

Trường: ĐH CNTP TP.HCM Khoa: Công nghệ thông tin Bộ môn: KHDL&TTNT MH: TH Cấu trúc rời rạc	Buổi 1. CHƯƠNG 1 BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH	
-------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------	--

A. MỤC TIÊU:

- Phân biệt được các cấu trúc dữ liệu biểu diễn đồ thị trên máy tính.
- Cài đặt được các cấu trúc dữ liệu biểu diễn đồ thị trên máy tính.
- Cài đặt thuật toán nhập đồ thị từ file.

B. DỤNG CỤ - THIẾT BỊ THÍ NGHIỆM CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

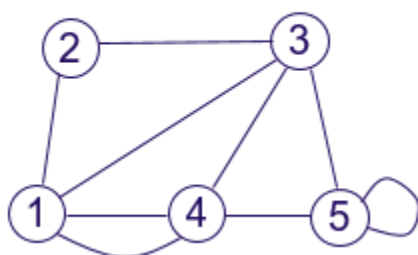
C. NỘI DUNG THỰC HÀNH

I. Tóm tắt lý thuyết

1. Ma trận kề

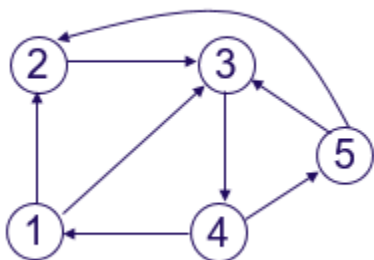
Cho $G=(V,E)$, $V=\{1,2,3,\dots,n\}$, ma trận kề $A=(A_{i,j})$ của G là ma trận vuông cấp n xác định bởi:

$A_{i,j}$ =số cạnh (cung) từ i đến j



$$A_{i,j} =$$

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



$$A_{i,j} =$$

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

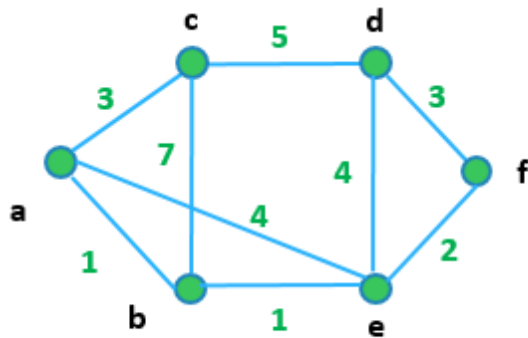
2. Ma trận trọng số

Đồ thị có trọng số là đồ thị mà mỗi cạnh (uv) có một giá trị $w(uv)$ gọi là trọng số của cạnh. Để biểu diễn đồ thị ta sử dụng **ma trận trọng số**

$W = \{w[u,v], u,v = 1,2,\dots,n\}$ với

○ $w[u,v] = w(u,v)$ nếu cạnh $(u,v) \in E$

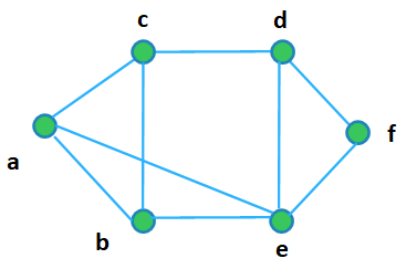
- $w[u,v] = 0$ nếu $u=v$
- $w[u,v] = \infty$ nếu cạnh $(u,v) \notin E$



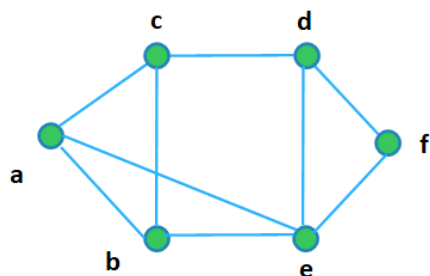
	a	b	c	d	e	f
a	0	1	3	∞	4	∞
b	1	0	7	∞	1	∞
c	3	7	0	5	∞	∞
d	∞	∞	5	0	4	3
e	4	1	∞	4	0	2
f	∞	∞	∞	3	2	0

3. Danh sách kề

Với mỗi đỉnh v của đồ thị, ta lưu trữ danh sách các đỉnh kề với v : $\mathbf{Ke}(v) = \{ u \in V : uv \in E \}$
Danh sách kề có thể được lưu dưới dạng mảng hoặc danh sách liên kết



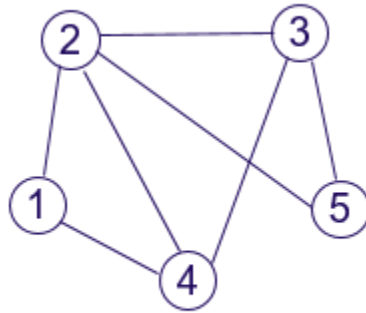
a	→	b	→	c	→	e	null		
b	→	a	→	c	→	e	null		
c	→	a	→	b	→	d	null		
d	→	c	→	e	→	f	null		
e	→	a	→	b	→	d	→	f	null
f	→	d	→	e	null				



	1	2	3	4	5
a	b	c	e	∞	∞
b	a	c	e	∞	∞
c	a	b	d	∞	∞
d	c	e	f	∞	∞
e	a	b	d	f	∞
f	d	e	∞	∞	∞

4. Danh sách cạnh

Trong trường hợp số cạnh ít hơn nhiều so với số đỉnh thì người ta thường dùng danh sách cạnh(cung) để lưu trữ đồ thị. Mỗi cạnh được biểu diễn bởi đỉnh đầu và đỉnh cuối.



