BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

Toán rời rạc 2

Nội dung

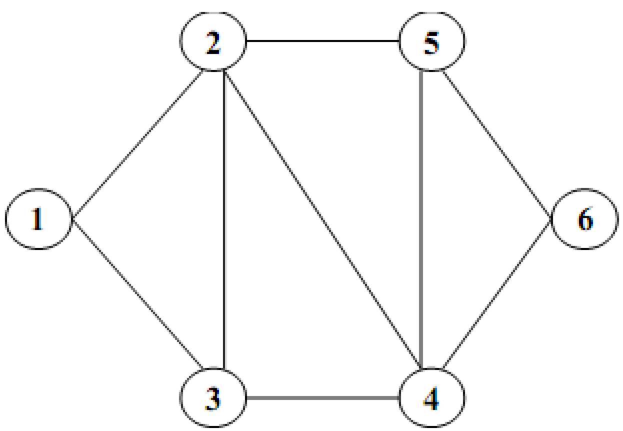
- Biểu diễn đồ thị bằng ma trận kề
- Biểu diễn đồ thị bằng ma trận liên thuộc
- Biểu diễn đồ thị bằng danh sách cạnh
- Biểu diễn đồ thị bằng danh sách kề

Biểu diễn đồ thị bằng ma trận kề

Ma trận kề của đồ thị vô hướng

- Xét đồ thị đơn vô hướng G =<V, E>, với tập đỉnh V = {1, 2, . . ., n}, tập cạnh E = {e1, e2,.., em}. Ta gọi ma trận kề của đồ thị G là ma trận có các phần tử hoặc bằng 0 hoặc bằng 1 theo qui định như sau:

$$A = \{\, a_{ij}: a_{ij} = 1 \text{ nếu } (i, j) \in E,\ a_{ij} = 0 \text{ nếu } (i,j) \notin E;\ i, j = 1, 2, \ldots, n \}$$



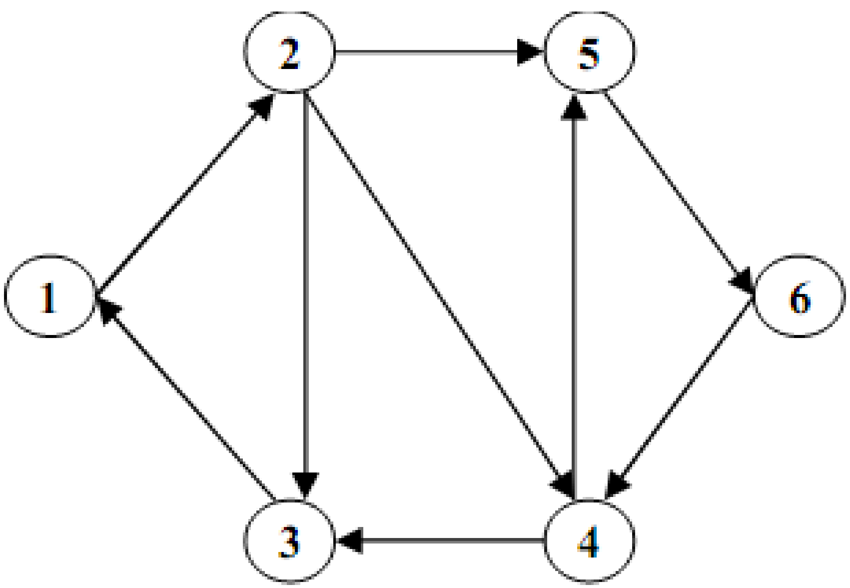
$$\begin{vmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{vmatrix}$$

Tính chất ma trận kề đối với đồ thị vô hướng

- Đối xứng qua đường chéo chính
- Tổng các phần tử của ma trận bằng hai lần số cạnh
  - $\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} = 2|E|$
- Tổng các phần tử của hàng u là bậc của đỉnh u
  - $\sum_{j=1}^{n} a_{uj} = \deg(u)$
- Tổng các phần tử của cột u là bậc của đỉnh u
  - $\sum_{i=1}^{n} a_{iu} = \deg(u)$
- Nếu ký hiệu a_{ij}^{p} $(i, j = 1,2, \dots, n)$ là các phần tử của ma trận $A^p = A.A\dots A$ (p lần), khi đó a_{ij}^{p} cho ta số đường đi khác nhau từ đỉnh i đến đỉnh j qua $p-1$ đỉnh trung gian

Ma trận kề của đồ thị có hướng

- Định nghĩa hoàn toàn tương tự với đồ thị vô hướng
  - Cần lưu ý tới hướng của cạnh
  - Ma trận kề của đồ thị có hướng là không đối xứng



