

GRAPH

Giảng viên: Tiến sĩ Bùi Thanh Hùng
Trưởng Lab Khoa học Phân tích dữ liệu và Trí tuệ nhân tạo
Giám đốc chương trình Hệ thống thông tin
Đại học Thủ Dầu Một

Email: hungbt3@fe.edu.vn

Website: <https://sites.google.com/site/hungthanhbui1980/>

There are many student fail the exams, here are example in our data (a number is the name of the course).

A: 1-3

E: 5-6

B: 1-2

F: 2-5

C: 2-4

G: 3-6

D: 3-5

How should we organize a schedule for retake exams

Đồ thị

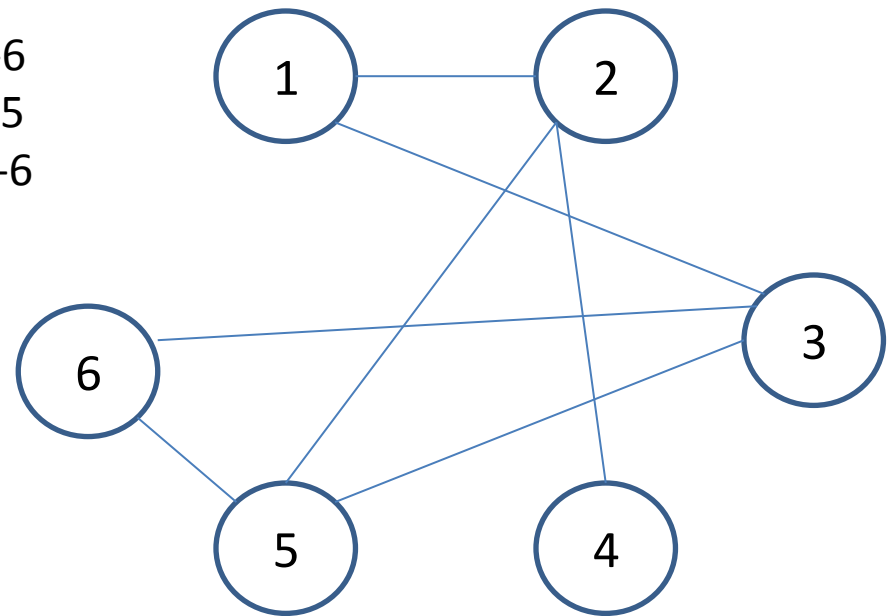
- Chúng ta có nhiều cách giải khác nhau
- Đây là bộ dữ liệu nhỏ
- Nếu bộ dữ liệu lớn, thì chúng ta sẽ phải sắp xếp lịch cho tối ưu
- Dùng Graph đem lại hiệu quả tốt
- $G(\text{Vertex}, \text{Edge})$
- V: Đỉnh, E: Cạnh

Graph(Vertice, Edge)

- Đỉnh: Môn
- Cạnh: nối 2 môn mà sinh viên phải thi lại

A: 1-3
B: 1-2
C: 2-4
D: 3-5

E: 5-6
F: 2-5
G: 3-6



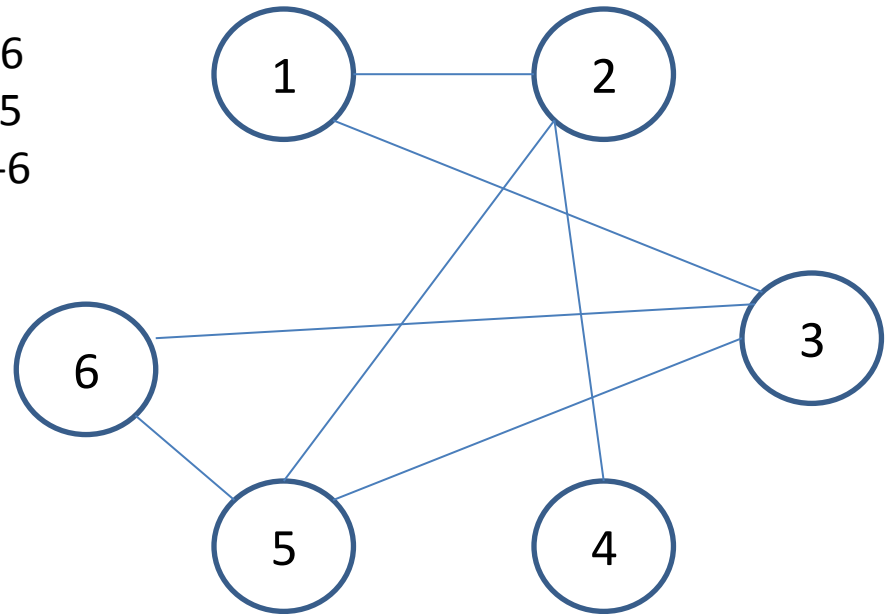
- Bài toán cần giải quyết:

