được tạo thành chu trình trong T.
4. Lặp lại Bước 3 cho đến khi nào số cạnh trong T bằng n−1, ta thu được cây khung nhỏ nhất cần tìm.

CHƯƠNG IV CÂY VÀ CÂY KHUNG CỦA ĐỒ THỊ

3. Thuật toán Kruskal:

Xét thí dụ: Tìm cây khung nhỏ nhất cho bởi đồ thị:



4. Thuật toán Prim: còn được gọi là phương pháp lân cận gần nhất.

1. V_T:={v*}, trong đó v* là đỉnh tuỳ ý của đồ thị G. E_T:=∅.
2. Với mỗi đỉnh $v_j \notin V_T$, tìm đỉnh $w_j \in V_T$ sao cho

$m(w_j,v_j) = \min m(x_i, v_j) =: \beta_j$; $x_i \in V_T$ và gán cho đỉnh v_j nhãn $[w_j, \beta_j]$. Nếu không tìm được w_j (tức là khi v_j không kề với bất cứ đỉnh nào trong V_T) thì gán cho v_j nhãn [0, ∞].

3. Chọn đỉnh v_{j^*} sao cho $\beta_{j^*} = \min \beta_j$; $v_j \notin V_T$

$V_T := V_T \cup \{v_{j^*}\}$, $E_T := E_T \cup \{(w_{j^*}, v_{j^*})\}$.

Nếu $|V_T| = n$ thì thuật toán dừng và (V_T, E_T) là cây khung nhỏ nhất.

Nếu $|V_T| < n$ thì chuyển sang Bước 4.

4. Đối với tất cả các đỉnh $v_j \notin V_T$ mà kề với v_{j^*}, ta thay đổi nhãn của chúng như sau:

Nếu $\beta_j > m(v_{j^*}, v_j)$ thì đặt $\beta_j := m(v_{j^*}, v_j)$ và nhãn của v_j là $[v_{j^*}, \beta_j]$. Ngược lại, ta giữ nguyên nhãn của v_j. Sau đó quay lại Bước 3.

CHƯƠNG IV CÂY VÀ CÂY KHUNG CỦA ĐỒ THỊ

4. Thuật toán Prim:

Xét thí dụ: Tìm cây khung nhỏ nhất cho bởi đồ thị:

	A	B	C	D	E	F	H	I
A	∞	15	16	19	23	20	32	18
B	15	∞	33	13	34	19	20	12
C	16	33	∞	13	29	21	20	19
D	19	13	13	∞	22	30	21	11
E	23	34	29	22	∞	34	23	21
F	20	19	21	30	34	∞	17	18
H	32	20	20	21	23	17	∞	14
I	18	12	19	11	21	18	14	∞

CHƯƠNG IV CÂY VÀ CÂY KHUNG CỦA ĐỒ THỊ

4. Thuật toán Prim:

Bảng nhãn của các đỉnh:

V.lặp	A	B	C	D	E	F	H	I	V_T	E_T
K.tạo	–	[A,15]	[A,16]	[A,19]	[A,23]	[A,20]	[A,32]	[A,18]	A	∅
1	–	–	[A,16]	[B,13]	[A,23]	[B,19]	[B,20]	[B,12]	A, B	(A,B)
2	–	–	[A,16]	[I,11]	[I,21]	[I,18]	[I,14]	–	A, B, I	(A,B), (B,I)
3	–	–	[D,13]	–	[I,21]	[I,18]	[I,14]	–	A, B, I, D	(A,B), (B,I), (I,D)
4	–	–	–	–	[I,21]	[I,18]	[I,14]	–	A, B, I, D, C	(A,B), (B,I), (I,D), (D,C)
5	–	–	–	–	[I,21]	[H,17]	–	–	A, B, I, D, C, H	(A,B), (B,I), (I,D), (D,C), (I,H)
6	–	–	–	–	[I,21]	–	–	–	A, B, I, D, C, H, F	(A,B), (B,I), (I,D), (D,C), (I,H), (H,F)
7	–	–	–	–	–	–	–	–	A, B, I, D, C, H, F, E	(A,B), (B,I), (I,D), (D,C), (I,H), (H,F), (I,E)

Vậy độ dài cây khung nhỏ nhất là:

15 + 12 + 11 + 13 + 14 + 17 + 21 = 103.

CHƯƠNG V CÂY VÀ CÂY KHUNG CỦA ĐỒ THỊ

5. Cây có gốc:

Định nghĩa: Cây có hướng là đồ thị có hướng mà đồ thị vô hướng nền của nó là một cây.

Cây có gốc là một cây có hướng, trong đó có một đỉnh đặc biệt, gọi là gốc, từ gốc có đường đi đến mọi đỉnh khác của cây.

Trong cây có gốc thì gốc r có bán bậc vào bằng 0, còn tất cả các đỉnh khác đều có bậc vào bằng 1.

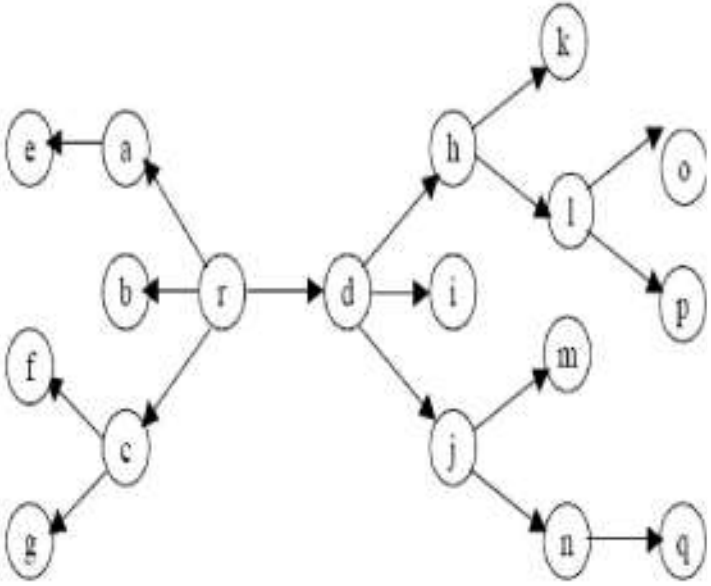
Một cây có gốc thường được vẽ với gốc r ở trên cùng và cây phát triển từ trên xuống, gốc r gọi là đỉnh mức 0. Các đỉnh kề với r được xếp ở phía dưới và gọi là đỉnh mức 1. Đỉnh ngay dưới đỉnh mức 1 là đỉnh mức 2, ...

