



石家莊鐵道大學
SHIJIAZHUANG TIEDAO UNIVERSITY

在线开放课程

数据结构— 图

图的存储结构

主讲：石玉晶

目录



在线开放课程

- ◆ 1、邻接矩阵表示法
- ◆ 2、使用邻接矩阵创建无向网
- ◆ 3、有向图的邻接矩阵表示
- ◆ 4、邻接矩阵表示法总结

如何存储图?

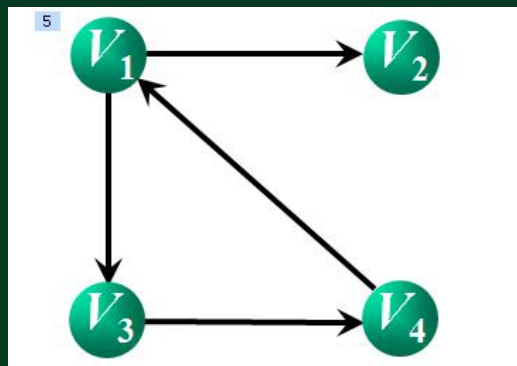
考虑图的定义，图是由顶点和边组成的，分别考虑如何存储顶点、如何存储边。

一、邻接矩阵表示法

- 将图的**顶点**信息存储在一个**一维数组**中；
- 并将它的**边/弧**存储在一个**二维数组**中即构成邻接矩阵。

$$A[i][j] = \begin{cases} 1, & \text{若}(v_i, v_j) \text{或} \langle v_i, v_j \rangle \in E(G) \\ 0, & \text{其它} \end{cases}$$

一、邻接矩阵表示法



$$\text{arc} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$\text{vexs} = \{ '1', '2', '3', '4' \}$

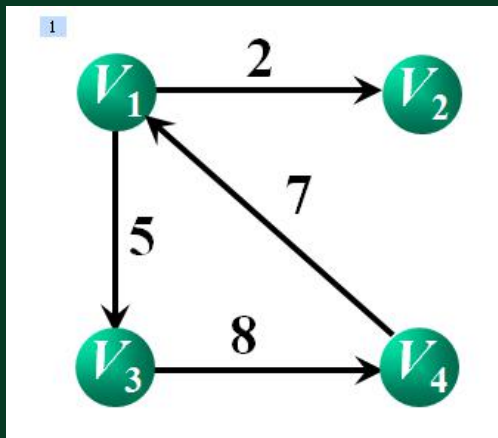
一、邻接矩阵表示法

- **网图**的邻接矩阵可定义为：

$$A[i][j] = \begin{cases} w_{ij} & \text{若 } (v_i, v_j) \in E \text{ (或 } \langle v_i, v_j \rangle \in E) \\ \infty & \text{其他} \end{cases}$$

一、邻接矩阵表示法

- 网图的邻接矩阵可定义为：



2

$$\text{arc} = \begin{pmatrix} \infty & 2 & 5 & \infty \\ \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 8 \\ 7 & \infty & \infty & \infty \end{pmatrix}$$

一、邻接矩阵表示法

图的数组(邻接矩阵)存储表示

```
#define MaxInt 32767;    // 最大值∞
#define MVNum 100;    // 最大顶点个数
typedef char VertexType;    // 设顶点数据类型为字符型
typedef int ArcType;    // 边的权值为整型
typedef struct {
    VertexType vexs[MVNum];    // 顶点表
    ArcType arcs[MVNum][MVNum];    // 邻接矩阵
    int vexnum, arcnum;    // 图的当前顶点数和
} MGraph;
```


二、使用邻接矩阵创建无向网

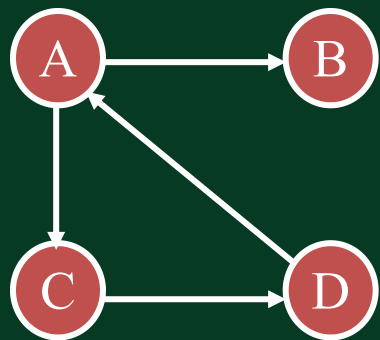
- **【算法思想】**

- (1) 输入总顶点数和总边数。
- (2) 依次输入点的信息存入顶点表中。
- (3) 初始化邻接矩阵，使每个权值初始化为极大值。
- (4) 构造邻接矩阵。

二、使用邻接矩阵创建无向网

```
Status CreateUDN(MGraph &G) {
    cin>>G.vexnum>>G.arcnum;    //依次输入顶点个数和边个数
    for (i = 0; i<G.vexnum; ++i) //依次输入顶点的信息
        cin>>G.vexs[i];
    /* 以下代码为创建边*/
    for (i = 0; i<G.vexnum; ++i) //初始化邻接矩阵
        for (j = 0; j<G.vexnum; ++j)
            G.arcs[i][j] = MaxInt;
    for (k = 0; k<G.arcnum; ++k) { //构造邻接矩阵
        cin>>v1>>v2>>w;
        i = LocateVex(G, v1); j = LocateVex(G, v2);
        G.arcs[i][j] =G.arcs[j][i] = w;
    } //for
    return OK;
}
```