Graph(Vertice, Edge)

- Đỉnh: Môn
- Cạnh: nối 2 môn mà sinh viên phải thi lại

A: 1-3
B: 1-2
C: 2-4
D: 3-5

E: 5-6
F: 2-5
G: 3-6



Bài toán cần giải quyết:

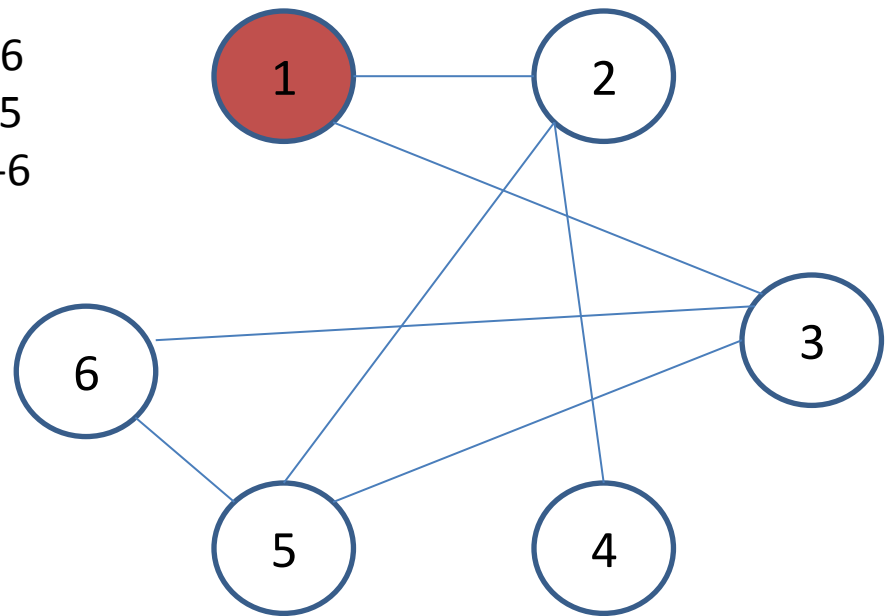
- Tô màu đồ thị
- Liên thông
- Duyệt đồ thị
- Đường đi
- Cây khung
- Euler, Hamilton
- Lát cắt cực đại, cực tiểu

Graph(Vertice, Edge)

- Đỉnh: Môn
- Cạnh: nối 2 môn mà sinh viên phải thi lại

A: 1-3
B: 1-2
C: 2-4
D: 3-5

E: 5-6
F: 2-5
G: 3-6



Bài toán cần giải quyết:

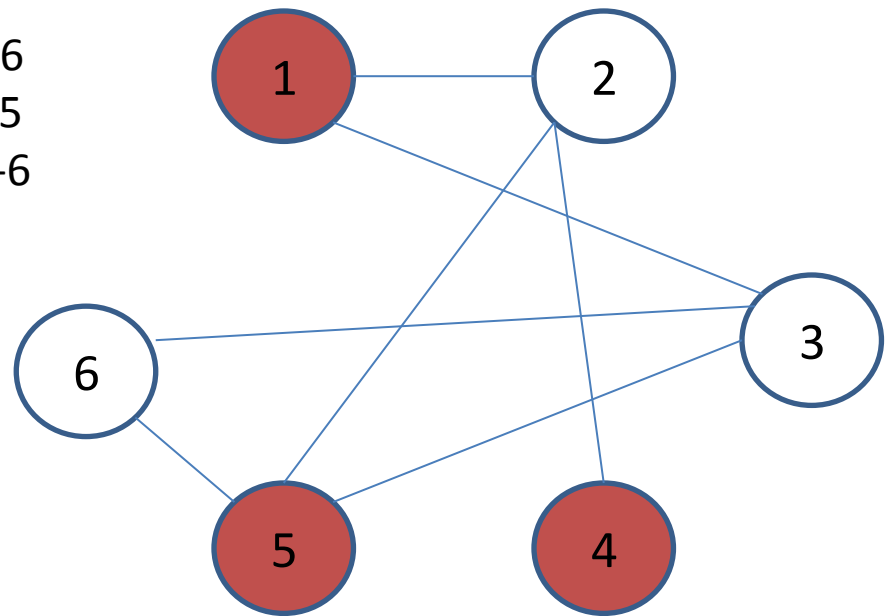
- Tô màu đồ thị
- Liên thông
- Duyệt đồ thị
- Đường đi
- Cây khung
- Euler, Hamilton
- Lát cắt cực đại, cực tiểu

Graph(Vertice, Edge)

- Đỉnh: Môn
- Cạnh: nối 2 môn mà sinh viên phải thi lại

A: 1-3
B: 1-2
C: 2-4
D: 3-5

E: 5-6
F: 2-5
G: 3-6



Bài toán cần giải quyết:

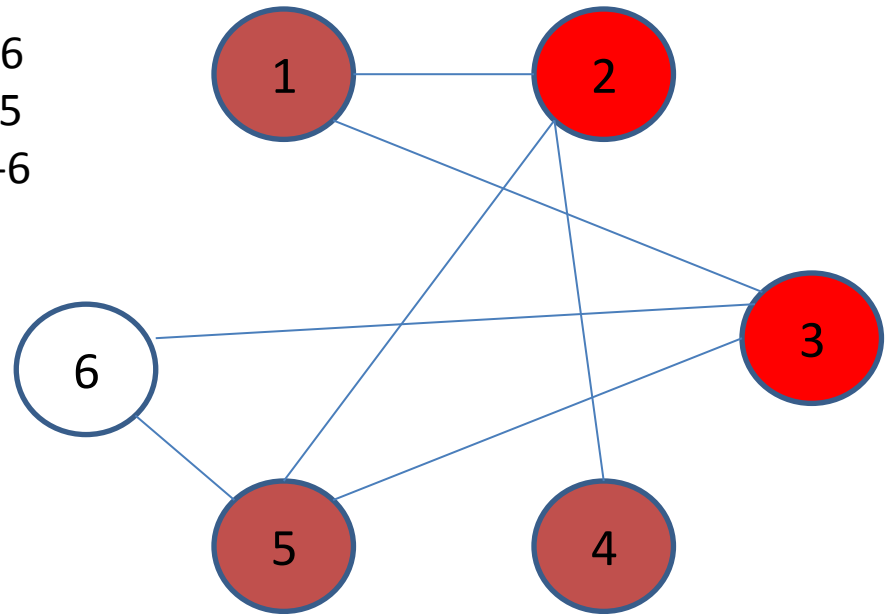
- Tô màu đồ thị
- Liên thông
- Duyệt đồ thị
- Đường đi
- Cây khung
- Euler, Hamilton
- Lát cắt cực đại, cực tiểu

Graph(Vertice, Edge)

- Đỉnh: Môn
- Cạnh: nối 2 môn mà sinh viên phải thi lại

A: 1-3
B: 1-2
C: 2-4
D: 3-5

E: 5-6
F: 2-5
G: 3-6



Bài toán cần giải quyết:

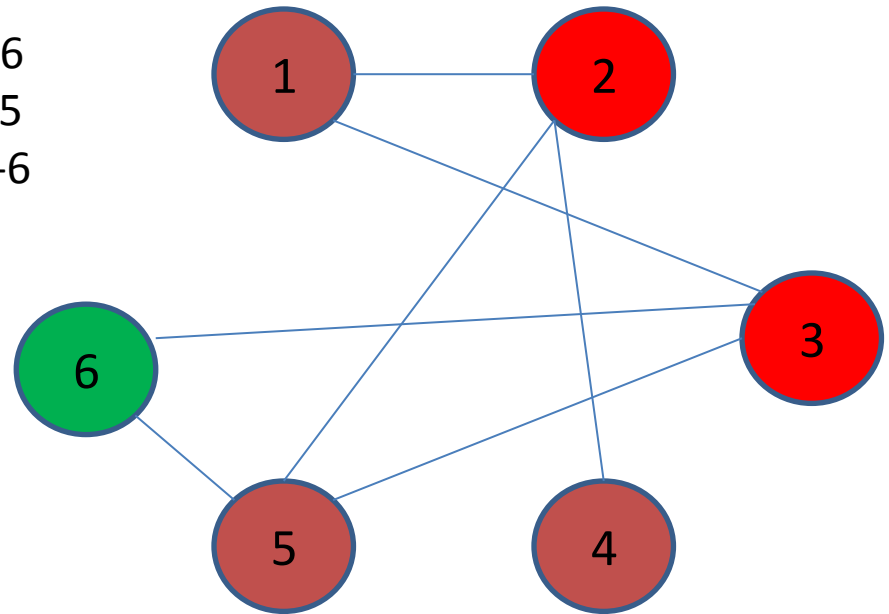
- Tô màu đồ thị
- Liên thông
- Duyệt đồ thị
- Đường đi
- Cây khung
- Euler, Hamilton
- Lát cắt cực đại, cực tiểu

Graph(Vertice, Edge)

- Đỉnh: Môn
- Cạnh: nối 2 môn mà sinh viên phải thi lại

A: 1-3
B: 1-2
C: 2-4
D: 3-5

E: 5-6
F: 2-5
G: 3-6



Bài toán cần giải quyết:

- Tô màu đồ thị
- Liên thông
- Duyệt đồ thị
- Đường đi
- Cây khung
- Euler, Hamilton
- Lát cắt cực đại, cực tiểu