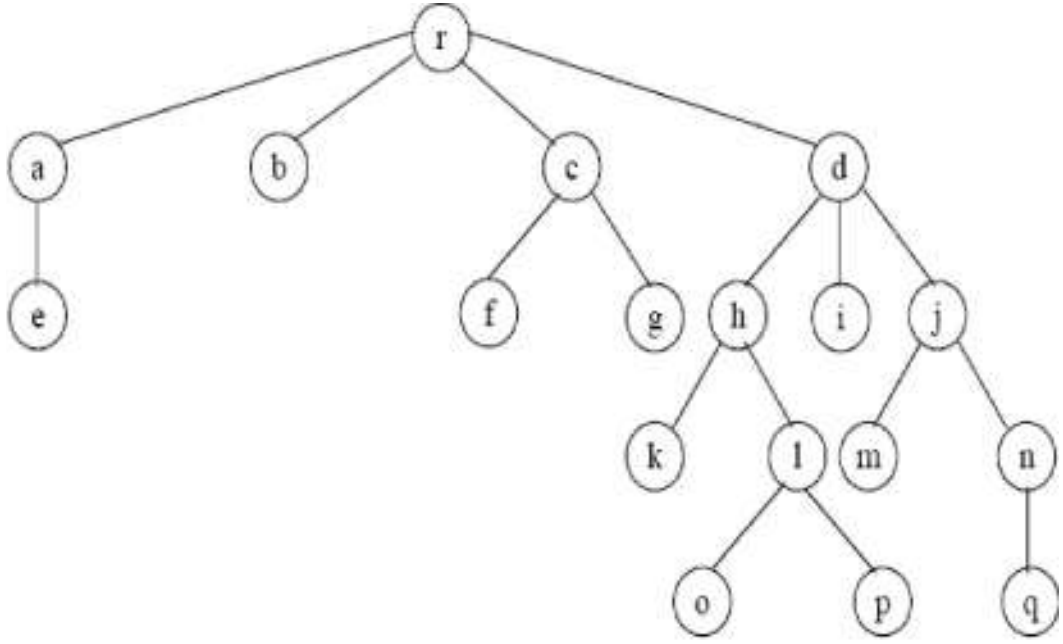
CHƯƠNG IV CÂY VÀ CÂY KHUNG CỦA ĐỒ THỊ

5. Cây có gốc:

Tổng quát, trong một cây có gốc thì v là đỉnh mức k khi và chỉ khi đường đi từ r đến v có độ dài bằng k.

Mức lớn nhất của một đỉnh bất kỳ trong cây gọi là chiều cao của cây.





CHƯƠNG IV CÂY VÀ CÂY KHUNG CỦA ĐỒ THỊ

5. Cây có gốc:

Định nghĩa: Cho cây T có gốc $r=v_0$. Giả sử $v_0, v_1, \ldots, v_{n-1}, v_n$ là một đường đi trong T. Ta gọi:

- v_{i+1} là con của v_i và v_i là cha của v_{i+1}.
- $v_0, v_1, \ldots, v_{n-1}$ là các tổ tiên của v_n và v_n là dòng dõi của $v_0, v_1, \ldots, v_{n-1}$.
- Đỉnh treo v_n là đỉnh không có con; đỉnh treo cũng gọi là lá hay đỉnh ngoài; một đỉnh không phải lá là một đỉnh trong.

Định nghĩa: Một cây có gốc T được gọi là cây m-phân nếu mỗi đỉnh của T có nhiều nhất là m con. Với m=2, ta có một cây nhị phân.

Trong một cây nhị phân, mỗi con được chỉ rõ là con bên trái hay con bên phải;

Cây có gốc T được gọi là một cây m-phân đầy đủ nếu mỗi đỉnh trong của T đều có m con.

CHƯƠNG V. TÌM ĐƯỜNG ĐI NGẮN NHẤT

- **Trong phần này, giới thiệu về giải thuật tìm đường đi ngắn nhất giữa 2 đỉnh trên đồ thị có trọng số.**
- **Nội dung gồm:**
  - 5.1. Giới thiệu về bài toán
  - 5.2. Thuật toán gán nhãn.
  - 5.3. Thuật toán Dijkstra

