# Biểu diễn đồ thị trên máy tính

# Biểu diễn đồ thị trên máy tính???

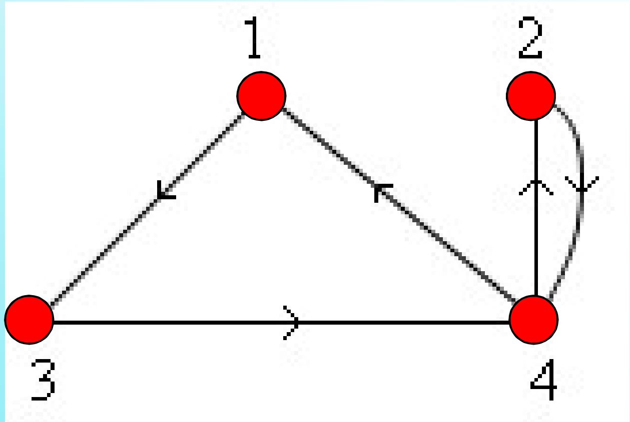
- Tại sao phải biểu diễn đồ thị trên máy tính???
  - Lý thuyết đồ thị ngày càng được ứng dụng rộng rãi.
  - Để xây dựng được các ứng dụng của đồ thị trên máy tính thì cần phải tìm cách biểu diễn đồ thị trên máy tính thích hợp.
  - Máy tính không thể hiểu được các đồ thị dưới dạng hình vẽ thông thường.
- Tiêu chuẩn để lựa chọn cách thức biểu diễn đồ thị trên máy tính?
  - Cấu trúc dữ liệu phải đơn giản, phù hợp với từng bài toán ứng dụng.
  - Dễ biểu diễn, dễ cài đặt các ứng dụng trên đó.

# Ma trận kề

- Cho đồ thị G = <V, E>, với V = {$v_1$, $v_2$, …, $v_n$}. Ma trận kề biểu diễn G là một ma trận vuông A, kích thước nxn, được xác định như sau:

$$A_{ij} = \begin{cases} 1, & (v_i, v_j) \in E \\ 0, & (v_i, v_j) \notin E \end{cases}$$

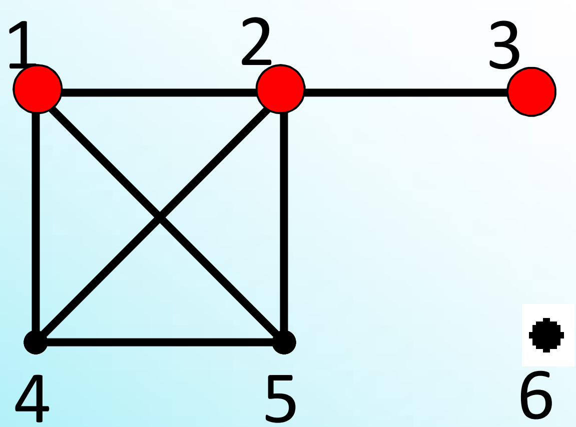
**VD:**




$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

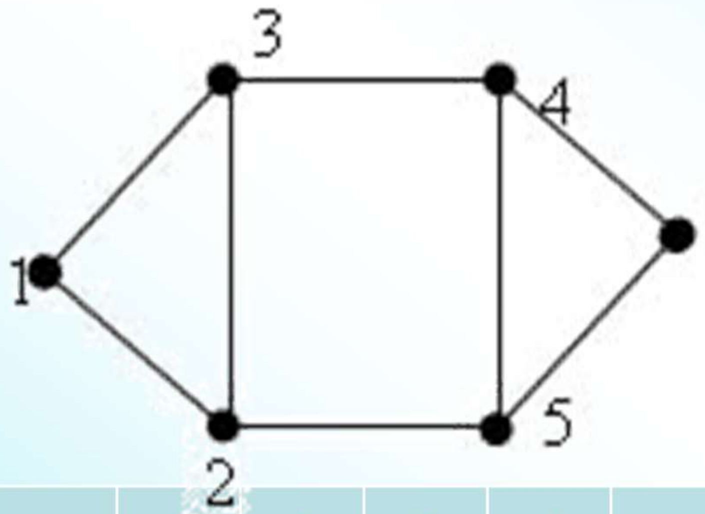
# Ma trận kề (tt)

**VD:**

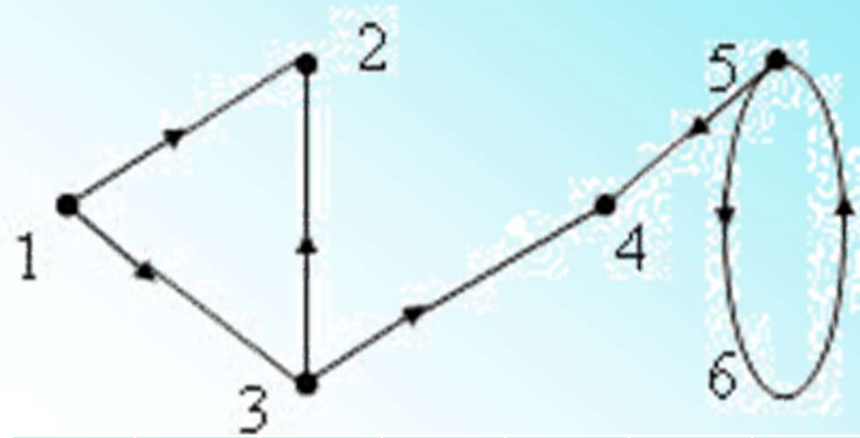



$$A = \begin{array}{c c} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} & \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 &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