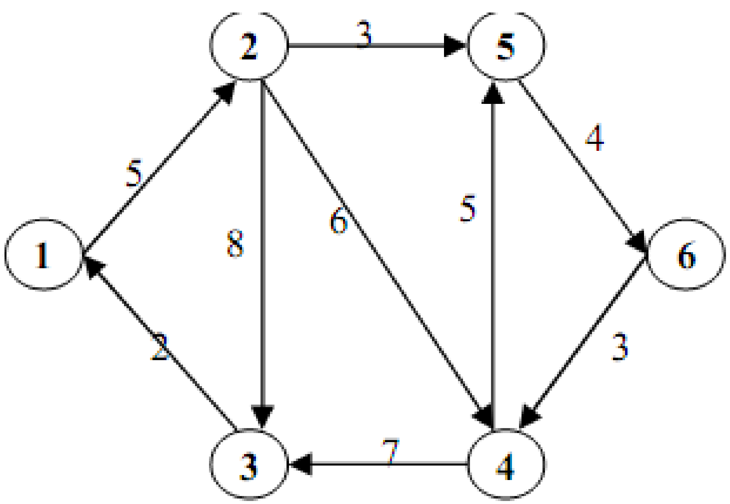
$$\begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{vmatrix}$$

Tính chất của ma trận kề của đồ thị có hướng

- Tổng các phần tử của ma trận bằng số cạnh
  - $\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} = |E|$
- Tổng các phần tử của hàng u là bán bậc ra của đỉnh u
  - $\sum_{j=1}^{n} a_{uj} = deg^{+}(u)$
- Tổng các phần tử của cột u là bán bậc vào của đỉnh u
  - $\sum_{i=1}^{n} a_{iu} = deg^{-}(u)$
- Nếu ký hiệu a_{ij}^{p} $(i, j = 1,2, \dots, n)$ là các phần tử của ma trận $A^p = A.A \dots A$ (p lần), khi đó a_{ij}^{p} cho ta số đường đi khác nhau từ đỉnh i đến đỉnh j qua $p - 1$ đỉnh trung gian

Ma trận trọng số

- Mỗi cạnh $e = (u, v)$ của đồ thị được gán bởi một số $c(e) = c(u, v)$ gọi là trọng số của cạnh e
  - Đồ thị trong trường hợp như vậy gọi là đồ thị trọng số
  - Ma trận trọng số $c = c[i,j]$, $c[i,j] = c(i,j)$ nếu $(i,j) \in E$, $c[i,j] = \theta$ nếu $(i,j) \notin E$. θ nhận các giá trị: $0, \infty, -\infty$ tuỳ theo từng tình huống cụ thể của thuật toán



$$\begin{vmatrix} \infty & 5 & \infty & \infty & \infty & \infty \\ \infty & \infty & 8 & 6 & 3 & \infty \\ 2 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 7 & \infty & 5 & \infty \\ \infty & \infty & \infty & \infty & \infty & 4 \\ \infty & \infty & \infty & 3 & \infty & \infty \end{vmatrix}$$

Ưu & nhược điểm của ma trận kề

- Ưu điểm
  - Đơn giản, dễ cài đặt trên máy tính
  - Sử dụng một mảng hai chiều để biểu diễn ma trận kề
  - Dễ dàng kiểm tra được hai đỉnh *u,v* có kề với nhau hay không
  - Đúng một phép so sánh (a[u][v]≠0?)
- Nhược điểm
  - Lãng phí bộ nhớ: bất kể số cạnh nhiều hay ít ta cần n^2 đơn vị bộ nhớ để biểu diễn
  - Không thể biểu diễn được với các đồ thị có số đỉnh lớn
  - Để xem xét đỉnh đỉnh *u* có những đỉnh kề nào cần mất *n* phép so sánh kể cả đỉnh *u* là đỉnh cô lập hoặc đỉnh treo

Qui ước khuôn dạng lưu trữ ma trận kề

- Dòng đầu tiên ghi lại số đỉnh của đồ thị
- N dòng kế tiếp ghi lại ma trận kề của đồ thị.
  – Hai phần tử khác nhau của ma trận kề được viết cách nhau một vài khoảng trống.

