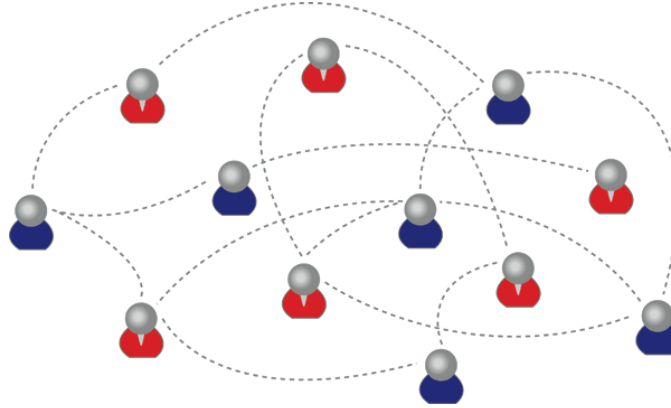


LAB 1: BIỂU DIỄN ĐỒ THỊ VÀO MÁY TÍNH

I. Ứng dụng đồ thị:



Hình 1 Mạng xã hội(Social Networks)

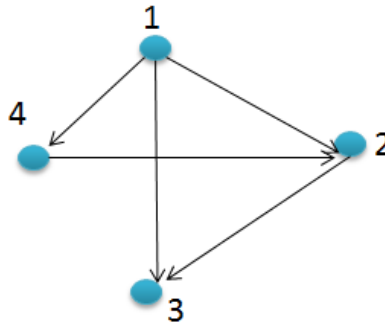


Hình 2 Finding Shortest Path

II. Ma trận kề (Adjacency matrix):

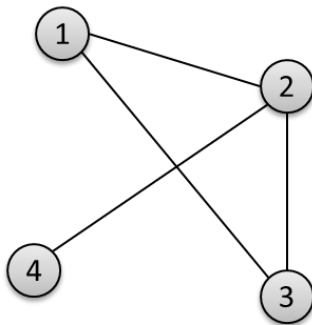
Người ta thường dùng ma trận kề($n \times n$) biểu diễn một đồ thị. Một giá trị $a[i,j]$ ứng với trọng số của cạnh giữa hai đỉnh i và j trong đồ thị. Nếu giữa 2 đỉnh của đồ thị không có cung thì phần tử $a[i, j]=0$

1. **Biểu diễn ma trận cho đồ thị có hướng:**



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

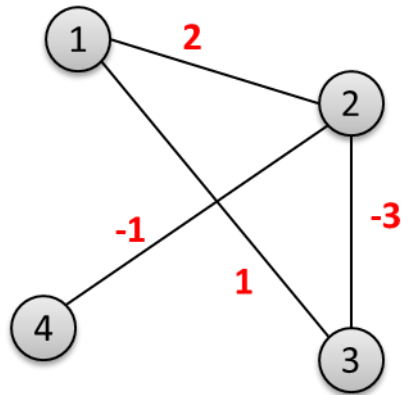
2. **Biểu diễn ma trận cho đồ thị vô hướng:**



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

Đối với đồ thị vô hướng ta sẽ có giá trị $a[i, j] = a[j, i]$ chính là một ma trận đối xứng qua đường chéo chính.

3. Đồ thị có trọng số:



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

4. Cài đặt:

Thông thường ma trận kề được lưu trữ trên tập tin và được chương trình đọc lên để thực hiện các thuật toán trên đồ thị tương ứng.

Ví dụ: Đồ thị có hướng sẽ được lưu thành tập tin Graph.txt như sau:

```

4
0 1 1 1
0 0 1 0
0 0 0 0
0 1 0 0

```

5. Tổ chức chương trình

Ta có thể dùng struct hoặc class.

```
#define MAX 100
```

```
struct GRAPH {
```

```
int n;
```

```
int a[MAX][MAX];
```

```
}GRAPH;
```

```

class GRAPH {                                     //các phương thức khác
    int n;                                       };
    int a[MAX][MAX];

```

6. Cài đặt:

- Thủ tục: (*sinh viên tự cài đặt*)
- Lớp và toán tử operator.

```

class GRAPH
{
    private:
        int n;
        int a[MAX][MAX];
    public:
        GRAPH(void);
        ~GRAPH(void);
        friend istream& operator >> (istream& inDevice, GRAPH
&g);
        friend ostream& operator << (ostream& outDevice, GRAPH
&g);
        //...các hàm khác
        //....

};

istream& operator>>(istream& inDevice, GRAPH &g)
{
    int n,temp;
    int i,j;
    inDevice>>g.n;
    for( i=0;i<g.n;i++)
        for(j=0;j<g.n;j++)
        {
            g.a[i][j]=0;
            if(inDevice)
            {
                inDevice>>temp;
                g.a[i][j]=temp;
            }
        }
    return inDevice;
}

ostream& operator<<(ostream& outDevice, GRAPH &g)
{
    for(int i=0;i<g.n;i++)
    {
        for(int j=0;j<g.n;j++)
        {

```

```

        outDevice<<(int)g.a[i][j]<<" ";
    }
    outDevice<<endl;
}
return outDevice;
}