# Ma trận kề (tt)




| | 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** |

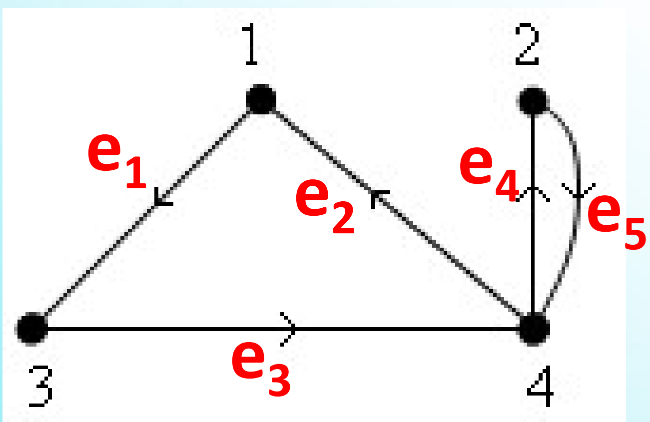



| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | **0** | **1** | **1** | **0** | **0** | **0** |
| 2 | **0** | **0** | **0** | **0** | **0** | **0** |
| 3 | **0** | **1** | **0** | **1** | **0** | **0** |
| 4 | **0** | **0** | **0** | **0** | **0** | **0** |
| 5 | **0** | **0** | **0** | **1** | **0** | **1** |
| 6 | **0** | **0** | **0** | **0** | **1** | **0** |

# Danh sách cạnh

- Cho đồ thị G = <V, E> có m cạnh. Danh sách cạnh của G sẽ bao gồm hai mảng 1 chiều có kích thước m:
  - Mảng **Dau** sẽ lưu các đỉnh đầu của các cạnh
  - Mảng **Cuoi** sẽ lưu đỉnh cuối của các cạnh

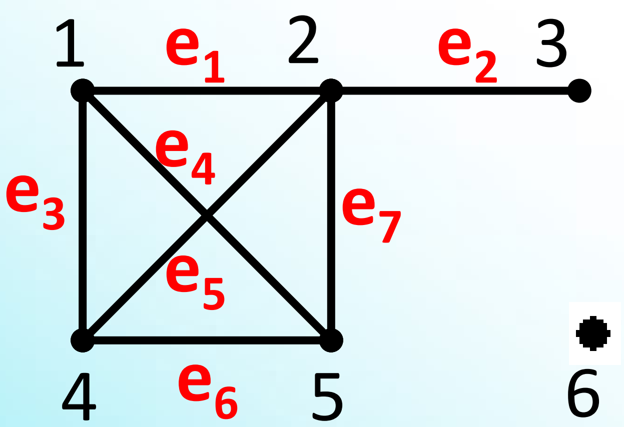
**VD:**




| Dau | Cuoi |
|---|---|
| 1 | 3 |
| 4 | 1 |
| 3 | 4 |
| 4 | 2 |
| 2 | 4 |

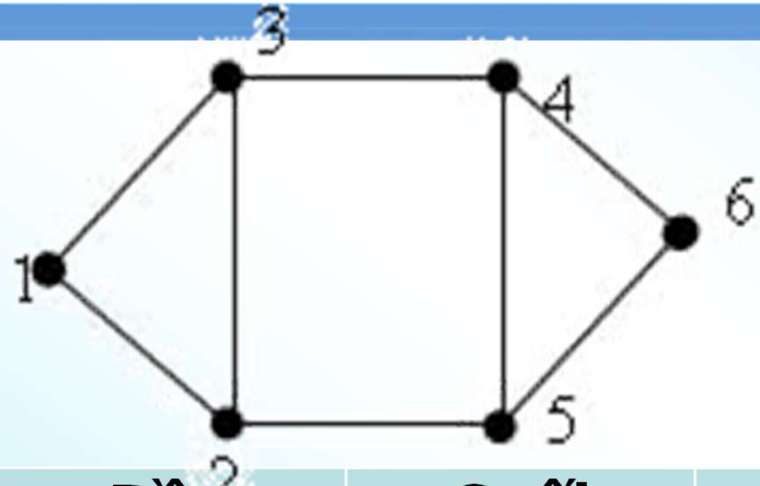
# Danh sách cạnh (tt)

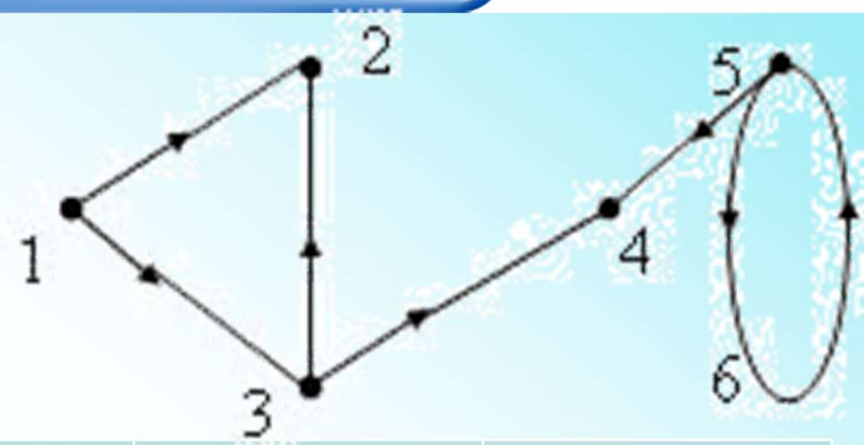
- **VD:**




| Dau | Cuoi |
|---|---|
| 1 | 2 |
| 2 | 3 |
| 1 | 4 |
| 1 | 5 |
| 4 | 2 |
| 4 | 5 |
| 2 | 5 |

# Danh sách cạnh (tt)




| Đầu | Cuối | | Đầu | Cuối |
|---|---|---|---|---|
| 1 | 2 | | 1 | 2 |
| 1 | 3 | | 1 | 3 |
| 2 | 3 | | 3 | 2 |
| 2 | 5 | | 3 | 4 |
| 3 | 4 | | 5 | 4 |
| 4 | 5 | | 5 | 6 |
| 4 | 6 | | 6 | 5 |
| 5 | 6 | | | |
| Danh sách cạnh của G | | | Danh sánh cung của $G_1$ | |