CHƯƠNG V. TÌM ĐƯỜNG ĐI NGẮN NHẤT

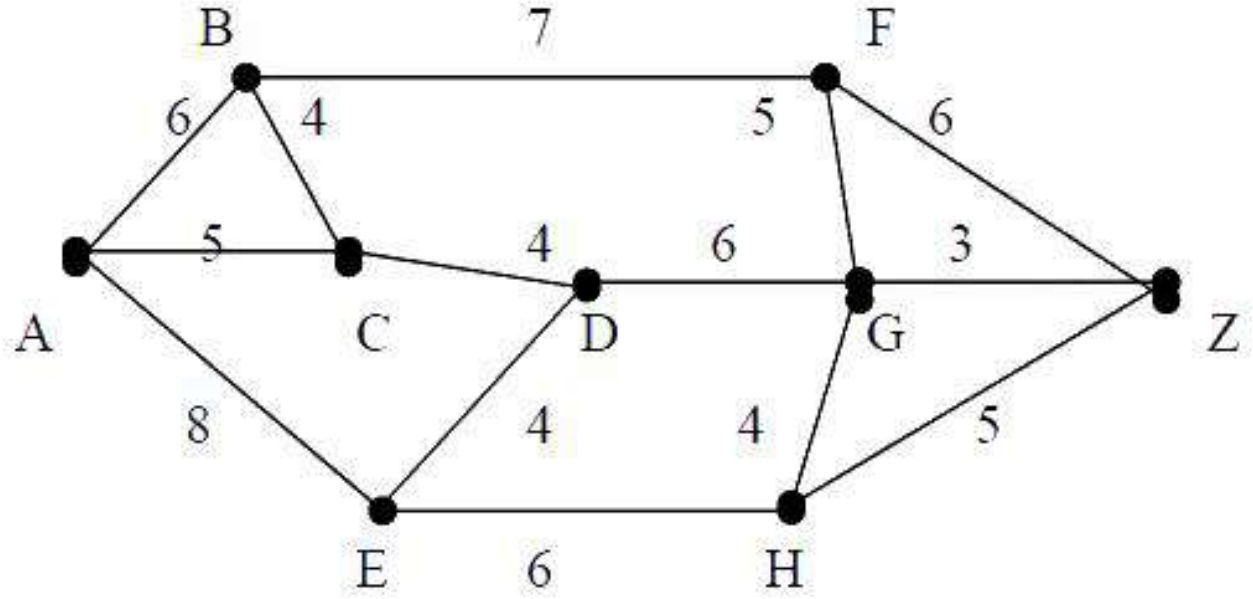
- **5.1. Giới thiệu về bài toán**
  - Xét đồ thị $G =< V, E >$ với $|V| = n, |E| = m$.
  - Với mỗi cạnh $(u, v) \in E$, có một giá trị trọng số $A[u, v]$.
  - Đặt $A[u, v] = \infty$ nếu $(u, v) \notin E$.
  - Nếu dãy v0, v1,..., vk là một đường đi trên G thì
  - $\sum_{i=1}^{k} A[v_{i-1}, v_i]$
  - được gọi là độ dài của đường đi.
  - Bài toán: Tìm đường đi ngắn nhất từ đỉnh s đến đỉnh t của đồ thị G.

CHƯƠNG V. TÌM ĐƯỜNG ĐI NGẮN NHẤT

- **5.2. Thuật toán gán nhãn (1/4)**
  - Thuật toán được mô tả như sau:
  - Từ ma trận trọng số A[u,v], u,v∈V, tìm cận trên d[v] của khoảng cách từ s đến tất cả các đỉnh v∈V.
  - Nếu thấy d[u] + A[u,v] < d[v] thì d[v] = d[u] + A[u, v] (làm tốt lên giá trị của d[v])
  - Quá trình sẽ kết thúc khi không thể làm "tốt lên" được nữa.
  - Khi đó d[v] sẽ cho ta giá trị ngắn nhất từ đỉnh s đến đỉnh v.
  - Giá trị d[v] được gọi là nhãn của đỉnh v.

CHƯƠNG V. TÌM ĐƯỜNG ĐI NGẮN NHẤT

- **5.2. Thuật toán gán nhãn (2/4)**
  - Ví dụ về thuật toán:
  - Tìm đường đi ngắn nhất từ A đến Z trong đồ thị G sau.



CHƯƠNG V. TÌM ĐƯỜNG ĐI NGẮN NHẤT

- **5.2. Thuật toán gán nhãn (3/4)**
  - Các bước thực hiện của giải thuật:
  - Bước 1: Gán cho nhãn đỉnh A là 0, $\mathrm{d}[\mathrm{A}] = 0$;
  - Bước 2: Chọn cạnh có độ dài nhỏ nhất xuất phát từ A (cạnh AC), gán nhãn của đỉnh kề C với:

$$\mathrm{d}[\mathrm{C}] = \mathrm{d}[\mathrm{A}] + \mathrm{A}[\mathrm{A}, \mathrm{C}] = 0 + 5 = 5$$

CHƯƠNG V. TÌM ĐƯỜNG ĐI NGẮN NHẤT

- **5.2. Thuật toán gán nhãn (3/4)**
  - Bước 3: Tiếp đó, trong số các cạnh đi từ một đỉnh có nhãn là A hoặc C tới một đỉnh chưa được gán nhãn, chọn cạnh sao cho: nhãn của đỉnh + với trọng số cạnh tương ứng = nhỏ nhất gán cho nhãn của đỉnh cuối của cạnh
    - Như vậy, ta lần lượt gán được các nhãn như sau:
    - d[B] = 6 vì d[B] <d[C] + A[C,B] = 5 + 4;
    - d[E] = 8;
    - Tiếp tục làm như vậy cho tới khi đỉnh Z. Nhãn của Z là độ dài đường đi ngắn nhất từ A đến Z.

CHƯƠNG V. TÌM ĐƯỜNG ĐI NGẮN NHẤT

- **5.2. Thuật toán gán nhãn (4/4)**
  - Các bước được mô tả như trong bảng sau:

Bước	Đỉnh được gán nhãn	Nhãn các đỉnh	Đỉnh đã dùng để gán nhãn
Khởi tạo	A	0	
1	C	0 + 5 = 5	A
2	B	0 + 6 = 6	A
3	E	0 + 8 = 8	A
4	D	5 + 4 = 9	C
5	F	6 + 7 = 13	B
6	H	8 + 6 = 14	E
7	G	9 + 6 = 15	D
8	Z	15 + 3 = 18	Z

  - Như vậy, độ dài đường đi ngắn nhất từ A đến Z là 18.
  - Đường đi ngắn nhất từ A đến Z qua các đỉnh: A → C → D → G → Z

CHƯƠNG V. TÌM ĐƯỜNG ĐI NGẮN NHẤT

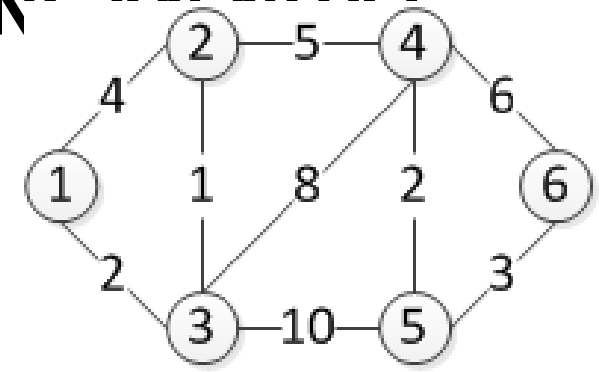
- **5.3. Thuật toán Dijkstra (1/6)**
  - Thuật toán này do E.Dijkstra, nhà toán học người Hà Lan, đề xuất năm 1959.
  - Thuật toán tìm đường đi ngắn nhất từ đỉnh s đến các đỉnh còn lại được Dijkstra đề nghị áp dụng cho trường hợp đồ thị có hướng với trọng số không âm.
  - Thuật toán được thực hiện trên cơ sở gán tạm thời cho các đỉnh.
  - Nhãn của mỗi đỉnh cho biết cận trên của độ dài đường đi ngắn nhất tới đỉnh đó.
  - Các nhãn này sẽ được biến đổi (tính lại) nhờ một thủ tục lặp, mà ở mỗi bước lặp một số đỉnh sẽ có nhãn không thay đổi, nhãn đó chính là độ dài đường đi ngắn nhất từ s đến đỉnh đó.