```

Hàm đọc ma trận từ tập tin:

```

int GRAPH::ReadFile(char *filename)
{
    ifstream fVar(filename);
    fVar>>*(this);
    return 1;
}

```

Hàm ghi ma trận vào tập tin:

```

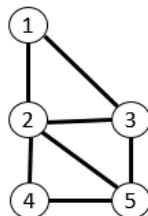
int GRAPH::WriteFile(char *filename)
{
    ofstream fVar1(filename);
    fVar1<<n<<endl;
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
            fVar1<<a[i][j];
        fVar1<<endl;
    }
    return 1;
}

```

III. Danh sách cạnh (Edge list):

Đồ thị biểu diễn dưới dạng danh sách cạnh bằng cách liệt kê tất cả các cạnh của đồ thị trong một danh sách, mỗi phần tử của danh sách là một cặp (u, v) tương ứng với một cạnh của đồ thị. Danh sách này có thể được cài đặt bằng mảng hoặc danh sách liên kết.

Ví dụ:

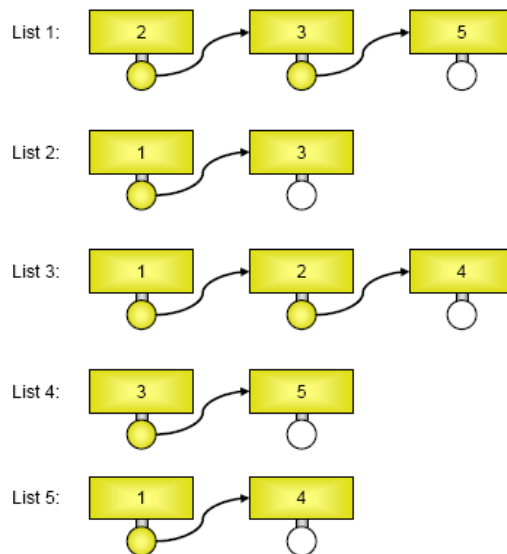
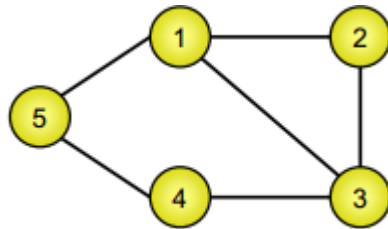


1	2	3	4	5	6	7
(1, 2)	(1, 3)	(2, 3)	(2, 4)	(2, 5)	(3, 5)	(4, 5)

IV. Danh sách kề (Adjacency list):

Trong cách biểu diễn này, với mỗi đỉnh v của đồ thị, ta lưu trữ tương ứng với một danh sách chứa các đỉnh kề với nó.

Ví dụ:



V. Bài tập:

Cho một đồ thị $G = (V, E)$ với V : số đỉnh của đồ thị, E : số cạnh của đồ thị.

- Viết chương trình biểu diễn đồ thị G vào máy tính dưới dạng cấu trúc ma trận kề (dùng mảng tĩnh, mảng động, vector).

Input : tập tin *Graph.txt* (lưu trữ dạng ma trận kề)

Output : Hiển thị ma trận kề ra màn hình

2. Viết chương trình biểu diễn đồ thị G vào máy tính dưới dạng cấu trúc danh sách kề.

Input : tập tin *Graph.txt* (lưu trữ dạng danh sách kề)

Output : Hiển thị danh sách kề ra màn hình

3. Viết chương trình biểu diễn đồ thị G vào máy tính dưới dạng danh sách cạnh và xuất ra file.

Input : tập tin *Graph.txt* (lưu trữ dạng danh sách cạnh)

Output : Hiển thị danh sách cạnh ra màn hình

4. Viết hàm (phương thức) chuyển đổi cấu trúc đồ thị G từ dạng ma trận kề sang danh sách kề.
5. Viết hàm (phương thức) chuyển đổi cấu trúc đồ thị G từ dạng ma trận kề sang danh sách cạnh.
6. Viết hàm (phương thức) chuyển đổi cấu trúc đồ thị G từ dạng danh sách kề sang ma trận kề.
7. Xuất ra tập tin “*InforGraph.txt*” theo yêu cầu sau:
 - Số cạnh, số đỉnh của đồ thị.
 - Các đỉnh có số bậc lớn nhất.
 - Các đỉnh có số bậc nhỏ nhất.
 - Liệt kê số lượng và tên các đỉnh bậc chẵn.
 - Liệt kê số lượng và tên các đỉnh bậc lẻ
 - Các đỉnh cô lập.
 - Các đỉnh treo.
8. Kiểm tra đồ thị có phải là đồ thị vô hướng, nếu đúng xuất ra file số bậc của đồ thị.

VI. Tài liệu tham khảo :

- [1.] Huỳnh Lê Tấn Tài, Bài tập thực hành Lý thuyết đồ thị, Khoa CNTT-TUD, ĐH Tôn Đức Thắng.
- [2.] Bài tập thực hành Lý thuyết đồ thị, Khoa CNTT, ĐH Khoa Học Tự Nhiên.