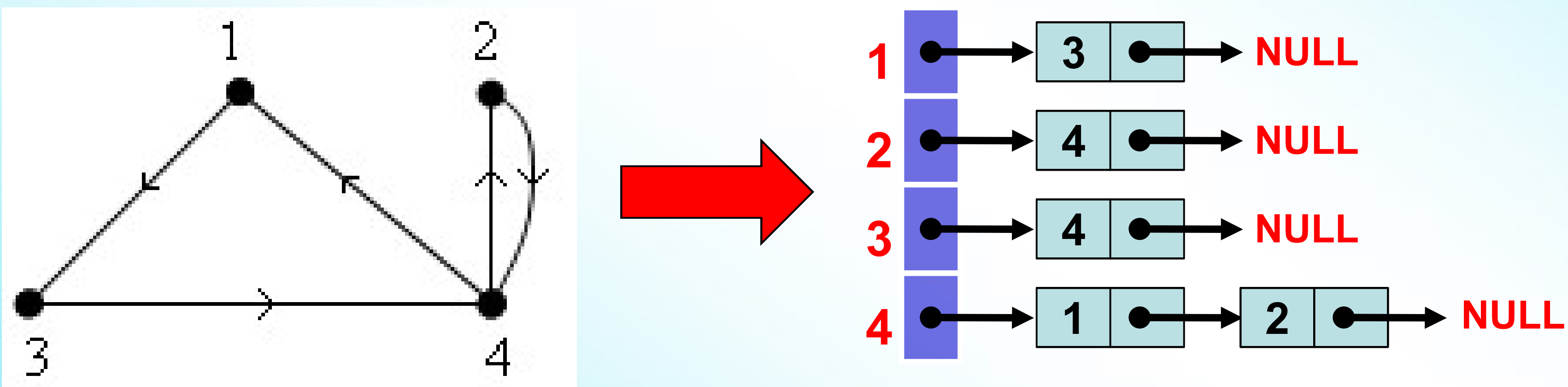
# Danh sách cạnh (tt)

- Xác định bậc của đỉnh dựa vào danh sách cạnh:
  - Đối với đồ thị vô hướng: **duyệt qua 2 mảng Dau va Cuoi**, số lần xuất hiện của một đỉnh chính là **bậc** của đỉnh đó.
  - Đối với đồ thị có hướng:
    - **Duyệt qua mảng Dau**, số lần xuất hiện của một đỉnh chính là **bán bậc ra** của đỉnh đó.
    - **Duyệt qua mảng Cuoi**, số lần xuất hiện của một đỉnh chính là **bán bậc vào** của đỉnh đó.

# Danh sách kề

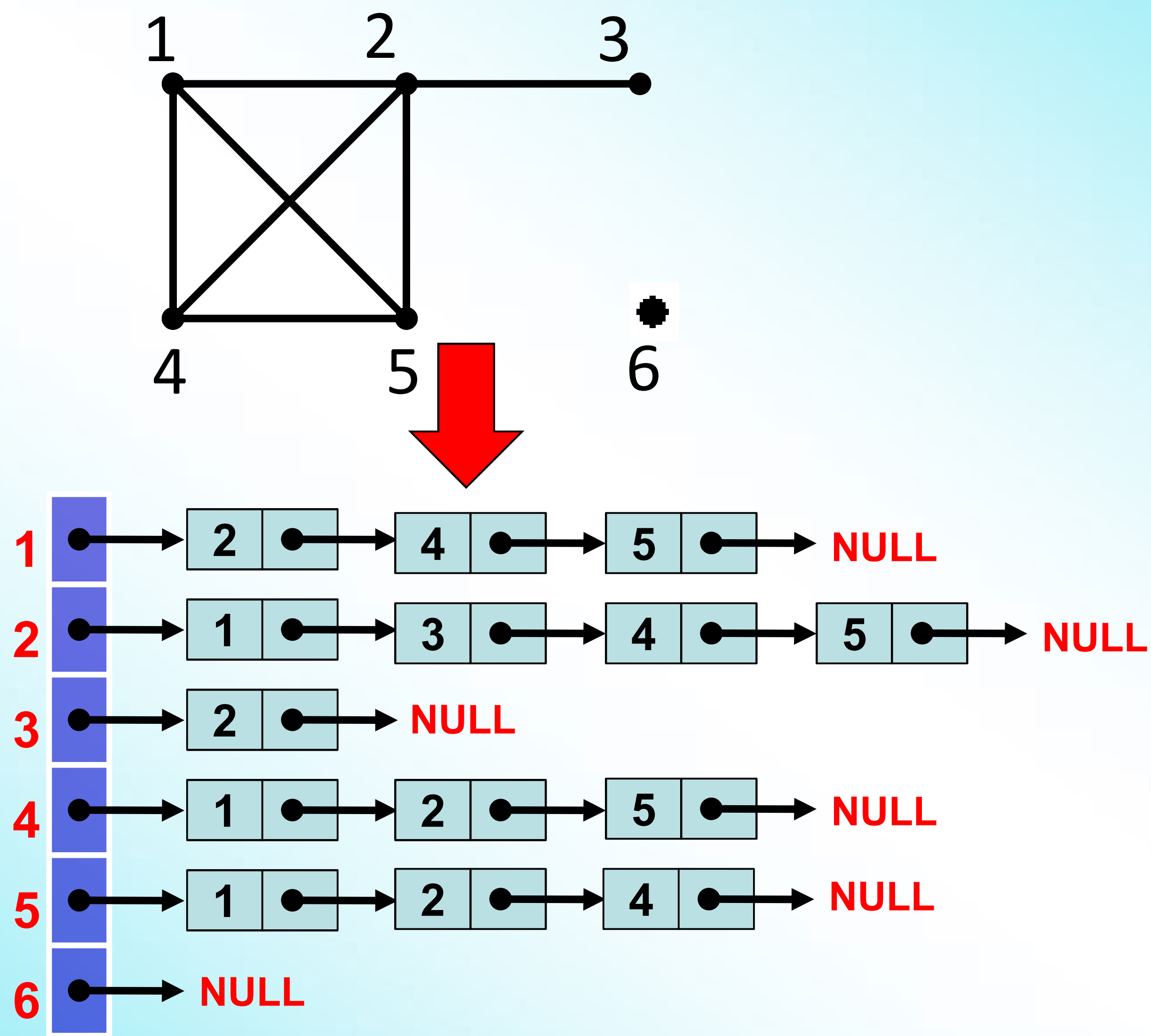
- Cho đồ thị G = <V,E> có n đỉnh. Đồ thị G có thể được biểu diễn bằng n danh sách liên kết. Mỗi danh sách liên kết thứ i sẽ biểu diễn các đỉnh kề với đỉnh $v_i$

