Phần 2

LÝ THUYẾT ĐỒ THỊ

Graph Theory

GV: Nguyễn Huy Đức

Bộ môn: Khoa học Máy tính – ĐH Thủy lợi

0903 402 655   ducnghuy@gmail.com

Nội dung

- Chương 1: Các khái niệm cơ bản
- **Chương 2: Biểu diễn đồ thị**
- Chương 3: Các thuật toán tìm kiếm trên đồ thị
- Chương 4: Đồ thị Euler và đồ thị Hamilton
- Chương 5: Cây và cây khung của đồ thị
- Chương 6: Bài toán đường đi ngắn nhất

Chương 2

BIỂU DIỄN ĐỒ THỊ

Representations of Graphs

Biểu diễn đồ thị

- Có nhiều cách biểu diễn. Việc lựa chọn cách biểu diễn phụ thuộc vào từng bài toán cụ thể cần xét, thuật toán cụ thể cần cài đặt.
- Có hai vấn đề chính cần quan tâm khi lựa chọn cách biểu diễn:
  – Bộ nhớ mà cách biểu diễn đó đòi hỏi
  – Thời gian cần thiết để trả lời các truy vấn thường xuyên đối với đồ thị trong quá trình xử lý đồ thị:
    - Chẳng hạn:
      – Có cạnh nối hai đỉnh *u*, *v* ?
      – Liệt kê các đỉnh kề của đỉnh *v* ?

Biểu diễn đồ thị

Các cách biểu diễn:

- Ma trận kề, Ma trận trọng số.
- Ma trận liên thuộc đỉnh – cạnh
- Danh sách kề.
- Danh sách cạnh.

Ma trận kề

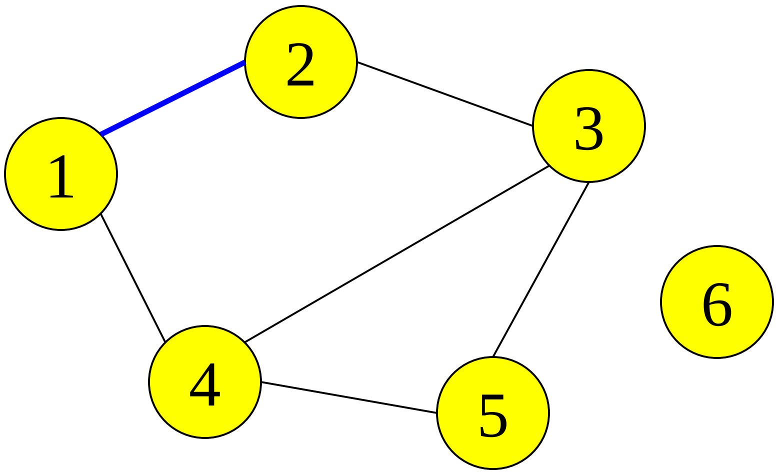
(Adjacency Matrix)

- Đồ thị *G=(V, E)*, với $V = \{1, 2, ..., n\}, E = \{e_1, e_2, ..., e_m\}$
- *Ma trận kề biểu diễn G* là ma trận nhị phân vuông cấp n $A = (a_{ij})_n$, ở đó:

$$a_{ij} = \begin{cases} 1 & \text{nếu } (i, j) \in E \\ 0 & \text{nếu trái lại} \end{cases}$$

- Có n! ma trận kề *A* do có n! cách sắp xếp n đỉnh.