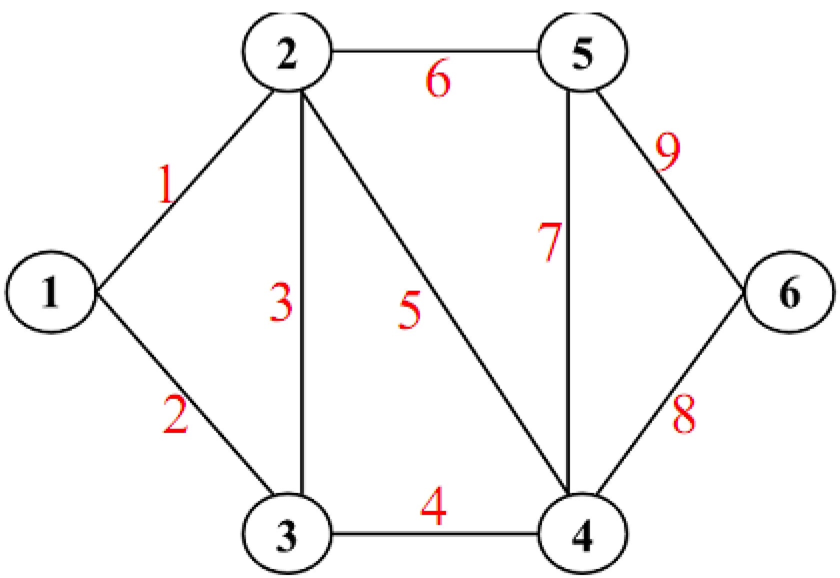
```
10
0    1    1    1    0    0    0    0    0    0
1    0    0    1    1    0    0    0    0    0
1    0    0    1    0    1    0    0    0    0
1    1    1    0    1    1    0    0    1    0
0    1    0    1    0    0    0    1    0    0
0    0    1    1    0    0    1    0    0    0
0    0    0    0    0    1    0    0    1    1
0    0    0    0    1    0    0    0    1    1
0    0    0    1    0    0    1    1    0    1
0    0    0    0    0    0    1    1    1    0
```

Biểu diễn đồ thị bằng ma trận liên thuộc

Ma trận liên thuộc: Đồ thị vô hướng

- Xét đồ thị vô hướng $G = (V, E), V = \{1,2, \dots, n\}, E = \{e_1, e_2, \dots, e_m\}$. Ma trận liên thuộc đỉnh-cạnh của G là ma trận kích thước $n \times m$ được xây dựng như sau:

$$a_{ij} = \begin{cases} 1, & \text{nếu đỉnh } i \text{ liên thuộc với cạnh } j \\ 0, & \text{nếu đỉnh } i \text{ không liên thuộc với cạnh } j \end{cases}$$

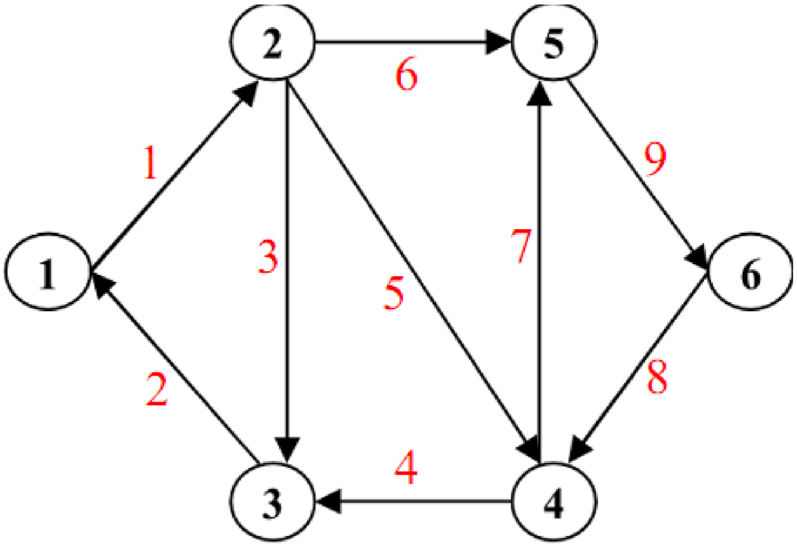


	1	2	3	4	5	6	7	8	9
1	1	1	0	0	0	0	0	0	0
2	1	0	1	0	1	1	0	0	0
3	0	1	1	1	0	0	0	0	0
4	0	0	0	1	1	0	1	1	0
5	0	0	0	0	0	1	1	0	1
6	0	0	0	0	0	0	0	1	1

Ma trận liên thuộc: Đồ thị có hướng

- Xét đồ thị có hướng $G = (V, E), V = \{1,2, \dots, n\}$, $E = \{e_1, e_2, \dots, e_m\}$. Ma trận liên thuộc đỉnh-cung của G là ma trận kích thước $n \times m$ được xây dựng như sau:

$$a_{ij} = \begin{cases} 1, \textit{nếu } i \textit{ là đỉnh đầu của cung } e_j \\ -1, \textit{nếu } i \textit{ là đỉnh cuối của cung } e_j \\ 0, \textit{nếu } i \textit{ không là đầu mút của cung } e_j \end{cases}$$



