

BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

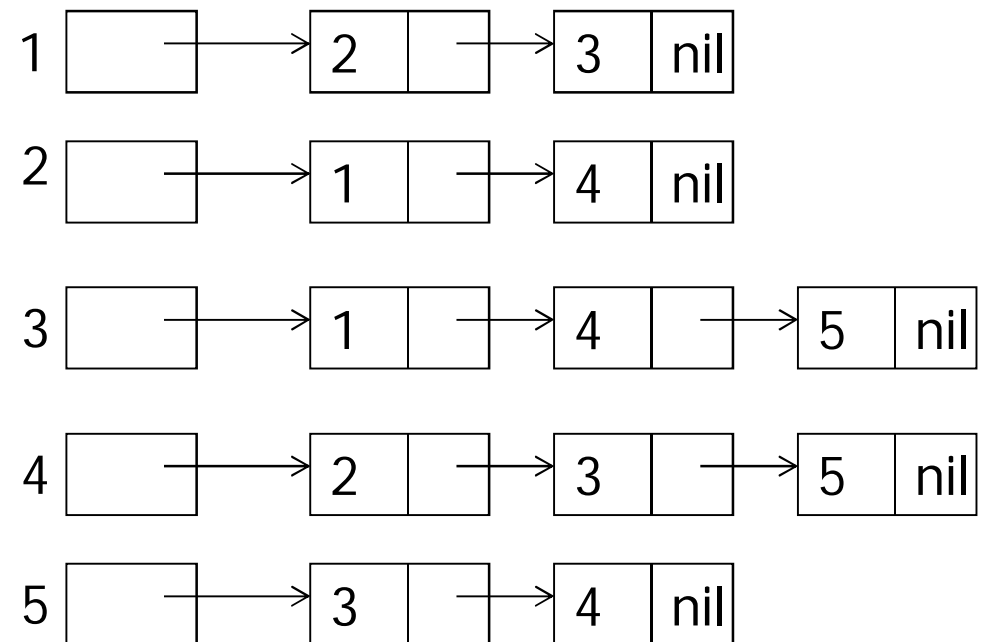
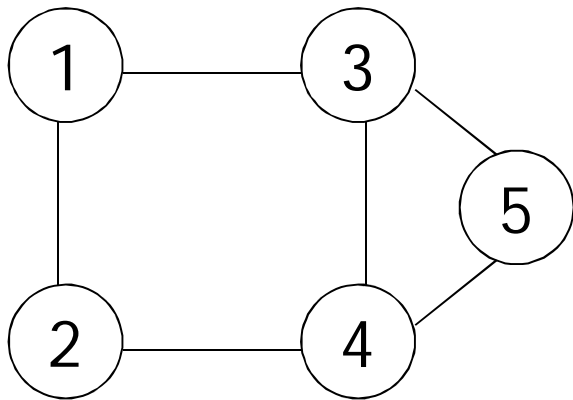
- Danh sách kề
- Ma trận kề

DANH SÁCH KẼ

- Danh sách kề của đỉnh u : $\text{Adj}(u) = \{v \in V \mid (u, v) \in E\}$
- Có thể biểu diễn đồ thị $G = (V, E)$ như một tập các danh sách kề bằng cách lưu trữ mỗi đỉnh $u \in V$ cùng với danh sách các đỉnh kề với u

DANH SÁCH KẼ

Ví dụ 1: Đồ thị vô hướng



DANH SÁCH KẼ C++

```
typedef struct node *LIST;
    struct node{
        int v;
        LIST next;
    };
typedef LIST GRAPH[maxv]; //maxv là số đỉnh tối đa
GRAPH G;
```

DANH SÁCH KẼ C++

```
void Initialize(GRAPH &G)
{
    int i,x,y,v;
    LIST t;
    cout<<"given number of edges and vertices of graph:";
    cin>>m>>n;
    for(i=1;i<=n;i++) G[i]=NULL;
    for(i=1;i<=m;i++)
    {
        cout<<"given edge "<<i<<"(x, y):"; cin>>x>>y;
        t=new(node); t->v=x; t->next=G[y]; G[y]=t;
        t=new(node); t->v=y; t->next=G[x]; G[x]=t;
    }
}
```

MA TRẬN KỀ

- Cho đơn đồ thị $G = (V, E)$, với tập đỉnh $V = \{1, 2, \dots, n\}$, ma trận kề của G là