**VD:**

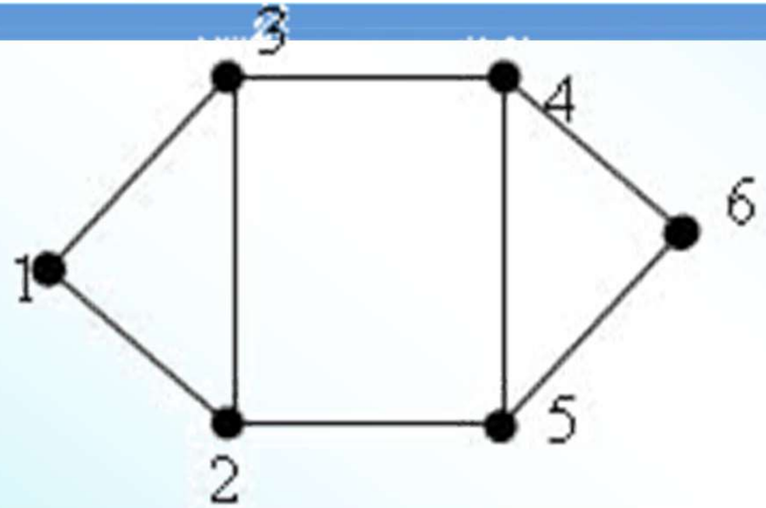


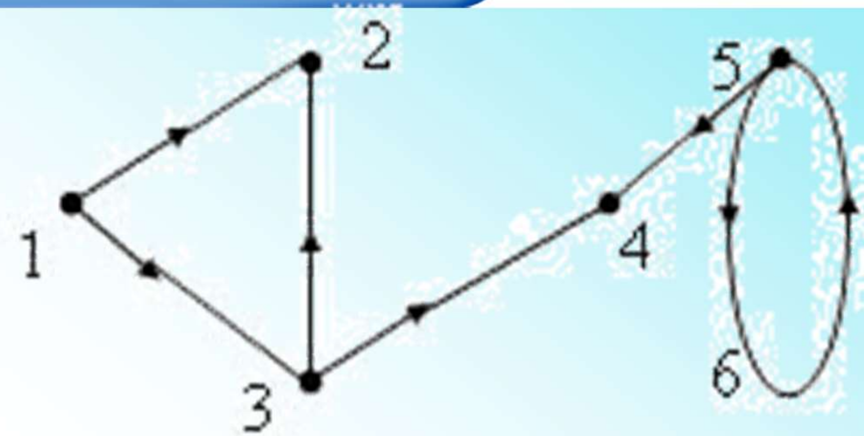
# Danh sách kề (tt)

- **VD:**



# Danh sách kề (tt)




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

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

# Danh sách kề (tt)

- Xác định bậc của đỉnh dựa vào danh sách kề:
  - Đối với đồ thị vô hướng: Số phần tử của mỗi danh sách sẽ là bậc của đỉnh tương ứng
  - Đối với đồ thị có hướng:
    - Số phần tử của mỗi danh sách sẽ là bán bậc ra của đỉnh tương ứng
    - Việc xác định bán bậc vào khó khăn hơn nhiều: phải duyệt qua tất cả các danh sách, số lần xuất hiện của 1 đỉnh trong các danh sách chính là bán bậc vào của đỉnh đó.

# Thuật ngữ Việt - Anh

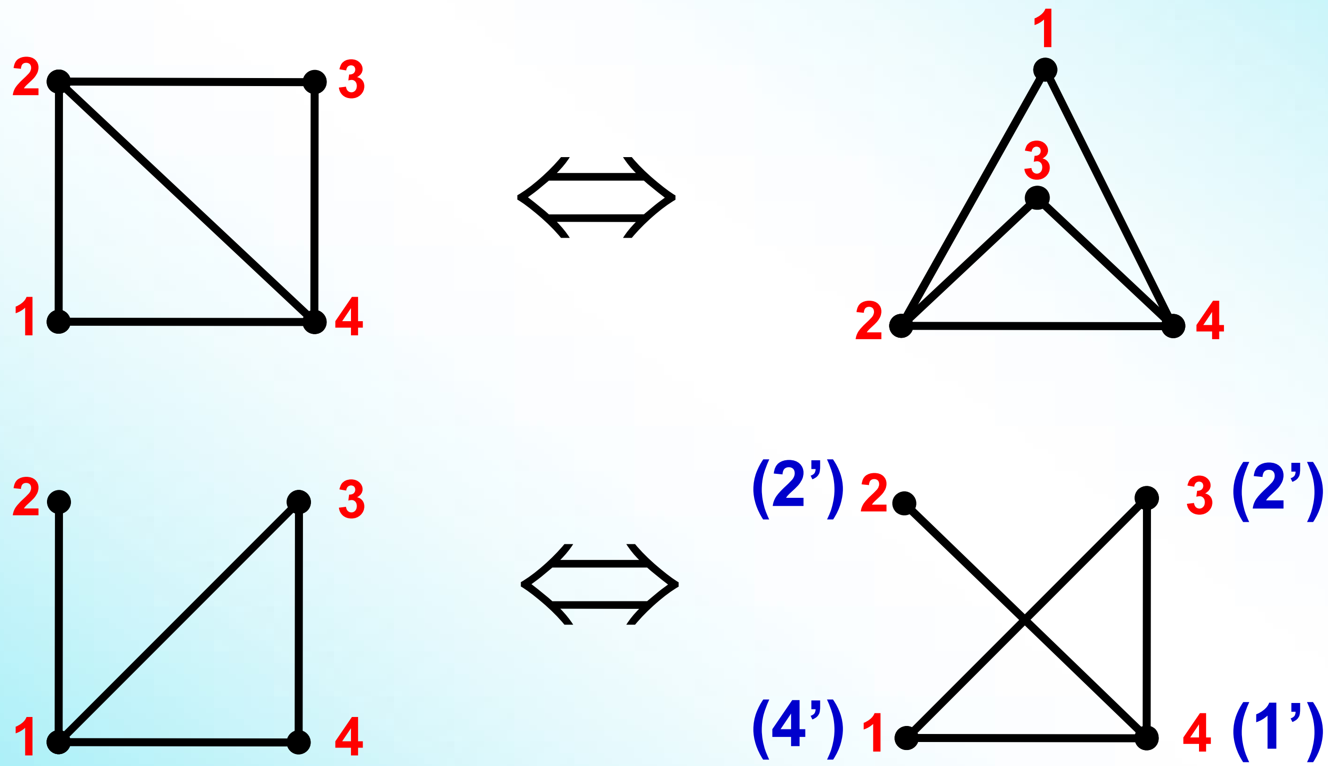
| | |
|---|---|
| Ma trận kề | Adjacent Matrix |
| Ma trận liên thuộc | Incident Matrix |
| Danh sách cạnh | Edge List |
| Danh sách kề | Adjacent List |
| Đẳng cấu | Isomorphism |