	1	2	3	4	5	6	7	8	9
1	1	-1	0	0	0	0	0	0	0
2	-1	0	1	0	1	1	0	0	0
3	0	1	-1	-1	0	0	0	0	0
4	0	0	0	1	-1	0	1	-1	0
5	0	0	0	0	0	-1	-1	0	1
6	0	0	0	0	0	0	0	1	-1

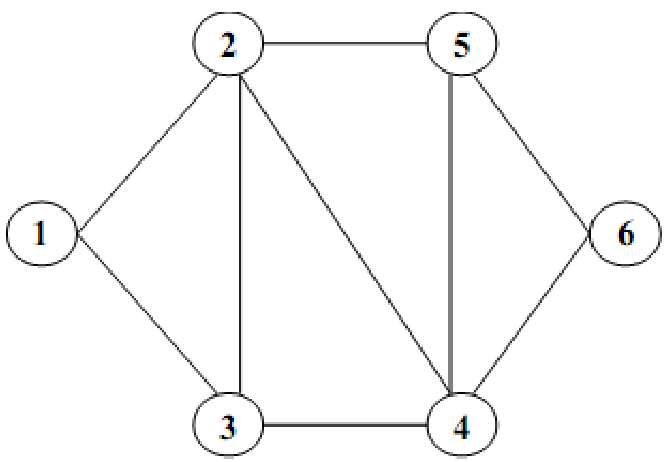
Biểu diễn đồ thị bằng danh sách cạnh

Danh sách cạnh (cung)

- Trong trường hợp đồ thị thưa (đồ thị có số cạnh $m < 6n$), người ta thường biểu diễn đồ thị dưới dạng danh sách cạnh.
  - Ta lưu trữ danh sách tất cả các cạnh (cung) của đồ thị vô hướng (có hướng). Mỗi cạnh (cung) e(x, y) được tương ứng với hai biến dau[e], cuoi[e].
  - Như vậy, để lưu trữ đồ thị, ta cần 2m đơn vị bộ nhớ.
  - Nhược điểm: để nhận biết những cạnh nào kề với cạnh nào chúng ta cần m phép so sánh trong khi duyệt qua tất cả m cạnh (cung) của đồ thị.
  - Nếu là đồ thị có trọng số, ta cần thêm m đơn vị bộ nhớ để lưu trữ trọng số của các cạnh.

Biểu diễn đồ thị vô hướng bằng danh sách cạnh

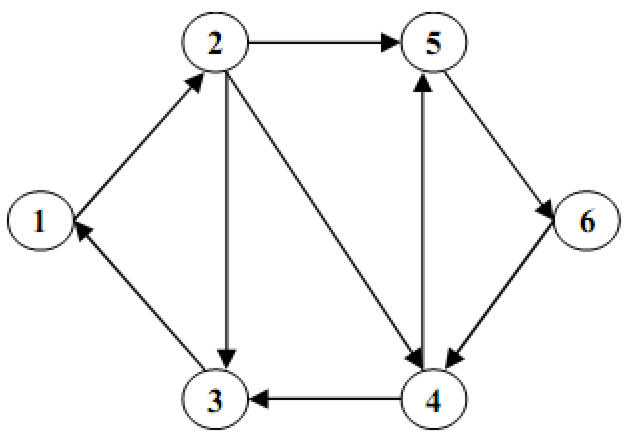
- Chỉ cần liệt kê các cạnh (u,v) mà không cần liệt kê cạnh (v,u).
- Nên liệt kê các cạnh theo thứ tự tăng dần của đỉnh đầu mỗi cạnh.
- Tính chất danh sách cạnh của đồ thị vô hướng:
  - Đỉnh đầu nhỏ hơn đỉnh cuối mỗi cạnh.
  - Số cạnh có giá trị u thuộc cả vế phải và vế trái của danh sách cạnh là bậc của đỉnh u.