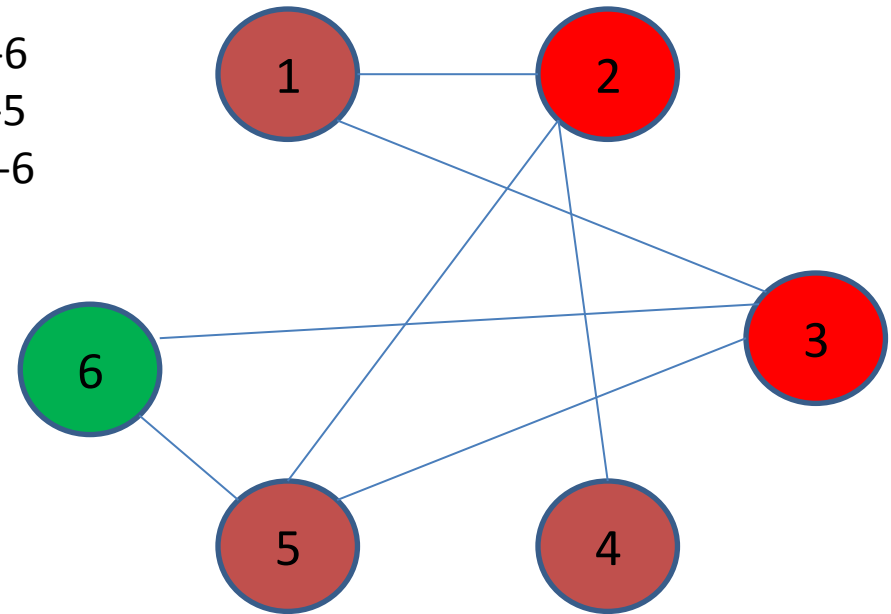
Graph(Vertice, Edge)

- 1- Bài toán có thể giải quyết bằng GRAPH
- 2- Xây dựng được GRAPH
- 3- Xác định bài toán trong GRAPH
- 4- Xác định giải thuật cho bài toán

- Đỉnh: Môn
- Cạnh: nối 2 môn mà sinh viên phải thi lại

A: 1-3
B: 1-2
C: 2-4
D: 3-5

E: 5-6
F: 2-5
G: 3-6



Bài toán cần giải quyết:

- Tô màu đồ thị
- Liên thông
- Duyệt đồ thị
- Đường đi
- Cây khung
- Euler, Hamilton
- Lát cắt cực đại, cực tiểu

Graph(Vertice, Edge)

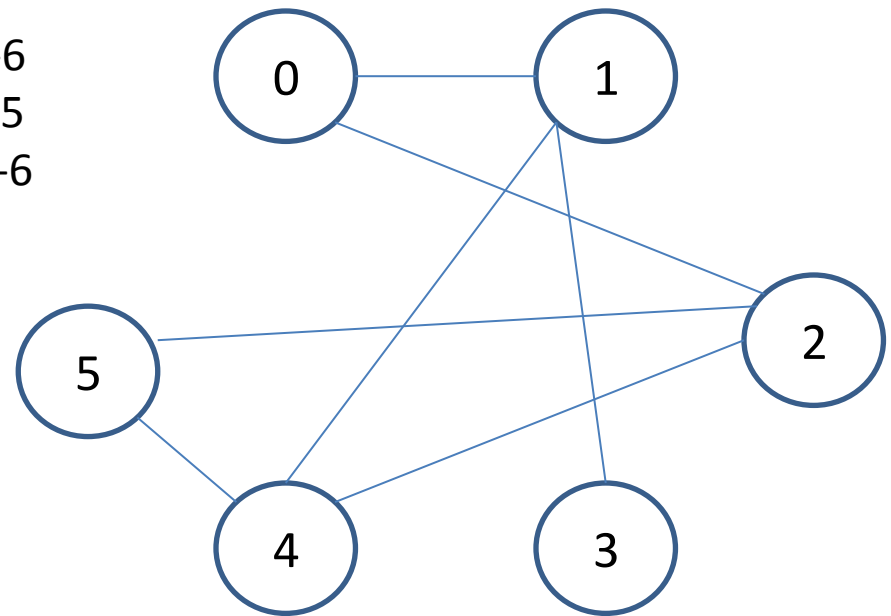
- Đỉnh: Môn
- Cạnh: nối 2 môn mà sinh viên phải thi lại

A: 1-3
B: 1-2
C: 2-4
D: 3-5

E: 5-6
F: 2-5
G: 3-6

6

0	1	1	0	0	0
1	0	0	1	1	0
1	0	0	0	1	1



Graph(Vertice, Edge)

- Đỉnh: Môn
- Cạnh: nối 2 môn mà sinh viên phải thi lại

Bài toán cần giải quyết:

- Tô màu đồ thị
- Liên thông
- Duyệt đồ thị
- Đường đi
- Cây khung
- Euler, Hamilton
- Lát cắt cực đại, cực tiểu

Graph(**V**ertice,**E**dge)

Các loại đồ thị

- Đơn, Đa đồ thị
- Có hướng/Vô hướng
- Trọng số/Không trọng số,...