# Phần 2.2

# Sự đẳng cấu của đồ thị

# Đặt vấn đề

- Xét hai đồ thị sau: chúng giống nhau hay khác nhau???



# Sự đẳng cấu của đồ thị

- Cho 2 đồ thị G = <V,E> và đồ thị G’ = <V’,E’>. Hai đồ thị G và G’ được nói là **đẳng cấu (đẳng hình, đồng cấu)** với nhau nếu và chỉ nếu tồn tại một **song ánh**:

$$f : V \to V'$$

sao cho:

$$\forall u, v \in V, (u, v) \in E \Leftrightarrow \left( f(u), f(v) \right) \in E'$$

***(Hai đỉnh tạo thành cạnh trong G thì hai ảnh của chúng cũng tạo thành cạnh trong G’, và ngược lại)***

***Ký hiệu:*** $G \cong G'$

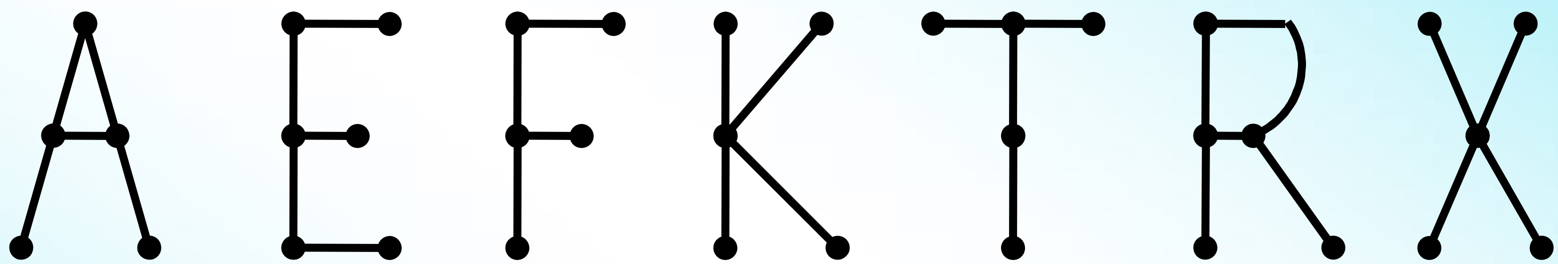
# Sự đẳng cấu của đồ thị

## *Chú ý*

- Nếu G và G' là các đơn đồ thị vô hướng đẳng cấu qua ánh xạ f thì chúng có:
  - Cùng số đỉnh
  - Cùng số cạnh
  - Cùng số đỉnh với bậc cho sẵn (vd: số đỉnh bậc 2 của G và G' bằng nhau)
  - deg v = deg f(v)

# Sự đẳng cấu của đồ thị (tt)

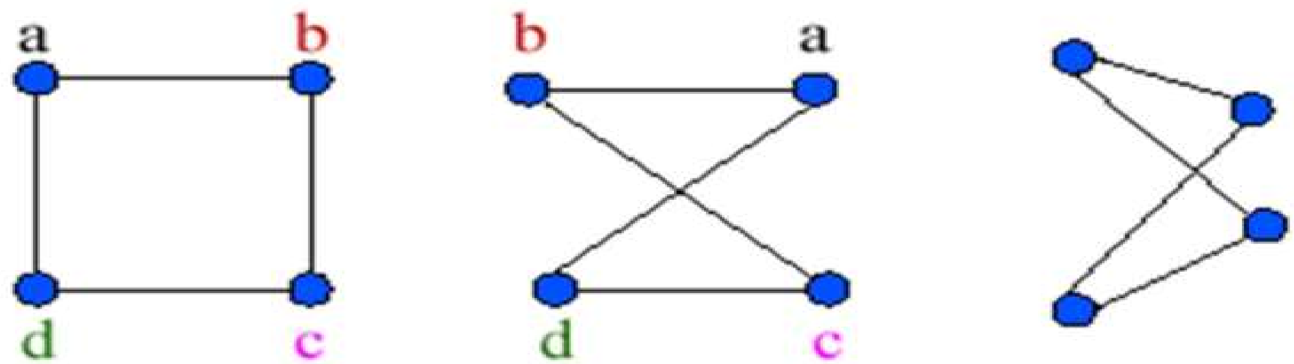
- Hãy tìm các đồ thị đẳng cấu trong các đồ thị sau:



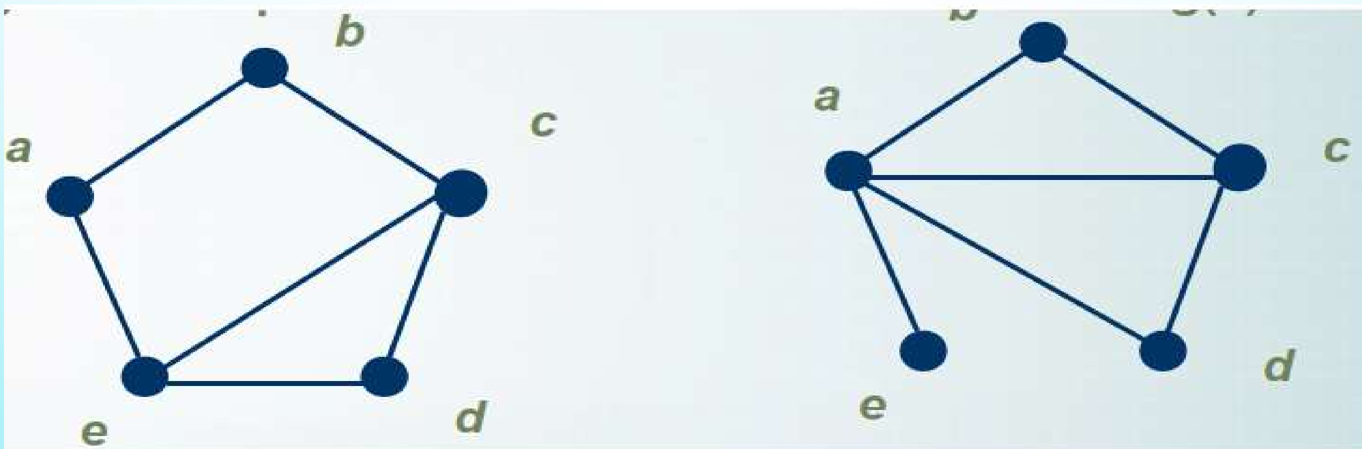
(G1) (G2) (G3) (G4) (G5) (G6) (G7)