Đỉnh đầu	Đỉnh cuối
1	2
1	3
2	3
2	4
2	5
3	4
4	5
4	6
5	6

Biểu diễn đồ thị có hướng bằng danh sách cạnh

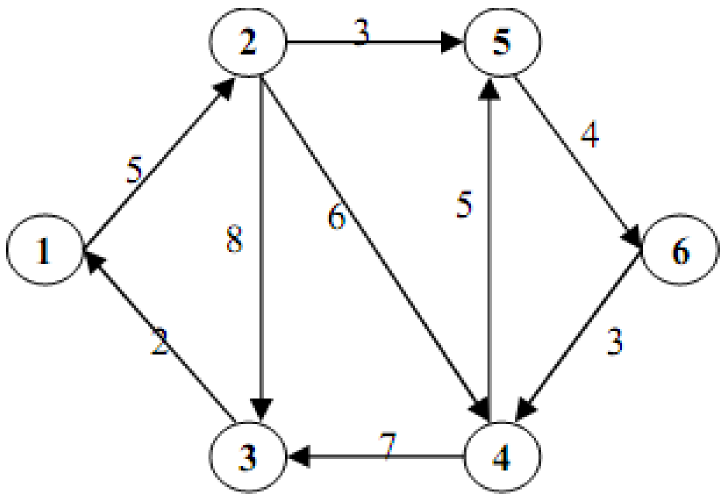
- Mỗi cạnh là bộ có tính đến thứ tự các đỉnh.
- Đặc biệt chú ý đến hướng của các cạnh
- Tính chất danh sách cạnh của đồ thị vô hướng:
  - Đỉnh đầu không nhất thiết phải nhỏ hơn đỉnh cuối mỗi cạnh.
  - Số cạnh có giá trị u thuộc cả vế phải các cạnh là $deg^+(u)$.
  - Số cạnh có giá trị u thuộc cả vế trái các cạnh là $deg^-(u)$.



Đỉnh đầu	Đỉnh Cuối
1	2
2	3
2	4
2	5
3	1
4	3
4	5
5	6
6	4

Biểu diễn đồ thị trọng số bằng danh sách cạnh

- Bổ sung thêm một cột là trọng số của mỗi cạnh



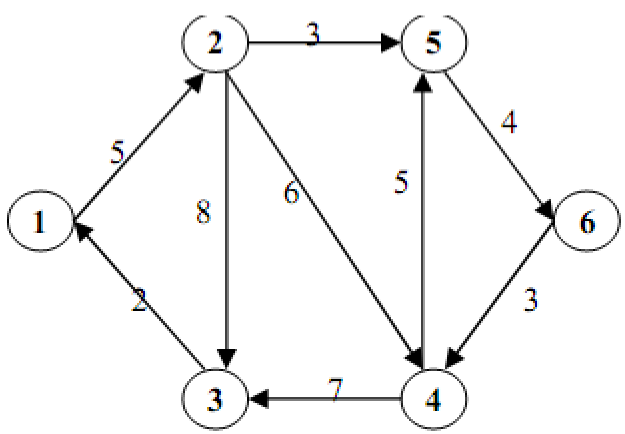
Đỉnh đầu	Đỉnh Cuối	Trọng Số
1	2	5
2	3	8
2	4	6
2	5	3
3	1	2
4	3	7
4	5	5
5	6	4
6	4	3

Ưu & nhược điểm của danh sách cạnh

- Ưu điểm của danh sách cạnh:
  – Trong trường hợp đồ thị thưa ($m<6n$), biểu diễn bằng danh sách cạnh tiết kiệm được không gian nhớ;
  – Thuận lợi cho một số thuật toán chỉ quan tâm đến các cạnh của đồ thị.
- Nhược điểm của danh sách cạnh:
  – Khi cần duyệt các đỉnh kề với đỉnh u bắt buộc phải duyệt tất cả các cạnh của đồ thị.
    - Điều này làm cho thuật toán có chi phí tính toán cao.

Khuôn dạng lưu trữ danh sách cạnh

- Dòng đầu tiên ghi lại số N, M tương ứng với số đỉnh và số cạnh của đồ thị.
  – Hai số được viết cánh nhau một vài khoảng trống;
- M dòng kế tiếp, mỗi dòng gi lại một cạnh của đồ thị
  – Đỉnh đầu và đỉnh cuối mỗi cạnh được viết cách nhau một vài khoảng trống.



dothi.in

```
6    9
1    2    5
2    3    8
2    4    6
2    5    3
3    1    2
4    3    7
4    5    5
5    6    4
6    4    3
```

Cấu trúc dữ liệu biểu diễn danh sách cạnh (1/4)