Đầu	Cuối
1	2
1	4
2	3
2	4
2	5
3	4
3	5

II. Bài tập hướng dẫn mẫu

Bài tập 1: Xây dựng struct để biểu diễn một đồ thị.

```
constexpr auto MAX = 50;
struct Graph
{
    int flag;
    int n;
    int m[MAX][MAX];
};
```

Bài tập 2: Xây dựng hàm đọc ma trận kề hoặc ma trận trọng số từ file

**** Chú ý:** File input có dạng như sau

input_mtk.txt					
0					
5					
0	2	100	4	100	
2	0	-2	1	2	
100	-2	0	3	7	
4	1	3	0	100	
100	2	7	100	0	

Số đầu tiên thể hiện ma trận có hướng hoặc vô hướng. Số tiếp theo là số đỉnh của đồ thị. Bên dưới thể hiện ma trận kề hoặc ma trận trọng số.

```
Graph Doc_ma_tran_ke(string fileName)
{
    Graph g;

    ifstream fileIn;
    fileIn.open(fileName, ios::in);
    if (fileIn.fail())
    {
        cout << "\nFile không tồn tại !!!";
        g.flag = -1;
        return g;
    }

    fileIn >> g.flag;
    fileIn >> g.n;

    for (int i = 0; i < g.n; i++)
    {
```

```

        for (int j = 0; j < g.n; j++)
        {
            fileIn >> g.m[i][j];
        }
    }
    fileIn.close();
    return g;
}

```

III. Bài tập ở lớp

Bài tập 3: Viết hàm đọc danh sách cạnh từ file và trả về đồ thị được định nghĩa ở BT1.

**** Chú ý:** File input có dạng như sau

input_dsc.txt	
0	
5	
4	
0	2
1	3
1	4
3	4

Số đầu tiên thể hiện ma trận có hướng hoặc vô hướng. Hai số tiếp theo là số đỉnh và số cạnh của đồ thị. Bên dưới thể hiện danh sách cạnh gồm đỉnh đầu và đỉnh cuối.

Graph Doc_danh_sach_canh(string fileName)

```

{
    Graph g;

    // Đọc file
    ifstream fileIn;
    fileIn.open(fileName, ios::in);
    if (fileIn.fail())
    {
        cout << "\nFile không tồn tại !!!";
        g.flag = -1;
        return g;
    }

    int flag, n;
    fileIn >> flag;
    fileIn >> n;
    g = khoi_tao(n);
    g.flag = flag;

    int so_canh;
    fileIn >> so_canh;

    for (int i = 0; i < so_canh; i++)
    {
        int dinh_dau, dinh_cuoi;
        fileIn >> dinh_dau;
        fileIn >> dinh_cuoi;

        // Chuyển sang ma trận kề
        g.m[dinh_dau][dinh_cuoi] = 1;
        if (g.flag == 0)
            g.m[dinh_cuoi][dinh_dau] = 1;
    }
}

```

```

    fileIn.close();
    return g;
}

```

Bài tập 4: Viết hàm đọc danh sách kề từ file và trả về đồ thị được định nghĩa ở BT1.

**** Chú ý:** File input có dạng như sau

input_dsk.txt
0
13
0 1 3 11
1 3
2 6
3 5 6
4 5 7 8
5 6 12
7 8
9 10 11
10 11

Số đầu tiên thể hiện ma trận có hướng hoặc vô hướng. Số tiếp theo là số đỉnh của đồ thị. Bên dưới thể hiện danh sách kề gồm một đỉnh và danh sách các đỉnh kề với nó.

```

Graph Doc_danh_sach_ke(string fileName)
{
    Graph g;

    // Đọc file
    ifstream fileIn;
    fileIn.open(fileName, ios::in);
    if (fileIn.fail())
    {
        cout << "\nFile không tồn tại !!!";
        g.flag = -1;
        return g;
    }

    // Đọc số đỉnh
    int flag, n;
    fileIn >> flag;
    fileIn >> n;
    g = khoi_tao(n);
    g.flag = flag;

    // Đọc các thành phần của danh sách kề
    string line;
    while (getline(fileIn, line))
    {
        stringstream stream(line);
        int dinh_dau;
        stream >> dinh_dau;
        while (1)
        {
            int dinh_cuoi;
            stream >> dinh_cuoi;
            if (!stream) break;
            g.m[dinh_dau][dinh_cuoi] = 1;
        }
    }
}

```

```

        if (g.flag == 0)
            g.m[dinh_cuoi][dinh_dau] = 1;
    }
}
fileIn.close();
return g;
}

```

Bài tập 5: Viết hàm xuất ra màn hình đồ thị được định nghĩa ở BT1.

```

void Xuat_do_thi(Graph g)
{
    int n = g.n;
    cout << "Số đỉnh: " << n << "\n";
    if (g.flag) {
        cout << "Đồ thị có hướng.\n";
    }
    else {
        cout << "Đồ thị vô hướng.\n";
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            cout << g.m[i][j] << " ";
        }
        cout << "\n";
    }
}

```

IV. Bài tập về nhà

Bài tập 6: Viết hàm ghi ra file đồ thị được định nghĩa ở BT1 thành dạng ma trận trọng số hoặc ma trận kề. Dạng file đúng cấu trúc như BT3.

Bài tập 7: Viết hàm ghi ra file đồ thị được định nghĩa ở BT1 thành dạng danh sách cạnh. Dạng file đúng cấu trúc như BT4.

Bài tập 8: Viết hàm ghi ra file đồ thị được định nghĩa ở BT1 thành dạng danh sách kề. Dạng file đúng cấu trúc như BT5.

Bài tập 9: Viết hàm **Ke(g, v)** lấy thông số là đồ thị g và đỉnh v, xuất ra một mảng chứa danh sách các đỉnh kề của v.