$$G_1 \cong G_6$$
$$G_3 \cong G_5$$
$$G_4 \cong G_7$$

# Sự đẳng cấu của đồ thị

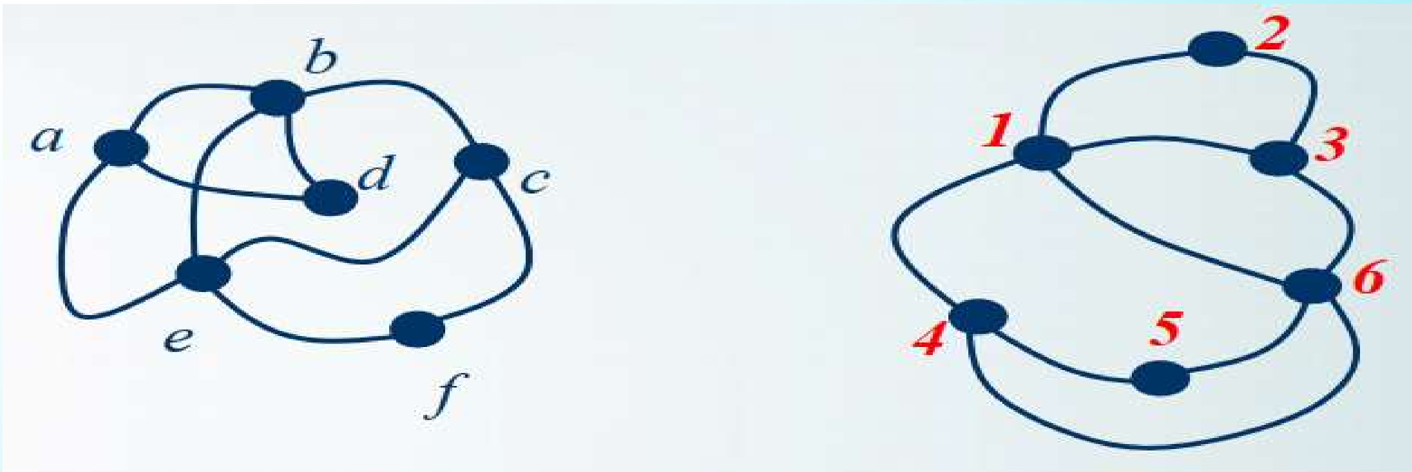



Các đồ thị sau là đẳng cấu không? Tại sao

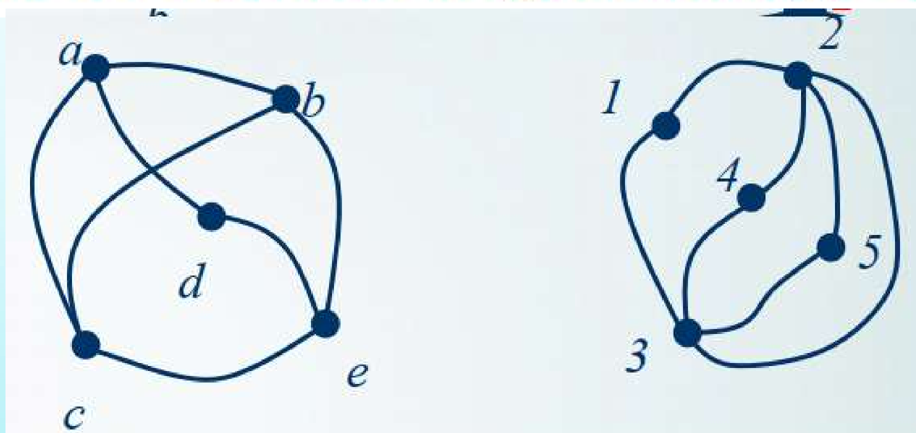



Các đồ thị sau là đẳng cấu không? Tại sao

# Sự đẳng cấu của đồ thị



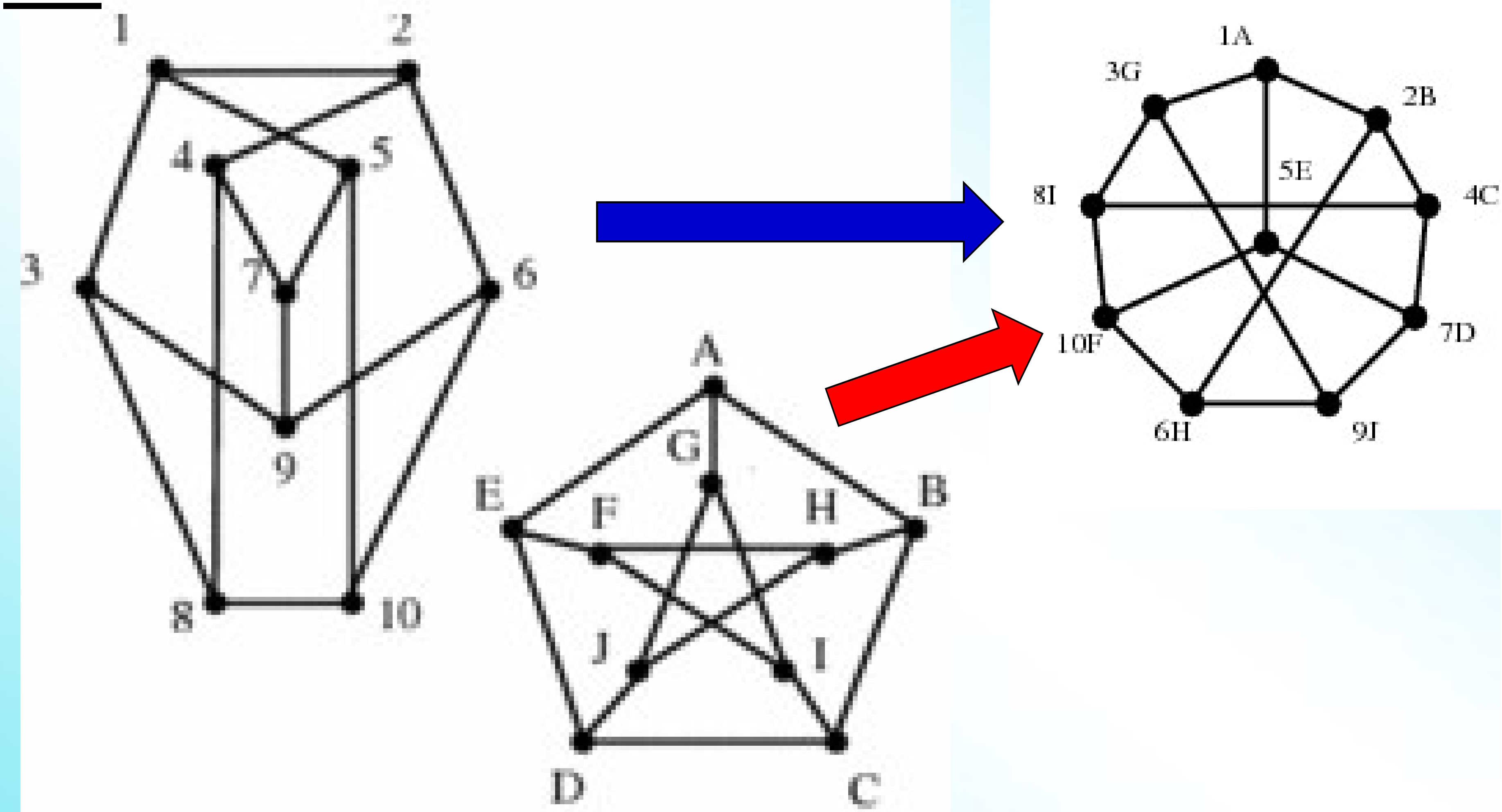

Các đồ thị sau là đẳng cấu không? Tại sao




Các đồ thị sau là đẳng cấu không? Tại sao

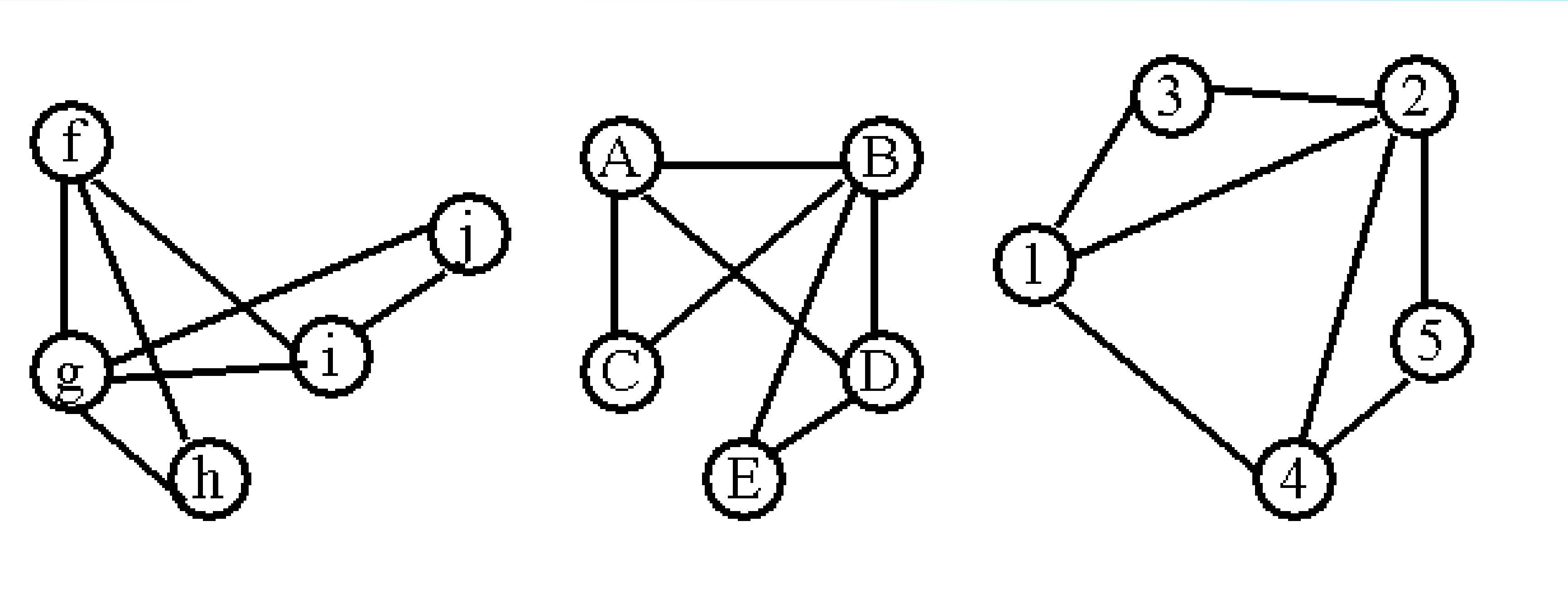
# Sự đẳng cấu của đồ thị (tt)

- **VD:**



# Sự đẳng cấu của đồ thị (tt)

- Các đồ thị sau có đẳng cấu không? Tại sao?