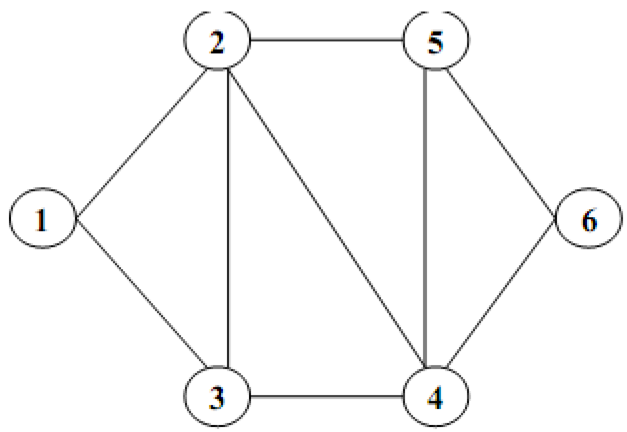
- Biểu diễn danh danh sách cạnh của đồ thị bằng mảng:

```
//Định nghĩa một cạnh của đồ thị
typedef struct {
  int dau;
  int  cuoi;
} Edge;

//Danh sách các cạnh được biểu diễn trong mảng G
Edge G[MAX];
```

Cấu trúc dữ liệu biểu diễn danh sách cạnh (2/4)

- Ví dụ:



Đỉnh đầu	Đỉnh cuối
1	2
1	3
2	3
2	4
2	5
3	4
4	5
4	6
5	6

Cạnh:	G[1]	G[2]	G[3]	G[4]	G[5]	G[6]	G[7]	G[8]	G[9]
G[i].dau	1	1	2	2	2	3	4	4	5
G[i].cuoi	2	3	3	4	5	4	5	6	6

Cấu trúc dữ liệu biểu diễn danh sách cạnh (3/4)

```
//Định nghĩa một cạnh có trọng số của đồ thị
typedef struct {
  int dau;
  int  cuoi;
  int trongso;
} Edge;

//Danh sách các cạnh được biểu diễn trong mảng G
Edge G[MAX];
```

Cấu trúc dữ liệu biểu diễn danh sách cạnh (4/4)

- Biểu diễn danh danh sách cạnh của đồ thị bằng danh sách liên kết:

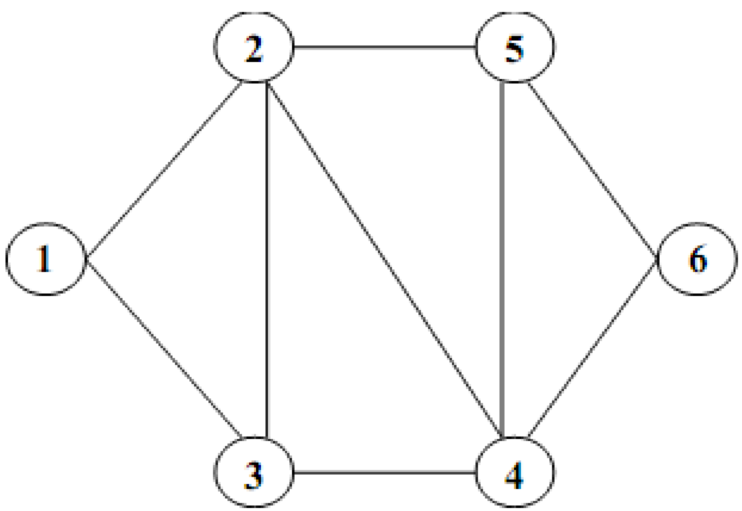
```
typedef struct canh{  //Định nghĩa một cạnh của đồ thị
    int  dau;
    int   cuoi;
    struct canh *next;
} *Edge;
Edge *G; //Các cạnh được biểu diễn bằng danh danh sách liên kết G.
```

Biểu diễn đồ thị bằng danh sách kề

Danh sách kề

- Với mỗi đỉnh u của đồ thị chúng ta lưu trữ danh sách các đỉnh kề với nó mà ta ký hiệu là Ke(u), nghĩa là

$$Ke(u) = \{\ v \in V: (u,\ v) \in E\}$$



Ke(1) = { 2, 3).

Ke(2) = {1, 3, 4, 5}.

Ke(3) = {1, 2, 4}.

Ke(4) = {2, 3, 5, 6}.

Ke(5) = {2, 4, 6}.

Ke(6) = { 4, 5}.

Ưu & nhược điểm của danh sách kề

- Ưu điểm của danh sách kề:
  - Dễ dàng duyệt tất cả các đỉnh của một danh sách kề;
  - Dễ dàng duyệt các cạnh của đồ thị trong mỗi danh sách kề;
  - Tối ưu về phương pháp biểu diễn.
- Nhược điểm của danh sách kề:
  - Khó khăn cho người đọc có kỹ năng lập trình yếu.