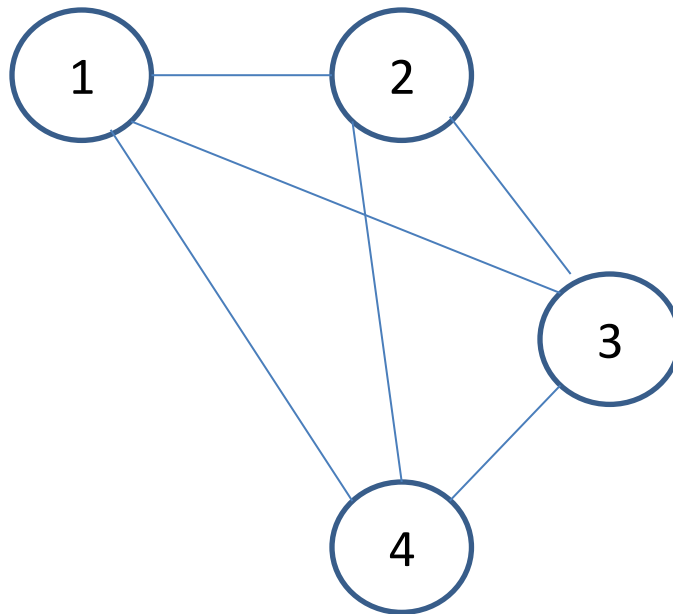
Graph(Vertice, Edge)

Các loại đồ thị

- Đơn, Đa đồ thị
 - Có hướng/Vô hướng
 - Trọng số/Không trọng số,...
-
- Các đồ thị đặc biệt: K_4 , C_3 , Q_3 , W_4 , $K_{2,3}$

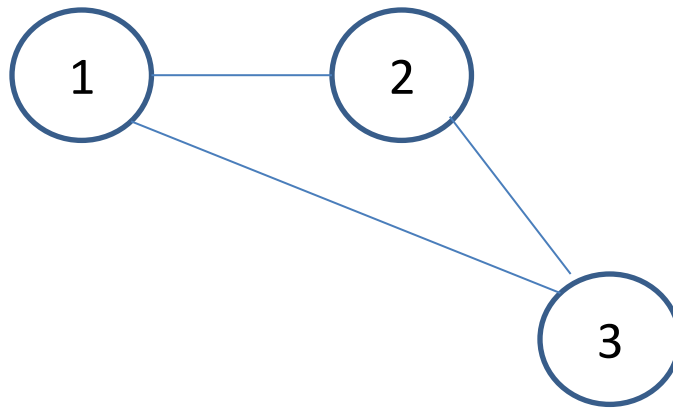
Graph(Vertice, Edge)

- Các đồ thị đặc biệt: **K4**, C3, Q3, W4, K23
- K_n : số cạnh $n(n-1)/2$: Complete



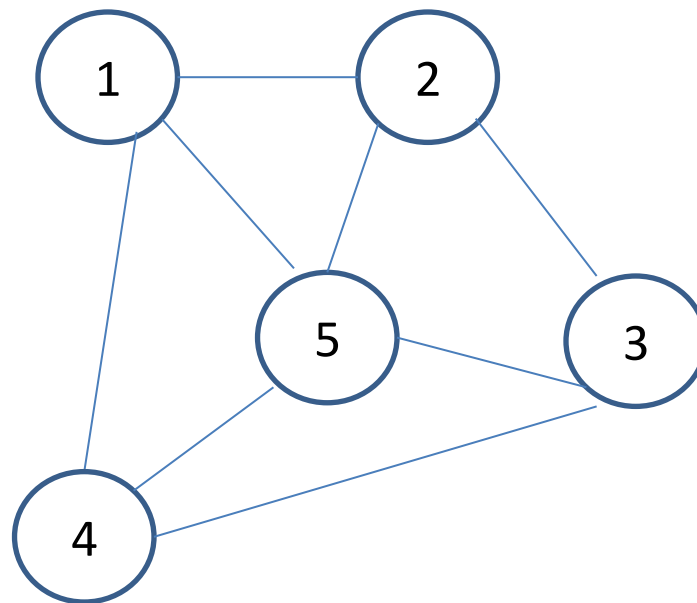
Graph(Vertice, Edge)

- Các đồ thị đặc biệt: K_4 , C_3 , Q_3 , W_4 , $K_{2,3}$
- C_3 – Circle: n đỉnh – n cạnh