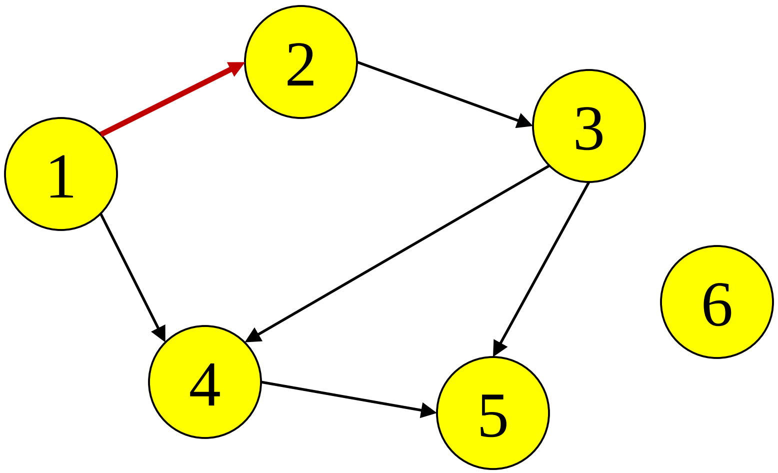
Ma trận kề của đồ thị vô hướng



$$A[u,v] = \begin{cases} 1 \text{ nếu } (u,v) \in E \\ 0 \text{ nếu trái lại} \end{cases}$$

$$\begin{array}{c|c} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} & \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Ma trận kề của đồ thị có hướng



$$\begin{array}{c c} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} & \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$A[u,v] = \begin{cases} 1 \text{ nếu } (u,v) \in E \\ 0 \text{ nếu trái lại} \end{cases}$$

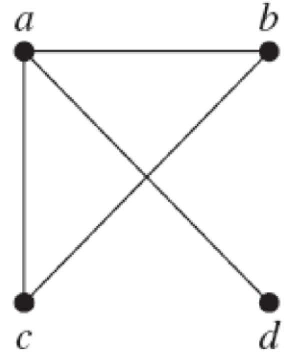
Tính chất của ma trận kề

– Gọi A là ma trận kề của đồ thị vô hướng:
  - A là ma trận đối xứng: $A = A^T$ ($a_{ij} = a_{ji}$)
  - deg(v) = Tổng các phần tử trên dòng v của A
  - Nếu ký hiệu $A^k = (a^{(k)}[u,v])$ thì $a^{(k)}[u,v]$ là số lượng đường đi từ u đến v đi qua không quá k-1 đỉnh trung gian.

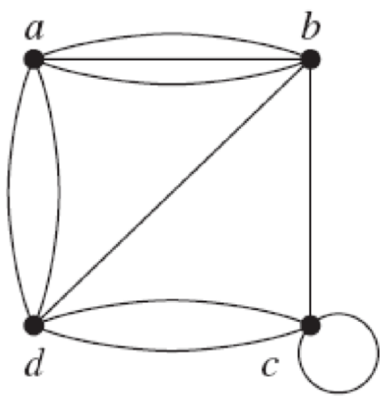
– Khái niệm ma trận kề có thể mở rộng để biểu diễn đa đồ thị vô hướng: a_{uv} – số lượng cạnh nối hai đỉnh u và v.

– Ma trận kề của đồ thị có hướng đối xứng không ?

Ví dụ



$$
\begin{array}{c}
\\
a \\ b \\ c \\ d
\end{array}
\begin{array}{c}
\begin{array}{cccc} a & b & c & d \end{array} \\
\begin{bmatrix}
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$



$$
\begin{array}{c}
\\
a \\ b \\ c \\ d
\end{array}
\begin{array}{c}
\begin{array}{cccc} a & b & c & d \end{array} \\
\begin{bmatrix}
0 & 3 & 0 & 2 \\
3 & 0 & 1 & 1 \\
0 & 1 & 1 & 2 \\
2 & 1 & 2 & 0
\end{bmatrix}
\end{array}
$$

Nhận xét

- Bộ nhớ (Space): $|V|^2$ bits
- Trả lời các truy vấn
  - Hai đỉnh i và j có kề nhau không ?

    Chỉ cần 01 phép so sánh a[i,j] =1 ?
  - Bổ sung hoặc loại bỏ cạnh: cần 1 phép gán sửa 1 phần tử của A
  - Bổ sung đỉnh: tăng kích thước ma trận kề

Ma trận trọng số

- Trong trường hợp đồ thị có trọng số trên cạnh, thay vì ma trận kề, để biểu diễn đồ thị ta sử dụng **ma trận trọng số**

$$C = c[i, j], \ i, j = 1, 2, \ldots, n,$$

với

$$c[i,j] = \begin{cases} c(i,j), & \text{nếu } (i,j) \in E \\ \theta, & \text{nếu } (i,j) \notin E, \end{cases}$$

trong đó θ là giá trị đặc biệt để chỉ ra một cặp (i,j) không là cạnh, tuỳ từng trường hợp cụ thể, có thể được đặt bằng một trong các giá trị sau: 0, $+\infty$, $-\infty$.