# Phần 2.3

# MINH HỌA VỀ BiỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

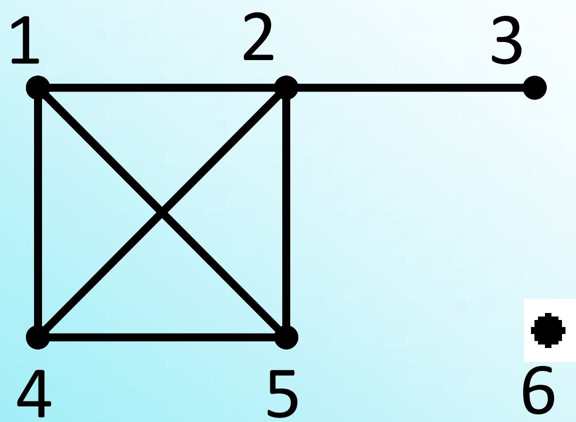
# Biểu diễn đồ thị bằng ma trận kề

- Định nghĩa đồ thị: Cấu trúc dữ liệu biểu diễn đồ thị có thể được thiết kế như sau:

```
typedef struct DOTHI
	{
		int nV;			// số đỉnh
		int nE;			// số cạnh
		int type;		// 0: vô hướng, 1: có hướng
		int mtke[maxV][maxV];	// ma trận kề
	};
```

# Nhập đồ thị từ file

- Sử dụng file text để lưu thông tin về đồ thị
- Cấu trúc chung của file text như sau:




Dòng đầu tiên chứa 3 con số thể hiện lần lượt về loại đồ thị, số đỉnh và số cạnh của đồ thị

Các dòng tiếp theo, mỗi dòng sẽ thể hiện đỉnh đầu và đỉnh cuối của một cạnh.

**DOTHI.INP**

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

# Nhập đồ thị từ file (tt)

```
int Nhap_Tu_File(char *filename, DOTHI &g)
{
    FILE *f = fopen(filename,"rt");
    if (f == NULL)
        return 0;        // Lỗi! Không mở được file
    fscanf(f,"%d %d %d \n",&g.type, &g.nV, &g.nE);
    int dd, dc;
    for (int i=1; i<=g.nV; i++)
        for (int j=1; j<=nV; j++)
            g.mtke[i][j] = 0;
    for (int k=1; k<=g.nE; k++)
    {
        fscanf(f,"%d %d \n",&dd, &dc);
        g.mtke[dd][dc] = 1;
        if (g.type==0)
            g.mtke[dc][dd] = 1;
    }
    fclose(f);      // Dong file
    return 1;
}
```

Hàm mở tập tin có tên là filename

Nhập các tham số type, nV, nE

Khởi đầu, gán toàn bộ MT kề là 0

Nếu có cạnh thì gán phần tử tương ứng là 1

Nếu là đồ thị vô hướng thì gán thêm lệnh này

# Xuất đồ thị ra màn hình

- Xuất đồ thị ra màn hình thực chất là xuất ma trận kề của nó và các thông tin liên quan: loại, số đỉnh, số cạnh.

```
void Xuat_Ra_MH(DOTHI g)
{
    cout<<“Cac thong tin ve do thi: “<<endl;
    if (g.type==1)
        cout<<“     - Day la do thi co huong. “;
    else
        cout<<“     - Day la do thi vo huong. “;
    cout<<“Do thi co “<<g.nV<<“ dinh va co “<<g.nE<<“ canh.”<<endl;
    cout<<“     - Ma tran ke cua do thi la: “<<endl;
    for (int i=1; i<=g.nV; i++)
    {
        for (int j=1; j<=g.nV; j++)
            cout<<g.mtke[i][j]<<“  “;
        cout<<endl;
    }
}
```

# Hàm tính bậc của đỉnh

```
int bac(DOTHI g, int v)
{
    if (g.type==1)
            return -1;          // Khong la do thi vo huong
    if (v<=0 || v > g.nV)
            return -1;          // Dinh khong hop le
    int bac = 0;
    for (int i=1; i<=g.nV; i++)
            if (g.mtke[v][i]==1)
                    bac ++;
    return bac;
}
```

# Hàm tính bán bậc ra (vào) của đỉnh

```
int bac_ra(DOTHI g, int v)
{
    if (g.type==0)
        return -1;
    if (v<=0 || v > g.nV)
        return -1;
    int bac = 0;
    for (int i=1; i<=g.nV; i++)
        if (g.mtke[v][i]==1)
            bac ++;
    return bac;
}
```

```
int bac_vao(DOTHI g, int v)
{
    if (g.type==0)
        return -1;
    if (v<=0 || v > g.nV)
        return -1;
    int bac = 0;
    for (int i=1; i<=g.nV; i++)
        if (g.mtke[i][v]==1)
            bac ++;
    return bac;
}
```

# Chương trình

```
#include …
#define filename “D:\\DOTHI.INP”
// Khai báo đồ thị
// Hàm Nhap_Tu_File …
// Hàm Xuat_ra_MH …

void main()
{
    DOTHI g;
    if ( ! Nhap_Tu_File(filename, g))
            cout<<“ Khong mo duoc file!!!”<<endl;
    else
            Xuat_Ra_MH(g);
    getch();
}
```