Biểu diễn danh sách kề dựa vào mảng

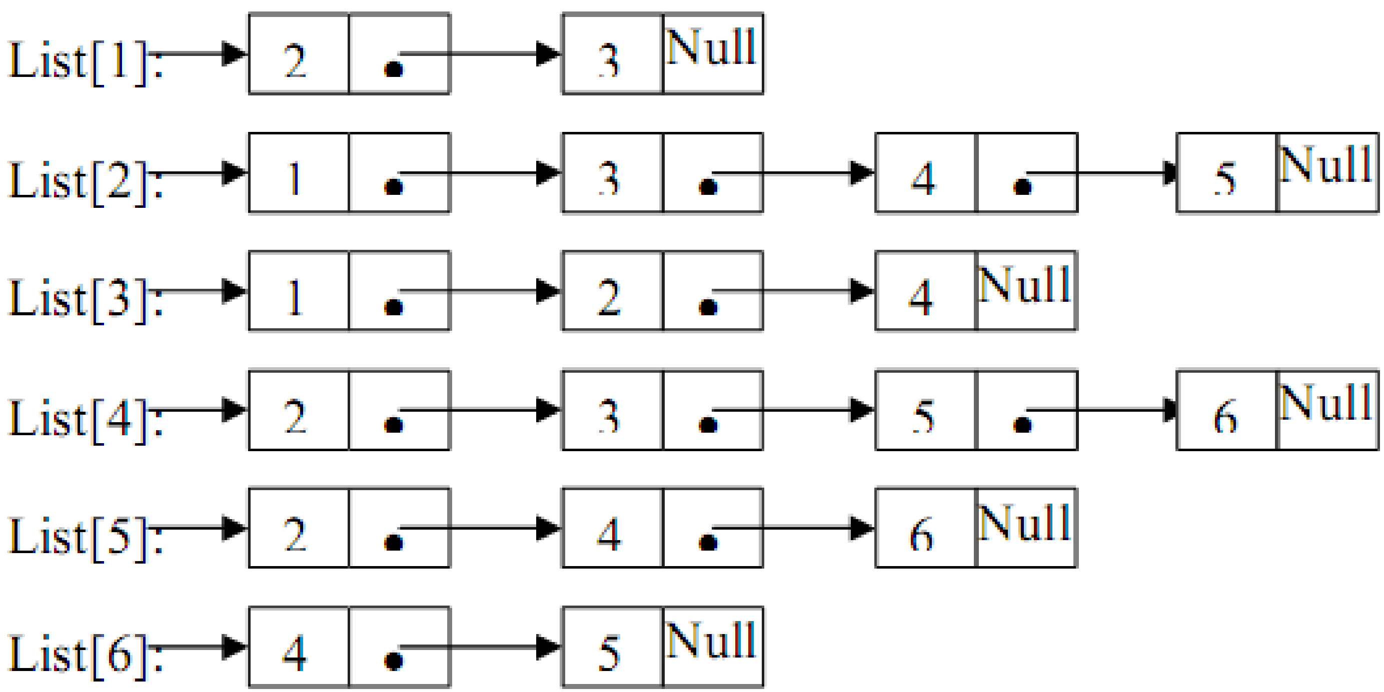
- Mảng được chia thành n đoạn
  - Đoạn thứ i trong mảng lưu trữ danh sách kề của đỉnh thứ $i \in V$.
  - Để biết một đoạn thuộc mảng bắt đầu từ phần tử nào đến phần tử nào ta sử dụng một mảng khác dùng để lưu trữ vị trí các phần tử bắt đầu và kết thúc của đoạn.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A[i]=?	2	3	1	3	4	5	1	2	4	2	3	5	6	2	4	6	4	5
	Đoạn 1		Đoạn 2				Đoạn 3			Đoạn 4				Đoạn 5			Đoạn 6	

VT[6] = {0, 2, 6, 9, 13, 16, 18}

Biểu diễn danh sách kề bằng danh sách liên kết

- Với mỗi đỉnh u∈V, ta biểu diễn mỗi danh sách kề của đỉnh bằng một danh sách liên kết List(u).



Khuôn dạng lưu trữ danh sách kề

- Dòng đầu tiên ghi lại số đỉnh của đồ thị
  - N dòng kế tiếp ghi lại danh sách kề của đỉnh tương ứng theo khuôn dạng:
- Phần tử đầu tiên là vị trí kết thúc của đoạn, tiếp đến là danh sách các đỉnh của danh sách kề
  - Các phần tử được ghi cách nhau một vài khoảng trống

<u>dothi.in</u>

6				
2	2	3		
6	1	3	4	5
9	1	2	4	
13	2	3	5	6
16	2	4	6	
18	4	5		

Bài tập 1

- Trong một buổi gặp mặt, một số khách mời bắt tay với một số khách mời khác. Chứng minh rằng tổng số lượt bắt tay của tất cả các khách mời là số chẵn.