Ma trận liên thuộc đỉnh - cạnh

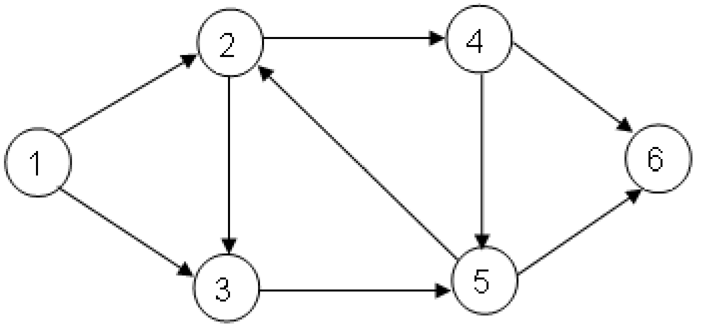
- Xét $G = (V, E)$, $(V = \{1, 2, ..., n\}, E = \{e_1, e_2, ..., e_m\})$, là đơn đồ thị có hướng.
- Ma trận liên thuộc đỉnh cạnh $A = (a_{ij}: i = 1, 2, ..., n; j = 1, 2, ..., m)$, với

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

- Ma trận liên thuộc đỉnh-cạnh là một trong những cách biểu diễn rất hay được sử dụng trong các bài toán liên quan đến đồ thị có hướng mà trong đó phải xử lý các cung của đồ thị.

Ma trận liên thuộc đỉnh - cạnh

- Ví dụ:

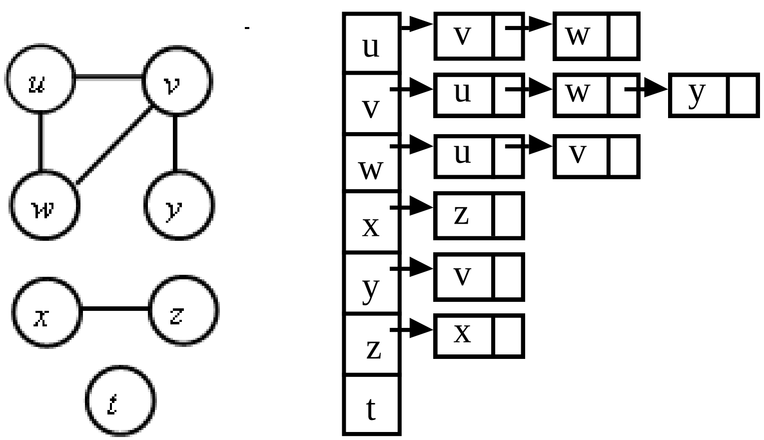


$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{ccccccccc} (1,2) & (1,3) & (2,3) & (2,4) & (3,5) & (4,5) & (4,6) & (5,2) & (5,6) \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & -1 & 0 &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

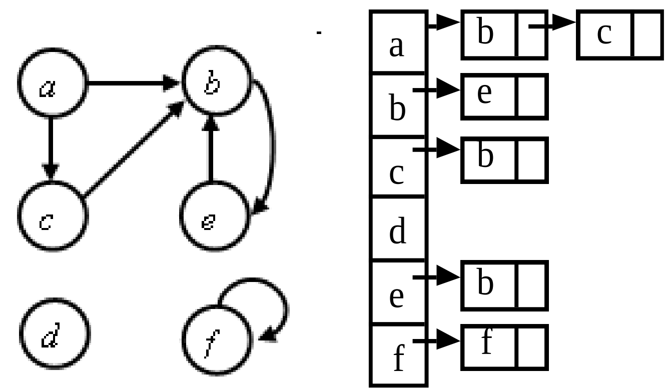
Danh sách kề

- **Danh sách kề** (Adjacency Lists): Với mỗi đỉnh *v* cất giữ danh sách các đỉnh kề của nó.
  - Là mảng *Ke* gồm |*V*| danh sách.
  - Mỗi đỉnh có một danh sách.
  - Với mỗi *u* ∈ *V*, *Ke*[*u*] bao gồm tất cả các đỉnh kề của *u*.
- **Ví dụ:**

