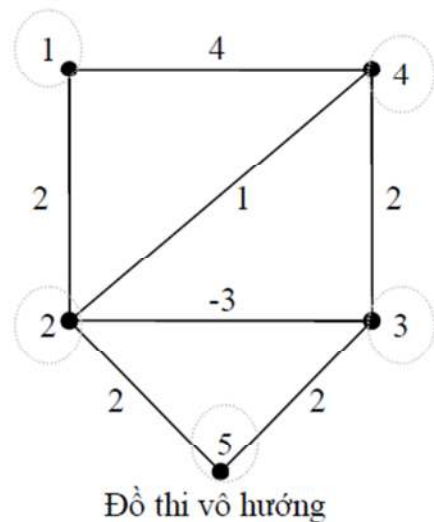
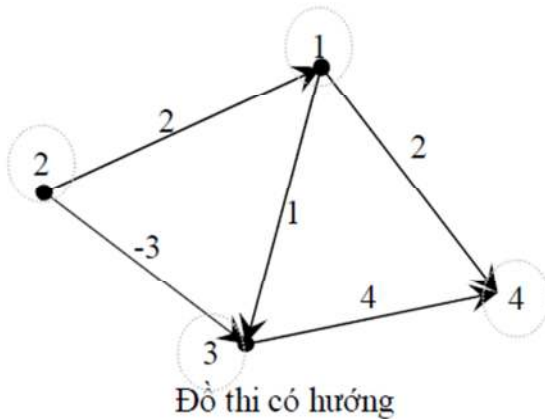


## Ma trận kề & Liên thông

### I. Ma trận kề

#### Lý thuyết:

Trong lý thuyết đồ thị, người ta thường dùng ma trận kề để biểu diễn một đồ thị. Một giá trị  $a[i,j]$  trong ma trận (dòng  $i$  cột  $j$  của ma trận  $A$ ) ứng số cạnh/trọng số của cạnh giữa hai đỉnh  $i$  và  $j$  trong đồ thị. Nếu giữa 2 đỉnh của đồ thị không có cạnh thì phần tử  $a[i,j] = 0$



Theo định nghĩa trên thì đồ thị có hướng ở trên sẽ có ma trận kề là

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

#### Nhận xét:

Đối với đồ thị vô hướng, ta sẽ có giá trị  $a[i,j] = a[j,i]$  (xem như có hai cạnh có hướng từ  $i \rightarrow j$  và từ  $j \rightarrow i$  trong đồ thị). Ma trận kề của đồ thị vô hướng ở trên là:

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

- Đường chéo chính trong đồ thị có giá trị 0 (do trong đồ thị không có khuyên)
- Ma trận kề của đồ thị vô hướng đối xứng qua đường chéo chính vì  $a[i,j] = a[j,i]$

### **Cài đặt:**

#### **1. Lưu trữ trên tập tin**

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 ở trên sẽ được lưu trong tập tin DOTH1.TXT như sau

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

#### **2. Tổ chức lớp**

Xây dựng lớp Matrix cài đặt **toán tử**  $>>$  và  $<<$  (cho việc nhập/xuất ma trận) cùng các phương thức cần thiết khác.

```
class Matrix
{
private:
    vector<vector<char>> array2D;

public:
    friend istream& operator >> (istream& inDevice, Matrix& a);
    friend ostream& operator << (ostream& outDevice, Matrix& a);

    //các phương thức khác
    //...
};
```

#### **3. Đọc/ghi ma trận kề**

Tham khảo đoạn mã nguồn mẫu trong bài hướng dẫn “Cài đặt mảng 2 chiều dùng vector<T>” – thầy Trần Đan Thu

#### 4. Kiểm tra tính hợp lệ của ma trận

Sau khi đã lấy dữ liệu từ tập tin, chúng ta cần kiểm tra dữ liệu này có hợp lệ không? Tùy thuộc vào loại đồ thị mà ta có cách kiểm tra ma trận kề phù hợp. Chẳng hạn với đồ thị vô hướng đơn không khuyên thì ta kiểm tra các giá trị  $a[i][i]$  xem có giá trị nào khác 0 hay không hoặc  $a[i][j]$  xem có giá trị nào  $> 1$  hay không, nếu có thì ma trận kề không hợp lệ.