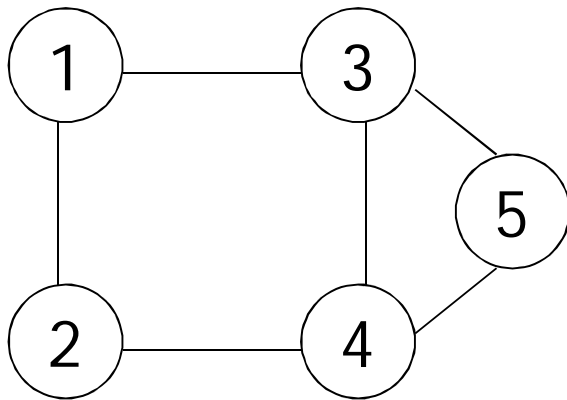
$$A = \{a_{ij} \mid i, j = 1, 2, \dots, n\}, \text{ } a_{ij} = 0 \text{ nếu } (i, j) \notin E \text{ và } a_{ij} = 1 \text{ nếu } (i, j) \in E$$

- Nếu G là đa đồ thị thì

$$a_{ij} = 0 \text{ nếu } (i, j) \notin E \text{ và } a_{ij} = k \text{ nếu có } k \text{ cạnh nối hai đỉnh } i \text{ và } j$$

MA TRẬN KẼ

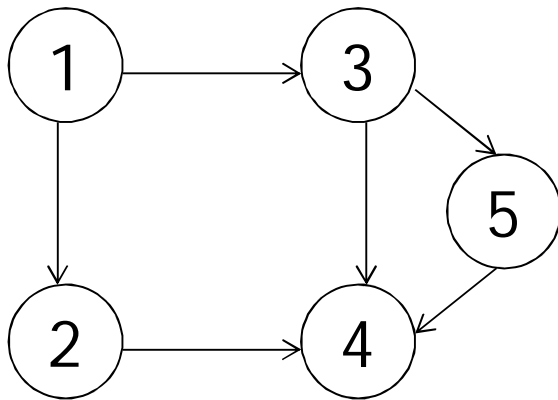
Ví dụ 1: Đồ thị vô hướng



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

MA TRẬN KẼ

Ví dụ 2: Đồ thị có hướng



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

MA TRẬN KỀ

- Ma trận kề của đồ thị vô hướng **đối xứng**

$$a_{ij} = a_{ji}, i, j = 1, 2, 3, \dots, n$$

- Tổng các phần tử trên dòng i (cột j) của ma trận kề là **bậc của đỉnh i (đỉnh j)**

MA TRẬN KẼ C++

```
typedef struct GRAPH {  
    int numv;  
    int A[MAX][MAX];  
};  
GRAPH G;
```

MA TRẬN KẼ C++

```
void Initialize(GRAPH &g) // undirected simple graph
{
    cout<<"Given number of vertices of graph:"; cin>>g.numv;
    cout<<"Enter adjacent matrix of graph:\n";
    for(int i=1;i<=g.numv;i++)
    {
        for(int j=i+1;j<=g.numv;j++)
        {
            cout<<"A["<<i<<","<<j<<"]="; cin>>g.A[i][j];
            g.A[j][i]=g.A[i][j];
        }
        g.A[i][i]=0; cout<<"\n";
    }
}
```

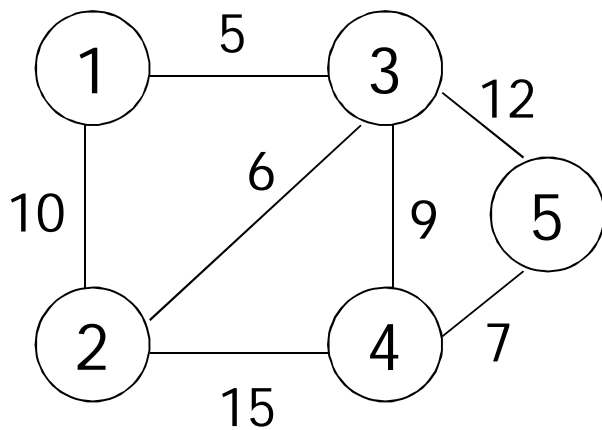
MA TRẬN KỀ

- Đồ thị **có trọng số** (weighted graph) là đồ thị mà mỗi cạnh (i, j) được **gán một số thực $w(i, j)$**
- Một đồ thị có trọng số với n đỉnh có thể được biểu diễn bởi ma trận trọng số