Đồ thị vô hướng

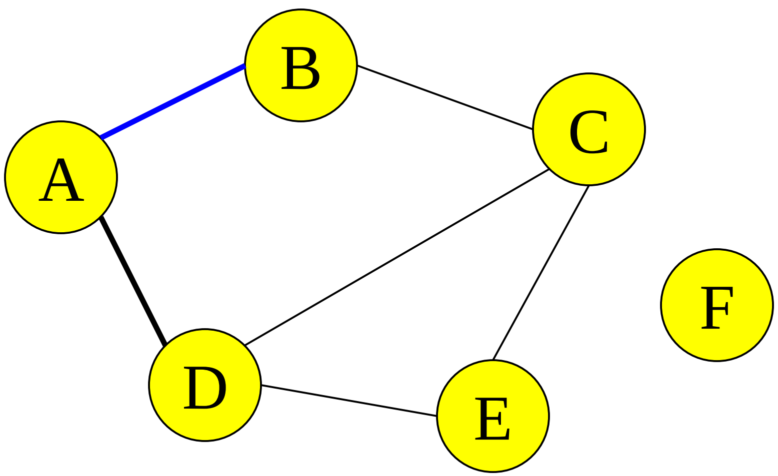


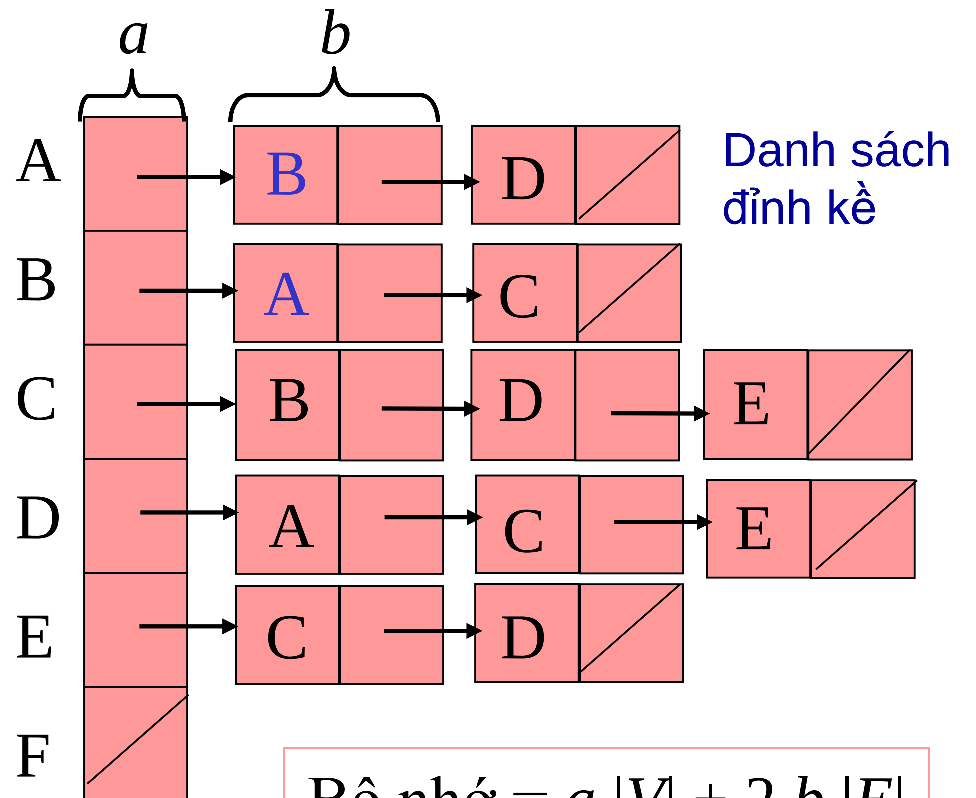
Đồ thị có hướng



Danh sách kề của đồ thị vô hướng

Với mỗi $v \in V$, Ke(v) = danh sách các đỉnh u: $(v, u) \in E$

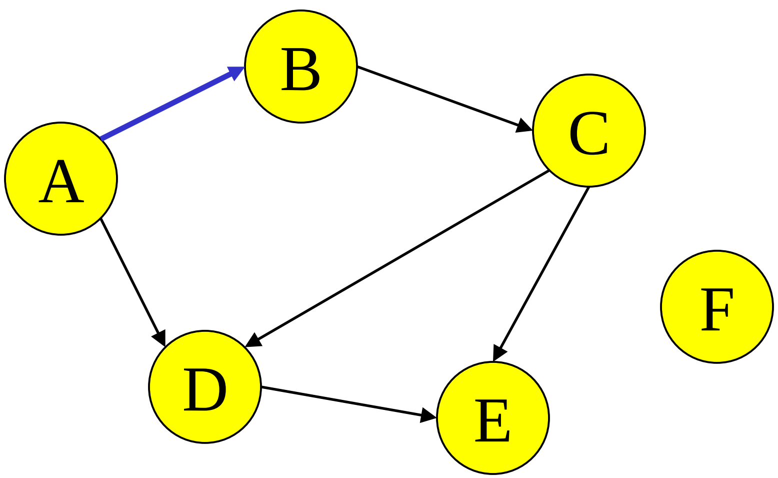


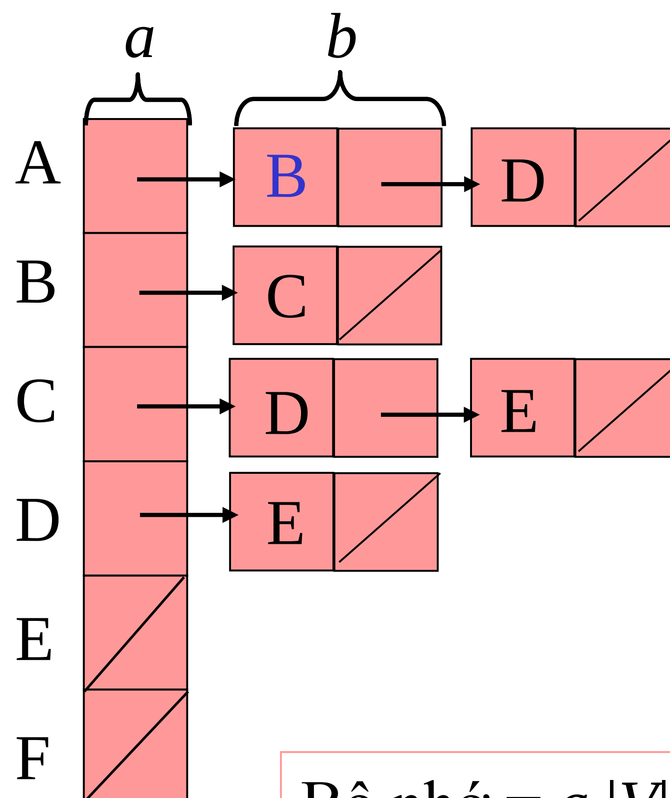


Bộ nhớ = $a\,|V| + 2\,b\,|E|$

Danh sách kề của đồ thị có hướng

Với mỗi $v \in V$, Ke(v) = $\{ u: (v, u) \in E \}$





Bộ nhớ = $a\,|V| + b\,|E|$

Nhận xét

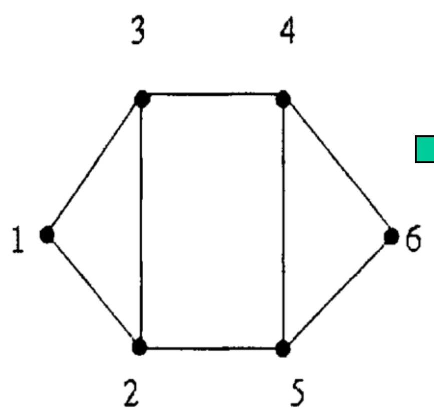
- Tổng cộng bộ nhớ: cỡ $(|V|+|E|)$ ô nhớ.
- Thường là nhỏ hơn nhiều so với $|V|^2$, nhất là đối với đồ thị thưa (sparse graph).