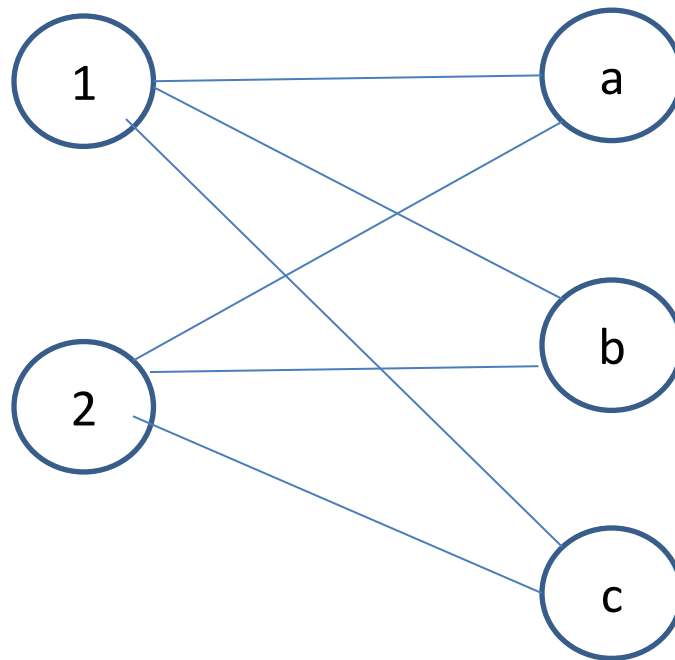
Graph(Vertice, Edge)

- Các đồ thị đặc biệt: K_4 , C_3 , Q_3 , W_4 , $K_{2,3}$
- W_4 – Wheel: n đỉnh – $2n$ cạnh



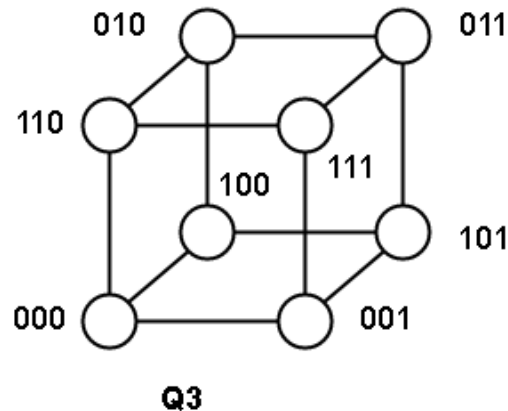
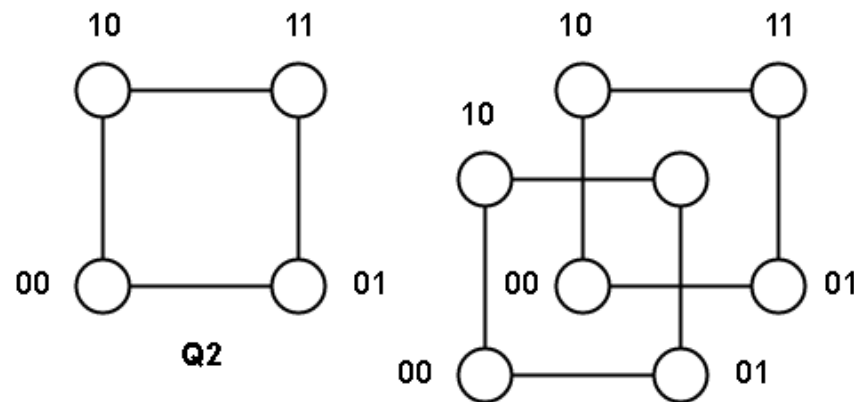
Graph(**V**ertice,**E**dge)

- Các đồ thị đặc biệt: K_4 , C_3 , Q_3 , W_4 , **$K_{2,3}$**
- **$K_{2,3}$** – Đồ thị đầy đủ 2 phía



Graph(Vertice, Edge)

- Các đồ thị đặc biệt: K_4 , C_3 , Q_3 , W_4 , $K_{2,3}$



Graph(Vertice, Edge)

- Tổng bậc = $2|E|$
- Tổng bậc lẻ = Tổng bậc chẵn



1+: ra 1

1-: vào 0

2+: ra 0

2-: vào 1

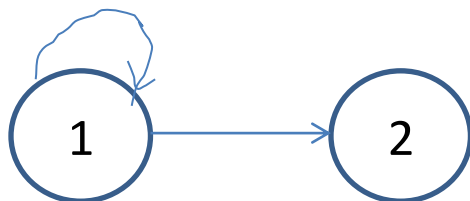


1+: ra 2

1-: vào 1

2+: ra 0

2-: vào 1



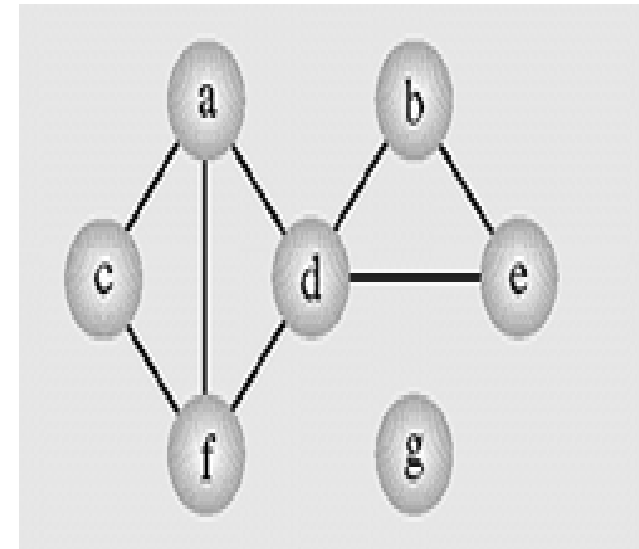
Biểu diễn đồ thị: Graph(**V**ertice, **E**dge)