CHƯƠNG V. TÌM ĐƯỜNG ĐI NGẮN NHẤT

- **5.3. Thuật toán Dijkstra (2/6)**
  - Giả mã của giải thuật Dijkstra:

```
void Dijkstra(void)
/*Đầu vào G=(V, E) với n đỉnh có ma trận trọng số A[u,v]≥ 0; s∈V */
/*Đầu ra là khoảng cách nhỏ nhất từ s đến các đỉnh còn lại d[v]: v∈V*/
/*Truoc[v]: ghi lại đỉnh trước v trong đường đi ngắn nhất từ s đến v*/
{
/*Bước 1: Khởi tạo nhãn tạm thời cho các đỉnh*/
  for ( v∈ V ) {
    d[v] = A[s,v];
    truoc[v]=s;
  }
  d[s]=0;
  T = V\{s};
/*T là tập đỉnh có nhãn tạm thời*/
/* Bước lặp */
  while (T != ∅ ) {
    Tìm đỉnh u ∈ T sao cho
      d[u] = min { d[z] | z∈T};
    T= T\{u};
    /*cố định nhãn đỉnh u*/
    for ( v ∈ T ) {
    /*Gán lại nhãn cho các đỉnh trong T*/
      if (d[v] > d[u] + A[u,v]) {
        d[v] = d[u] + A[u,v];
        truoc[v] =u;
      }
    }
  }
}
```

CHƯƠNG V. TÌM ĐƯỜNG ĐI NGẮN NHẤT



- **5.3 Thuật toán Dijkstra (3/6)**
  - Ví dụ về giải thuật Dijkstra:
  - Cho đồ thị G như trên, tìm đường từ 1->6
  - Các bước thực hiện

Bước lặp	Đỉnh 1	Đỉnh 2	Đỉnh 3	Đỉnh 4	Đỉnh 5	Đỉnh 6
Khởi tạo	0,1	4,1	2,1 (∗)	∞,1	∞,1	∞,1
1	-	3,3 (∗)	-	10,3	12,3	∞,1
2	-	-	-	8,2 (∗)	12,3	∞,1
3	-	-	-	-	10,4 (∗)	14,4
4	-	-	-	-	-	13,5 (∗)
5	-	-	-	-	-	-

CHƯƠNG V. TÌM ĐƯỜNG ĐI NGẮN NHẤT

- **5.3. Thuật toán Dijkstra (4/6)**
  - Chương trình minh họa giải thuật Dijkstra:

```
#include "conio.h"
#include "io.h"
#include "iostream"
using namespace std;
#define MAX 100
#define TRUE 1
#define FALSE 0
int n, s, z;
char chon;
int truoc[MAX],d[MAX],G[MAX][MAX];
int final[MAX];
void Init(void){
  FILE * fp;
  int i, j;
  fp = fopen("dothi.in","r");
  fscanf(fp,"%d", &n);
  cout<<"\n So dinh:"<<n;
  cout<<"\n Ma tran trong so:";
  for(i=1;i<=n;i++){
    cout<<"\n";
    for(j=1;j<=n;j++){
      fscanf(fp,"%d",&G[i][j]);
      cout<<"   "<<G[i][j];
      if(G[i][j]==0)
        G[i][j]=32000;
    }
  }
  fclose(fp);
}
void Result(void){
  int i,j;
  cout<<"\n Duong di ngan nhat tu
"<<s<<" den "<<z<<" la\n";
  cout<<" <="<<z;
  i=truoc[z];
```

CHƯƠNG V. TÌM ĐƯỜNG ĐI NGẮN NHẤT

- **5.3. Thuật toán Dijkstra (5/6)**
  - Chương trình minh họa giải thuật Dijkstra:

```
  while(i!=s){
    cout<<" <="<<i;
    i=truoc[i]; }
  cout<<" <="<<s;
  cout<<"\n Do dai duong di la:
"<<d[z]; }
void Dijkstra(void){
  int v, u, minp;
  cout<<"\n Tim duong di tu s= ";
  cin>>s;
  cout<<" den z=";
  cin>>z;
  for(v=1;v<=n; v++){
    d[v]=G[s][v];
    truoc[v]=s;
    final[v]=FALSE;
  }
  truoc[s]=0;
  d[s]=0;
final[s]=TRUE;
  while(!final[z]) {
    minp=32000;
    for(v=1; v<=n; v++){
      if((!final[v]) &&
          (minp>d[v]) ){
        u=v;
        minp=d[v];}
    }
    final[u]=TRUE;
    if(!final[z]){
      for(v=1; v<=n; v++){
        if ((!final[v]) &&
          (d[u]+ G[u][v]< d[v])){
            d[v]=d[u]+G[u][v];
            truoc[v]=u;}
      }
    }
  }
}
```

CHƯƠNG V. TÌM ĐƯỜNG ĐI NGẮN NHẤT

- **5.3. Thuật toán Dijkstra (6/6)**
  - Chương trình minh họa giải thuật Dijkstra:

```
void main(void)
{
        Init();
        Dijkstra();
        Result();
        getch();
}
```

CHƯƠNG VI BÀI TOÁN LUỒNG CỰC ĐẠI

Luồng vận tải:

- **Định nghĩa:** Mạng vận tải là một đồ thị có hướng, không có khuyên và có trọng số G=(V,E) với V={v_0, v_1, ..., v_n} thoả mãn:

1)Mỗi cung e$\in$E có trọng số m(e) là một số nguyên không âm và được gọi là khả năng thông qua của cung e.