- Đồ thị thưa là đồ thị mà $|E| = k\,|V|$ với $k < 10$.
- ***Chú ý:***
  - ***Phần lớn các đồ thị trong thực tế ứng dụng là đồ thị thưa!***
  - ***Cách biểu diễn này được sử dụng nhiều nhất trong ứng dụng***

Nhận xét

- <u>Thời gian trả lời các truy vấn:</u>
  - Thêm cạnh — Thêm nút vào Danh sách kề
  - Xoá cạnh — Duyệt qua danh sách kề của mỗi đầu mút.
  - Thêm đỉnh — Phụ thuộc vào cài đặt.
  - Liệt kê các đỉnh kề của *v:* duyệt danh sách kề của v (tốt hơn ma trận kề)
  - Hai đỉnh *i*, *j* có kề nhau? Cần tìm kiếm trên danh sách kề, không hiệu quả bằng ma trận kề.

Danh sách cạnh (cung)

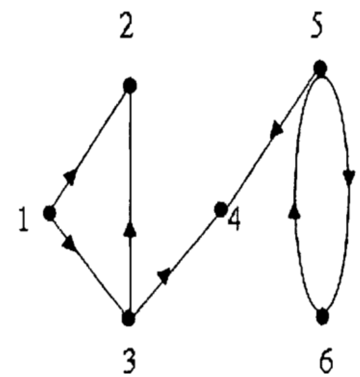
- Phù hợp với đồ thị thưa.