$C = \{c_{ij}: i, j = 1, 2, \dots, n\}$, trong đó $c_{ij} = w(i, j)$ nếu có cạnh (i, j) và $c_{ij} = 0, \infty$, hoặc $-\infty$ nếu không có cạnh (i, j)

MA TRẬN KỀ

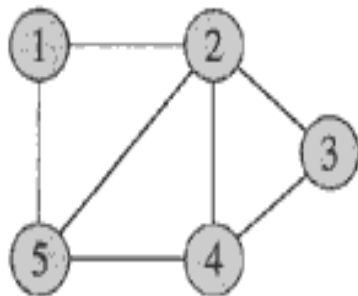
Ví dụ 3: Ma trận trọng số của đồ thị vô hướng



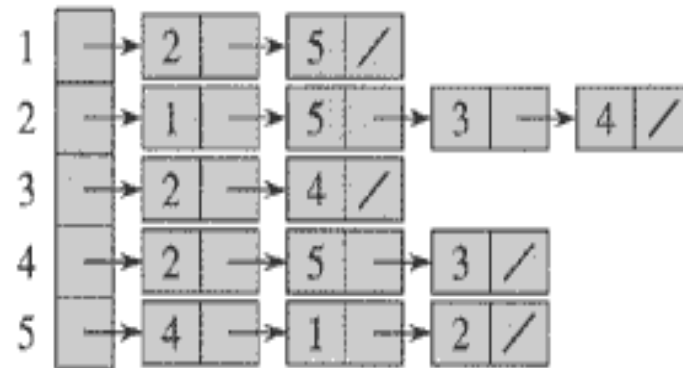
$$\begin{bmatrix} 0 & 10 & 5 & 0 & 0 \\ 10 & 0 & 6 & 15 & 0 \\ 5 & 6 & 0 & 9 & 12 \\ 0 & 15 & 9 & 0 & 7 \\ 0 & 0 & 12 & 7 & 0 \end{bmatrix}$$

SO SÁNH CÁC CÁCH BIỂU DIỄN

Biểu diễn đồ thị vô hướng bằng danh sách và ma trận



(a)



(b)

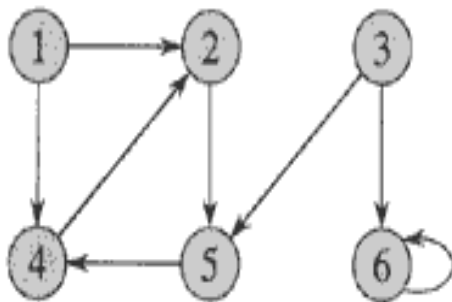
	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

(c)

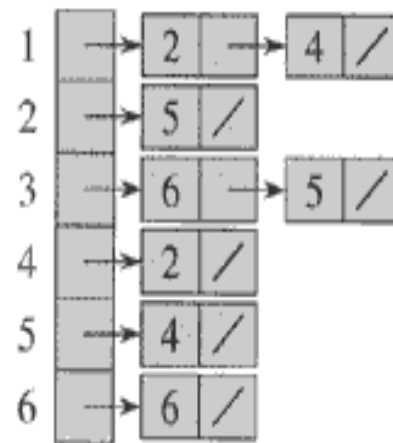
Figure 22.1 Two representations of an undirected graph. (a) An undirected graph G having five vertices and seven edges. (b) An adjacency-list representation of G . (c) The adjacency-matrix representation of G .

SO SÁNH CÁC CÁCH BIỂU DIỄN

Biểu diễn đồ thị có hướng bằng danh sách và ma trận



(a)



(b)

	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	1	0	0	0	0
5	0	0	0	1	0	0
6	0	0	0	0	0	1

(c)

Figure 22.2 Two representations of a directed graph. (a) A directed graph G having six vertices and eight edges. (b) An adjacency-list representation of G . (c) The adjacency-matrix representation of G .

SO SÁNH CÁC CÁCH BIỂU DIỄN

- Chi phí bộ nhớ cho ma trận là $O(|V|^2)$ và cho danh sách là $O(|V| + 2|E|)$
- Chi phí xử lý khi dùng ma trận là $O(1)$ và khi dùng danh sách là $O(|V|)$