- Adjacency Matrix - Ma trận kề $O(V \times V)$
- Adjacency List – Danh sách kề

Graph Representation – 1

(Adjacency matrix) $a[0..6,0..6]$

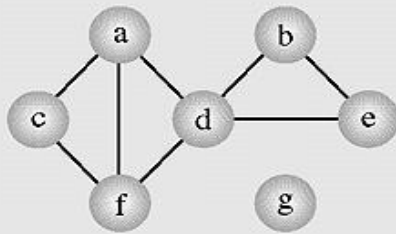
	a	b	c	d	e	f	g
a	0	0	1	1	0	1	0
b	0	0	0	1	1	0	0
c	1	0	0	0	0	1	0
d	1	1	0	0	1	1	0
e	0	1	0	1	0	0	0
f	1	0	1	1	0	0	0
g	0	0	0	0	0	0	0



Graph represented by an adjacency matrix

Graph Representation – 2

(Adjacency list)

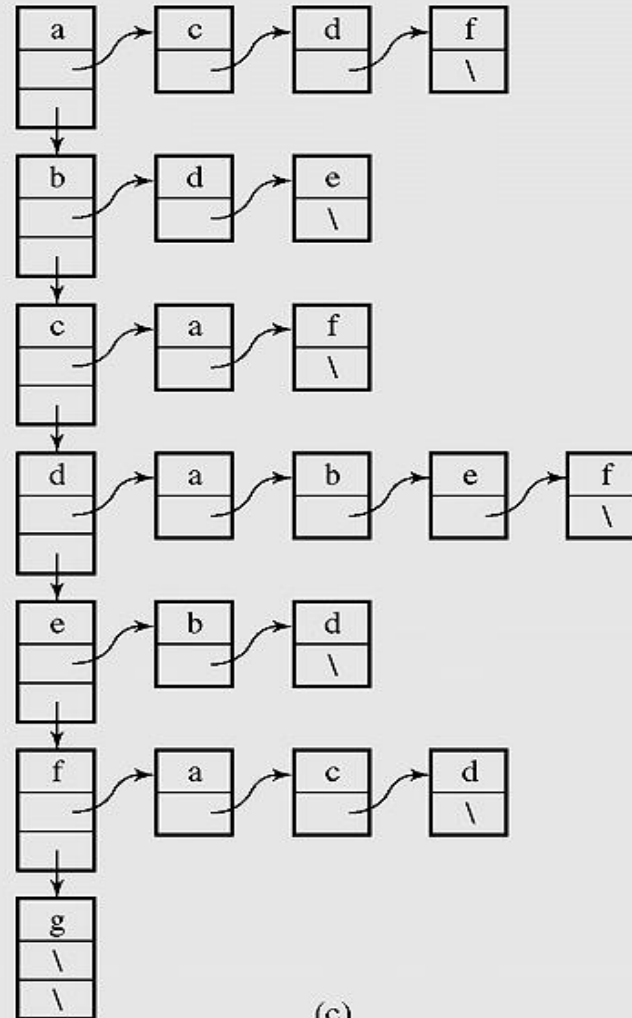


(a)

a	c	d	f	
b	d	e		
c	a	f		
d	a	b	e	f
e	b	d		
f	a	c	d	
g				

(b)

Graph representations.
A graph (a) can be represented as
(b–c) an adjacency list.



(c)

Biểu diễn đồ thị: Graph(**V**ertice, **E**dge)

- Adjacency Matrix - Ma trận kề
- Adjacency List – Danh sách kề
- Bộ nhớ đối với mỗi cách biểu diễn là bao nhiêu?