2) Có một và chỉ một đỉnh v_0 không có cung đi vào, tức là $deg^-(v_0)=0$. Đỉnh v_0 được gọi là lối vào hay đỉnh phát của mạng.

3) Có một và chỉ một đỉnh v_n không có cung đi ra, tức là $deg^+(v_n)=0$. Đỉnh v_n được gọi là lối ra hay đỉnh thu của mạng.

CHƯƠNG VI BÀI TOÁN LUỒNG CỰC ĐẠI

Luồng vận tải:

- **Định nghĩa:** Để xác định lượng vật chất chuyển qua mạng vận tải G=(V,E), người ta đưa ra khái niệm luồng vận tải và được định nghĩa như sau:

Hàm ϕ xác định trên tập cung E và nhận giá trị nguyên được gọi là luồng vận tải của mạng vận tải G nếu ϕ thoả mãn:

1) $\phi(e) \geq 0, \forall e \in E.$

2) $\sum_{e \in \Gamma^-(v)} \varphi(e) = \sum_{e \in \Gamma^+(v)} \varphi(e)$

$\forall v \in V; v \neq v_0; v \neq v_n$

$\Gamma^-(v) = \{e \in E;$ e có đỉnh cuối là v$\}$

$\Gamma^+(v) = \{e \in E;$ e có đỉnh đầu là v$\}$

CHƯƠNG VI BÀI TOÁN LUỒNG CỰC ĐẠI

Luồng vận tải:

3) $\phi(e) \le m(e), \forall e \in E$.

Ta xem $\phi(e)$ như là lượng hàng chuyển trên cung e=(u,v) từ đỉnh u đến đỉnh v và không vượt quá khả năng thông qua của cung này.

4) $$\sum_{e \in \Gamma^+(v_n)} \varphi(e) = \sum_{e \in \Gamma^-(v_n)} \varphi(e) =: \varphi(v_n)$$

Đại lượng ϕ_{vn} (ký hiệu là ϕ_n) được gọi là luồng qua mạng, hay cường độ luồng tại điểm v_n hay giá trị của luồng ϕ. Bài toán đặt ra ở đây là tìm ϕ để ϕ_{vn} đạt giá trị lớn nhất, tức là tìm giá trị lớn nhất của luồng.

CHƯƠNG VI BÀI TOÁN LUỒNG CỰC ĐẠI

Luồng vận tải:

- **Định nghĩa:** Cho mạng vận tải G=(V,E) và $A \subset V$.

Ký hiệu: $\Gamma^-(A)=\{(u,v)\in E \mid v\in A, u\notin A\}$,

$\Gamma^+(A)=\{(u,v)\in E \mid u\in A, v\notin A\}$.

Đối với tập cung M tuỳ ý, đại lượng $\varphi(M)=\sum_{e\in M}\varphi(e)$ được gọi là luồng của tập cung M.

- **Hệ quả:** Cho ϕ là luồng của mạng vận tải G=(V,E) và $A \subset V \setminus \{v_0,v_n\}$. Khi đó: $\phi(\Gamma^-(A))=\phi(\Gamma^+(A))$.

CHƯƠNG VI BÀI TOÁN LUỒNG CỰC ĐẠI

Cho mạng vận tải G=(V,E). Hãy tìm luồng ϕ để đạt ϕ_{vn} max trên mạng G.

- **Định nghĩa:** Cho A $\subset$ V là tập con tuỳ ý không chứa lối vào v_0 và chứa lối ra v_n. Tập $\Gamma^-(A)$ được gọi là một thiết diện của mạng vận tải G.

Đại lượng $m(\Gamma^-(A)) = \sum_{e \in \Gamma^-(A)} m(e)$

được gọi là khả năng thông qua của thiết diện $\Gamma^-(A)$

CHƯƠNG VI BÀI TOÁN LUỒNG CỰC ĐẠI

Định nghĩa: Cung e trong mạng vận tải G với luồng vận tải ϕ được gọi là cung bão hoà nếu ϕ(e)=m(e).

Luồng ϕ của mạng vận tải G được gọi là luồng đầy nếu mỗi đường đi từ v_0 đến v_n đều chứa ít nhất một cung bão hoà.

Ta thấy: nếu luồng ϕ trong mạng vận tải G chưa đầy thì nhất định tìm được đường đi α từ lối vào v_0 đến lối ra v_n không chứa cung bão hoà. Khi đó ta nâng luồng ϕ thành ϕ’ như sau:

CHƯƠNG VI BÀI TOÁN LUỒNG CỰC ĐẠI

$$\varphi'(e) = \begin{cases} \varphi(e) + 1 \; khi \; e \in \alpha, \\ \varphi(e) \qquad khi \; e \notin \alpha. \end{cases}$$

Khi đó ϕ’ cũng là một luồng, mà giá trị của nó là:

$$\phi'_n = \phi_n + 1 > \phi_n.$$

Như vậy, đối với mỗi luồng không đầy ta có thể nâng giá trị của nó và nâng cho tới khi nhận được một luồng đầy.

Thuật toán Ford-Fulkerson: Để tìm luồng cực đại của mạng vận tải G, xuất phát từ luồng tuỳ ý ϕ của G, rồi nâng luồng lên đầy, sau đó áp dụng thuật toán Ford-Fulkerson theo 3 bước:

Bước 1 (đánh dấu ở đỉnh của mạng): Lối vào v_0 được đánh dấu bằng 0.

1) Nếu đỉnh v_i đã được đánh dấu thì ta dùng chỉ số +i để đánh dấu cho mọi đỉnh y chưa được đánh dấu mà $(v_i,y)\in E$ và cung này chưa bão hoà ($\phi(v_i,y)<m(v_i,y)$).

2) Nếu đỉnh v_i đã được đánh dấu thì ta dùng chỉ số −i để đánh dấu cho mọi đỉnh z chưa được đánh dấu mà $(z,v_i)\in E$ và luồng của cung này dương ($\phi(z,v_i)>0$).