三、有向图的邻接矩阵表示



G1

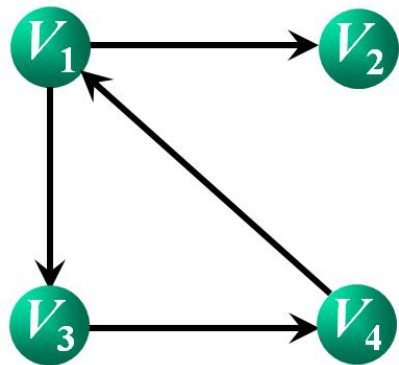
G1.vexnum=4

G1.arcnum=4

G1.vexs=['A','B','C','D']

$$G1.arcs = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

三、有向图的邻接矩阵表示



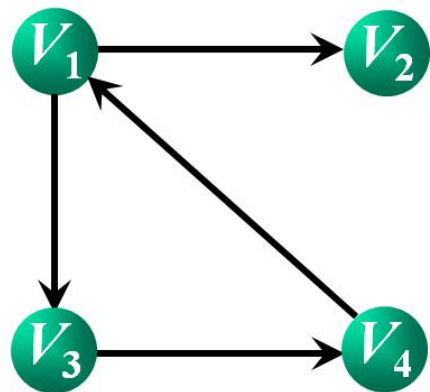
vertex= $V_1 \ V_2 \ V_3 \ V_4$

arc=
$$\begin{matrix} & \begin{matrix} V_1 & V_2 & V_3 & V_4 \end{matrix} \\ \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

① 如何求顶点 i 的出度？

邻接矩阵的第 i 行元素之和。

三、有向图的邻接矩阵表示



vertex=

V_1	V_2	V_3	V_4
-------	-------	-------	-------

arc=

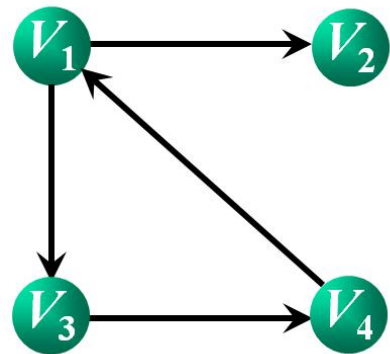
	V_1	V_2	V_3	V_4	
0	1	1	0	0	V_1
0	0	0	0	0	V_2
0	0	0	1	0	V_3
1	0	0	0	0	V_4



如何求顶点 i 的入度？

邻接矩阵的第 i 列元素之和。

三、有向图的邻接矩阵表示



vertex= V_1 V_2 V_3 V_4

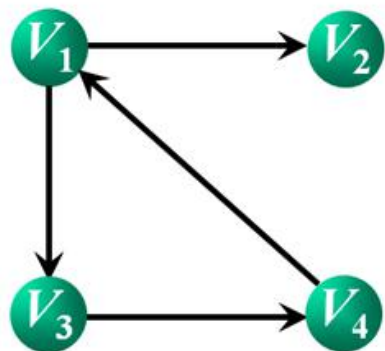
arc=

	V_1	V_2	V_3	V_4	
0	1	1	0		V_1
0	0	0	0		V_2
0	0	0	1		V_3
1	0	0	0		V_4

❓ 如何判断从顶点 i 到顶点 j 是否存在边？

测试邻接矩阵中元素 $\text{arc}[i][j]$ 是否为 1。

三、有向图的邻接矩阵表示




vertex=

V_1	V_2	V_3	V_4
-------	-------	-------	-------

arc=

	V_1	V_2	V_3	V_4	
}	0	1	1	0	V_1
	0	0	0	0	V_2
	0	0	0	1	V_3
	1	0	0	0	V_4

 如何求顶点 i 的所有邻接点

将数组中第 i 行元素扫描一遍，若 $\text{arc}[i][j]$ 为 1，则顶点 j 为顶点 i 的邻接点。

四、邻接矩阵表示法总结

❖ 特点:

- 无向图的邻接矩阵对称，可压缩存储；有 n 个顶点的无向图需存储空间为 $n(n+1)/2$ 。
- 有向图邻接矩阵不一定对称；有 n 个顶点的有向图需存储空间为 n^2 。
- 无向图中顶点 V_i 的度 $TD(V_i)$ 是邻接矩阵 A 中第 i 行元素之和。
- 有向图中，
 - ✓ 顶点 V_i 的出度是 A 中第 i 行元素之和。
 - ✓ 顶点 V_i 的入度是 A 中第 i 列元素之和。

❖ 邻接矩阵的优缺点

- **优点:** 容易判定顶点间有无边（弧）和计算顶点的度（出度、入度）。
- **缺点:** 边数较少时，空间浪费较大。

小结



在线开放课程

- 图的存储结构——邻接矩阵表示法
- 使用邻接矩阵表示图时的若干算法

小结



在线开放课程

- 接下来学习：
- 图的存储结构—邻接表

谢谢！