Biểu diễn đồ thị

- Adjacency Matrix - Ma trận kề: $O(V^2)$
- Adjacency List – Danh sách kề: $O(V+E)$

Bài tập

Bài 1: Vào 1 file matrix biểu diễn đồ thị graph.txt

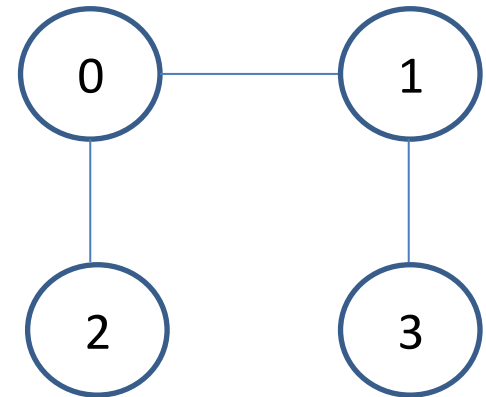
- dòng đầu là số đỉnh
- dòng sau là trọng số cạnh

a- Hãy đọc file đó và đổ dữ liệu vào mảng 2 chiều

b- Tính bậc của mỗi đỉnh

c- Tính tổng đường viền

d- DFS



Bài tập

Bài 1: Vào 1 file matrix biểu diễn đồ thị graph.txt

- dòng đầu là số đỉnh
- dòng sau là trọng số cạnh

a- Hãy đọc file đó và đổ dữ liệu vào mảng 2 chiều

b- Tính bậc của mỗi đỉnh

c- DFS(int i)

d- Tính tổng giá trị đường viền

graph.txt

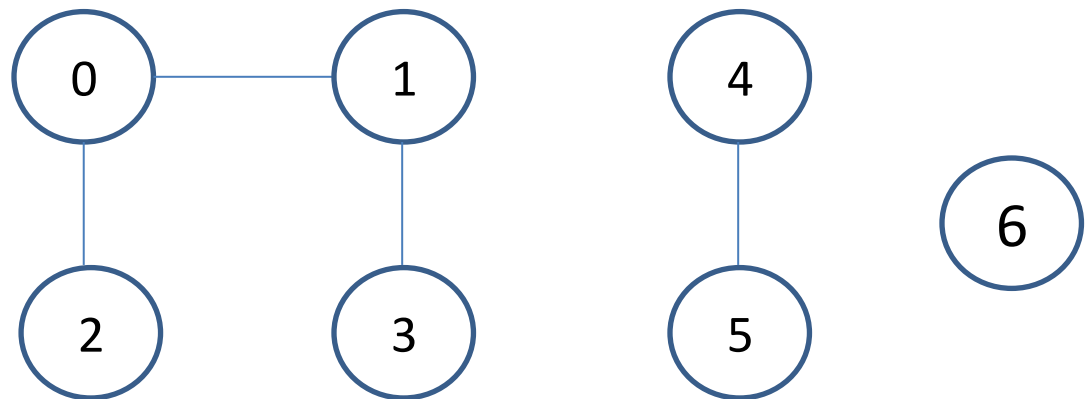
4

0 1 1 0

1 0 0 1

1 0 0 0

0 1 0 0



Example

```
int [][] a= {{0, 1, 1, 0},{0, 1, 1, 0},{0, 1, 1, 0},{0, 1, 1, 0}};  
    for (int i= 0; i<4; i++){  
        for (int j= 0; j<4; j++)  
            System.out.print(a[i][j] + " ");  
        System.out.println();  
    }
```

Exercise 1

```
class Graph{
    int V;
    Graph(){}
    Graph(int v) {this.V=v;
        //Khởi tạo mảng vuông 2 chiều V đỉnh
    }
    //1- loadFile("đường dẫn")
    //2- deg(int i){}
    //3- sumBound()
    //4- DFS(int i)
}
```

Exercise 2

Biểu diễn đồ thị bằng danh sách kề đỉnh-đỉnh Vào 1 file biểu diễn đồ thị đơn không trọng số graph.txt dòng đầu là số đỉnh dòng sau các đỉnh nối

Ví dụ:

4

0 1

0 2

1 3

a- Hãy đọc file đó

b- Tính bậc của mỗi đỉnh

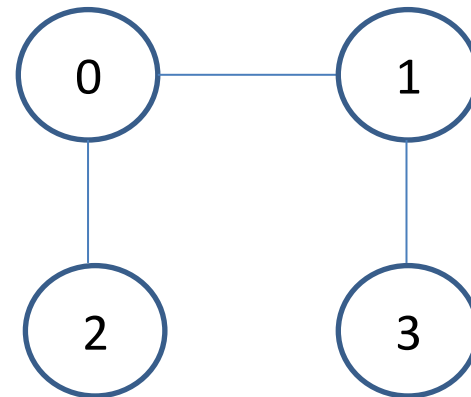
c- DFS (Duyệt theo chiều sâu)

[0] = [1,2]

[1] = [0,3]

[2] = [0]

[3] = [1]



Exercise 3

Biểu diễn đồ thị bằng danh sách kề đỉnh-đỉnh – trọng số Vào 1 file biểu diễn đồ thị đơn trọng số graph.txt dòng đầu là số đỉnh dòng sau các đỉnh nối và trọng số

Ví dụ:

4

0 1 3

0 2 5

1 3 7

a- Hãy đọc file đó

b- Tính bậc của mỗi đỉnh

c- DFS (Duyệt theo chiều sâu)

[0] = [[1,3],[2,5]]

[1] = [[0,3],[3,7]]

[2] = [[0,5]]

[3] = [[1,7]]