Như vậy, ta có thể xây dựng một lớp đối tượng Graph để thể hiện đồ thị. Trong đó, cài đặt phương thức kiểm tra tính hợp lệ của ma trận kề.

```
class Graph
{
    //lưu trữ ma trận kề của đồ thị
    Matrix adjacentMatrix;

    //các phương thức khác
    //...

    //Kiểm tra tính hợp lệ của ma trận kề
    bool IsAdjacentMatValid();
};
```

## II. Liên thông

### Yêu cầu

Với 1 đồ thị cho trước, yêu cầu sinh viên kiểm tra đồ thị có liên thông không và cho biết đồ thị có bao nhiêu thành phần liên thông. Đồ thị liên thông là đồ thị chỉ có **một** thành phần liên thông.

### Cài đặt

1. Tại thời điểm khởi động tất cả các đỉnh chưa viếng thăm ( $Nhan[i] = 0$ ).
2. Chọn 1 đỉnh  $i$  bất kỳ chưa được viếng thăm ( $Nhan[i] == 0$ ), sử dụng hàm ViengTham để duyệt đỉnh  $i$  và tất cả các đỉnh  $j$  chưa được viếng thăm ( $Nhan[j] == 0$ ) có nối với đỉnh  $i$ . Kết thúc mỗi lần duyệt, ta được 1 thành phần liên thông. Để đánh dấu đỉnh này đã viếng thăm ta gán  $Nhan[j] = SoThanhPhanLT$ .
3. Tăng giá trị của SoThanhPhanLT lên 1 ( $SoThanhPhanLT++$ )

## Bài tập thực hành

### III. Quy định

Thời gian làm bài: **1 tuần** (*dealine xem trên moodle*)

Loại bài tập: **cá nhân**

Cấu trúc bên trong thư mục **MSSV**: chứa toàn bộ source code.

Nén toàn bộ thư mục thành file **MSSV.zip** hoặc **.rar**

Nộp bài lên moodle.

**Lưu ý: tất cả các bài làm sai qui định sẽ không được chấm tức là 0 điểm.**

### IV. Đề bài

Xác định các *thành phần liên thông* của đồ thị vô hướng đơn dựa vào ma trận kề đọc từ tập tin đầu vào. Chương trình được viết dưới dạng Console với **hai** tham số dòng lệnh lần lượt là đường dẫn *tập tin đầu vào* và đường dẫn *tập tin đầu ra*.

*Ví dụ tham số dòng lệnh:*

`MSSV.exe input.txt Output.txt`

*Hoặc*

`MSSV.exe C:\input.txt D:\Output.txt`

**Lưu ý: đường dẫn/tên tập tin đầu vào và đầu ra có thể thay đổi (không cố định)**

#### **Cấu trúc file dữ liệu đầu vào:**

- Dòng đầu tiên: **số đỉnh đồ thị ( $N$ )**
- $N$  dòng tiếp theo: **ma trận** kề của đồ thị với quy ước:
  - ✓  $A[i][j] = 1$ : có đường nối trực tiếp từ  $i$  đến  $j$
  - ✓  $A[i][j] = 0$ : không có đường nối trực tiếp từ  $i$  đến  $j$

Các đỉnh được đánh **chỉ số từ 0**

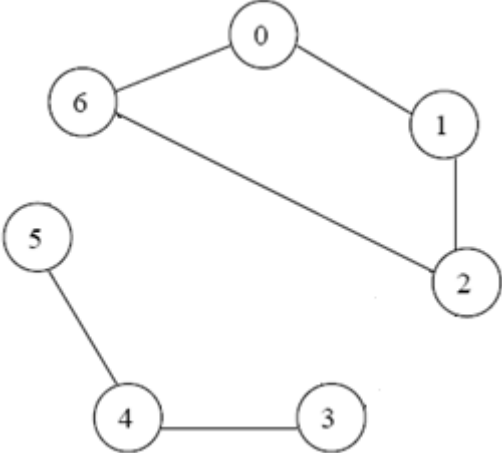
#### **Cấu trúc file dữ liệu đầu ra:**

- Dòng đầu tiên: **số thành phần liên thông của đồ thị ( $M$ )**
- $M$  dòng tiếp theo ứng với mỗi thành phần liên thông của đồ thị xuất ra:

- Chỉ số của các đỉnh thuộc thành phần liên thông đó (cách nhau bởi *khoảng trắng* và sắp theo *thứ tự tăng* chỉ số đỉnh)

Các thành phần liên thông được xuất theo thứ tự **tăng dần** theo số đỉnh.  
 Nếu hai thành phần liên thông có **cùng số đỉnh** thì thành phần liên thông có **đỉnh nhỏ hơn được xuất trước**.

🌈 Ví dụ:

Tập tin đầu vào	Đồ thị	Tập tin đầu ra
<pre> 7 0 1 0 0 0 0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0                     </pre>		<pre> 2 3 4 5 0 1 2 6                     </pre>