- Mỗi cạnh (cung) $e = (x,y)$ tương ứng với 2 biến Dau[e] và Cuoi[e].
- Để lưu trữ đồ thị cần $2m$ đơn vị bộ nhớ.



G

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

Danh sách cạnh của G



G_1

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

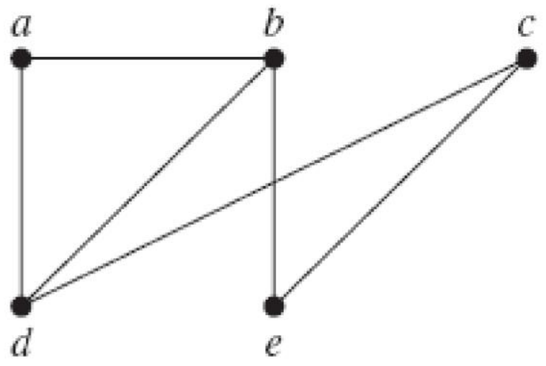
Danh sách cung của G_1

- Hai đỉnh *i*, *j* có kề nhau? Duyệt qua danh sách tất cả các cạnh của đồ thị (tồi hơn ma trận kề).

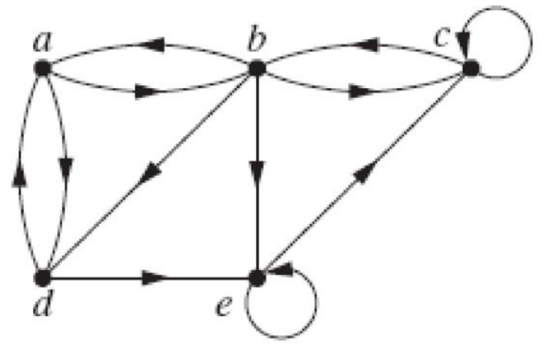
Bài tập

1. Biểu diễn đồ thị sau bằng ma trận kề:

2\.



4\.



2. Vẽ đồ thị có ma trận kề sau:

a) $\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

b) $\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$