Thuật toán Ford-Fulkerson:

Nếu ta đánh dấu được tới lối ra v_n thì trong G $\exists$ giữa v_0 và v_n một xích α, mọi đỉnh đều khác nhau và được đánh dấu theo chỉ số của đỉnh liền trước nó (chỉ sai khác nhau về dấu). Khi đó chắc chắn ta nâng được giá trị của luồng.

Bước 2 (nâng giá trị của luồng): Ta đặt:

$\phi'(e) = \phi(e)$, nếu $e \notin \alpha$,

$\phi'(e) = \phi(e)+1$, nếu $e \in \alpha$ được định hướng theo chiều của xích α đi từ v_0 đến v_n,

$\phi'(e) = \phi(e)-1$, nếu $e \in \alpha$ được định hướng ngược với chiều của xích α đi từ v_0 đến v_n.

Lặp lại một vòng mới. Vì khả năng thông qua của các cung đều hữu hạn, nên quá trình phải dừng lại sau một số hữu hạn bước.

CHƯƠNG VI BÀI TOÁN LUỒNG CỰC ĐẠI

Thuật toán Ford-Fulkerson:

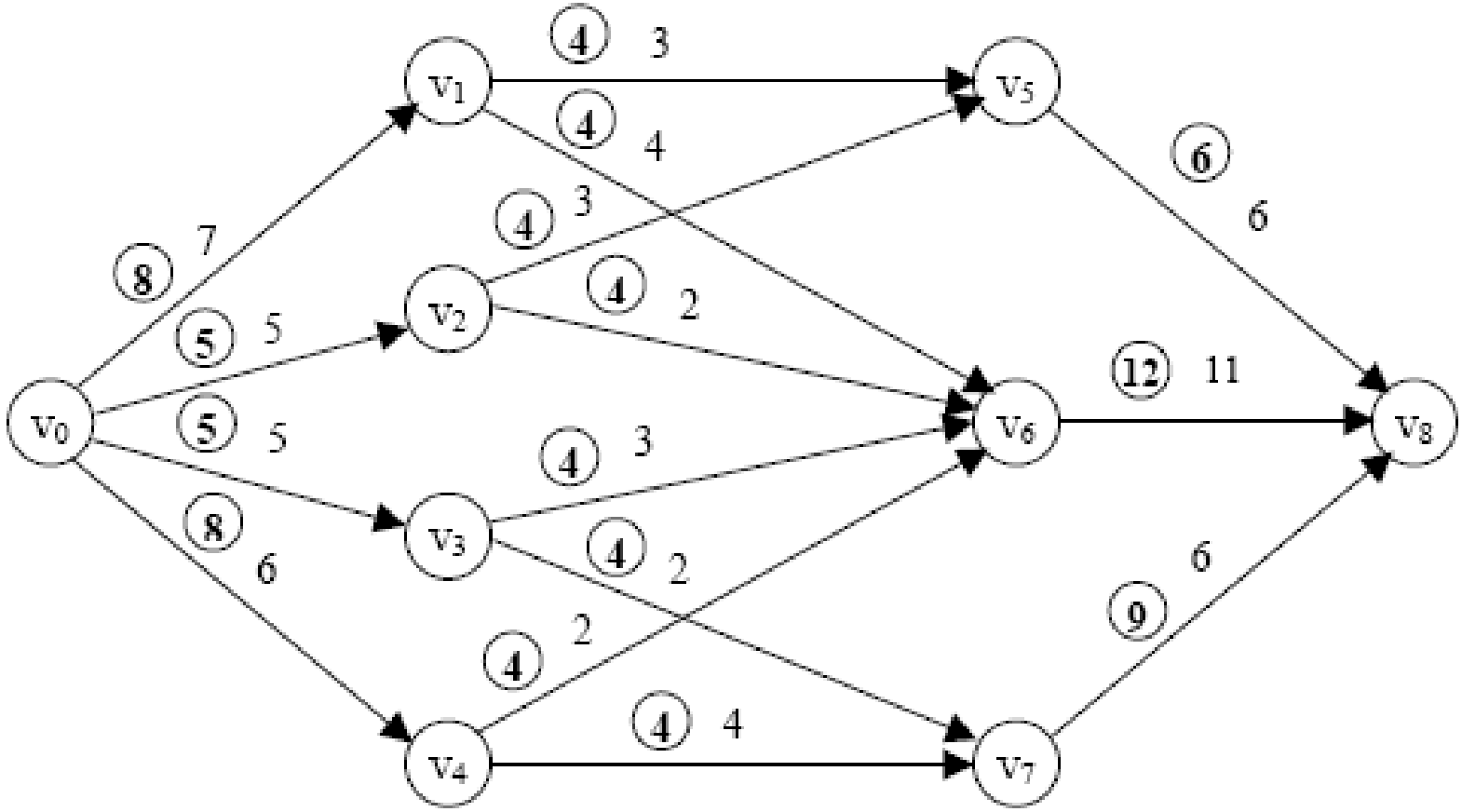
Bước 3:

Nếu với luồng ϕ^0 bằng phương pháp trên ta không thể nâng giá trị của luồng lên nữa, nghĩa là ta không thể đánh dấu được đỉnh v_n, thì ta nói rằng quá trình nâng luồng kết thúc và ϕ^0 đã đạt giá trị cực đại, đồng thời gọi ϕ^0 là luồng kết thúc.

- **Định lý (Ford-Fulkerson):** Trong mạng vận tải G=(V,E), giá trị lớn nhất của luồng bằng khả năng thông qua nhỏ nhất của thiết diện, nghĩa là:

$$\max_{\varphi} \varphi_{v_n} = \min_{A \subset V, v_0 \notin A, v_n \in A} m(\Gamma^-(A))$$

Ví dụ: Cho mạng vận tải với khả năng thông qua được đặt trong khuyên tròn, luồng được ghi trên các cung. Tìm luồng cực đại của mạng này.