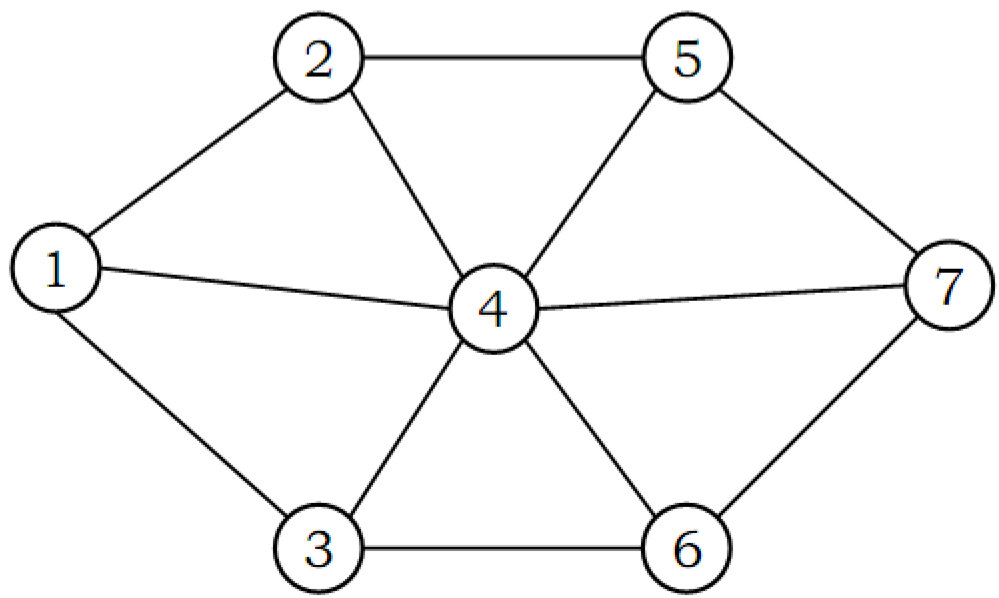
Bài tập 2

- Một đơn đồ thị vô hướng với n đỉnh có nhiều nhất là bao nhiêu cạnh?

Bài tập 3

Hãy biểu diễn đồ thị vô hướng dưới đây dưới dạng:

1) Ma trận kề
2) Danh sách cạnh
3) Danh sách kề



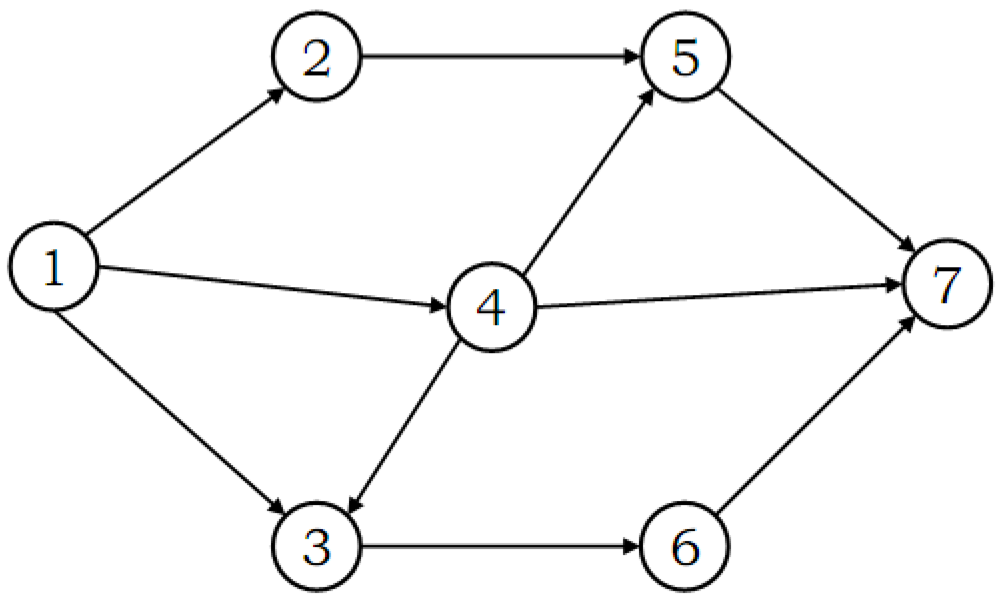
Bài tập 4

Hãy biểu diễn đồ thị có hướng dưới đây dưới dạng:

1) Ma trận kề

2) Danh sách cạnh

3) Danh sách kề



Bài tập 5

Hãy biểu diễn đồ thị có hướng dưới đây dưới dạng:

1) Ma trận kề

2) Danh sách cạnh