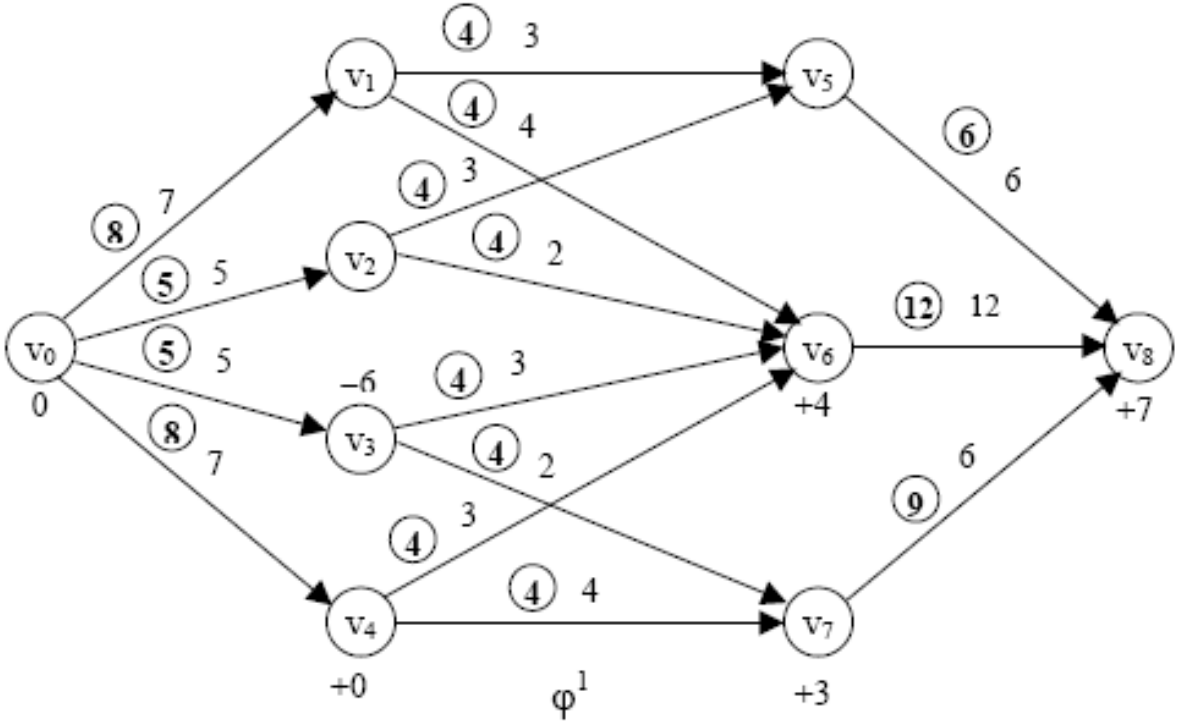


φ

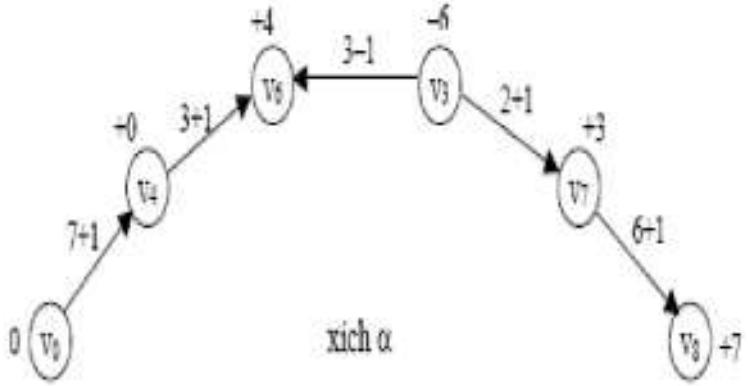
Luồng ϕ có đường đi (v_0,v_4), (v_4,v_6), (v_6,v_8) gồm các cung chưa bão hoà nên nó chưa đầy. Do đó có thể nâng luồng của các cung này lên một đơn vị, để được ϕ^1.

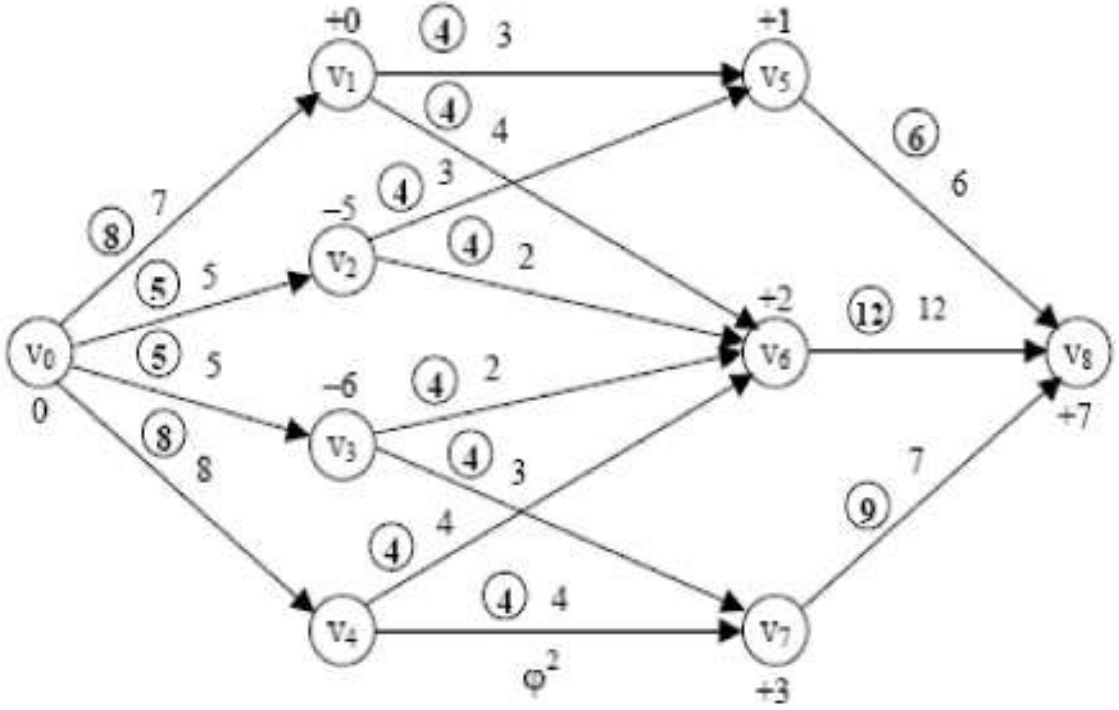
Do mỗi đường xuất phát từ v_0 đến v_8 đều chứa ít nhất một cung bão hoà, nên luồng ϕ^1 là luồng đầy. Song nó chưa phải là luồng cực đại.

Áp dụng TT Ford-Fulkerson để nâng luồng ϕ^1.

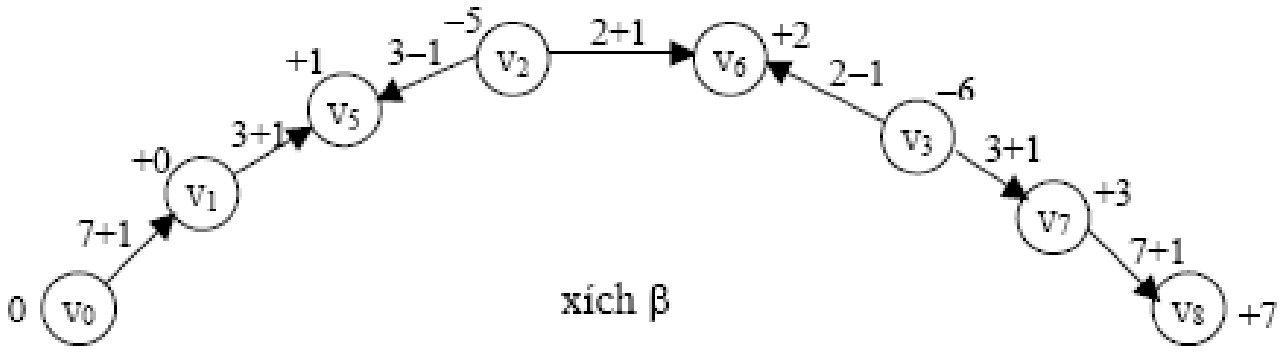


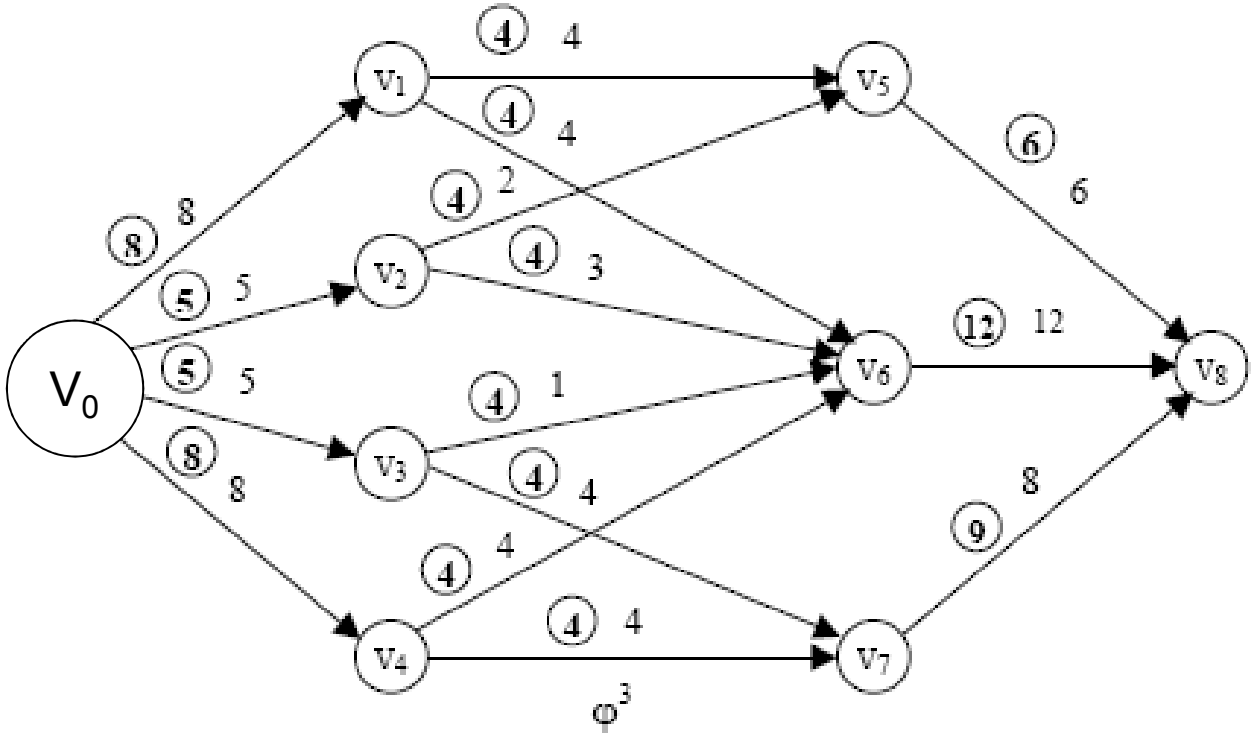
Xét xích $\alpha=(v_0, v_4, v_6, v_3, v_7, v_8)$. Quá trình đánh dấu từ v_0 đến v_8 để có thể nâng luồng ϕ^1 lên một đơn vị bằng cách biến đổi luồng tại các cung thuộc xích α được đánh dấu. Sau đó ta có luồng ϕ^2.





Tương tự, xét xích $\beta=(v_0, v_1, v_5, v_2, v_6, v_3, v_7, v_8)$. Nâng luồng ϕ^2 lên một đơn vị, sau đó ta có luồng ϕ^3.





CHƯƠNG VI BÀI TOÁN LUỒNG CỰC ĐẠI

Tiếp theo ta chỉ có thể đánh dấu được đỉnh v_0 nên quá trình nâng luồng kết thúc và ta được giá trị của luồng cực đại là:

$$\phi^3_{v8} = 6+12+8 = 26.$$

Mặt khác, thiết diện nhỏ nhất $\Gamma^-(A)$

với $A=\{v_1, v_2, ..., v_8\}$

là $\Gamma^- (A)=\{(v_0,v_1), (v_0,v_2), (v_0,v_3), (v_0,v_4)\}$.