第 1 章　基本概念

1.1　图论概述

离散数学是以研究离散量的结构和相互之间的关系为主要目标，其研究对象一般是有限个或可数个元素。离散数学充分契合了计算机科学的特点。离散数学是计算机科学重要的基础理论之一。
离散数学主要包括四个方面: 数理逻辑、集合论、图论、代数机构。

图论 [Graph Theory]是数学的一个分支，它以图为研究对象。

世界上各事物之间, 自然界内诸现象之间经常存在着某些必然的联系, 需要人们通过研究分析，去揭示这些关系。

人们常把事物、现象用结点表示，用有向或无向的边来表示它们之间的联系。这就构成了图论中所讨论的图。

历史上图论曾经被好多位数学家各自独立地建立过。关于图论的文字记载最早出现在欧拉 1736 年论著中，其原始问题有很强的实际背景。18 世纪在哥尼斯堡城（今俄罗斯加里宁格勒）的普莱格尔河上有 7 座桥[1]，将河中的岛屿和河岸连结。

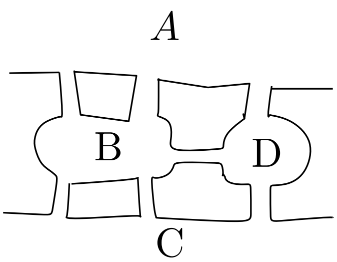


图 1.1: 哥尼斯堡 7 桥

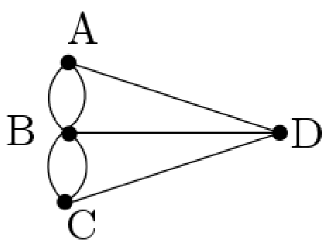


图 1.2: 哥尼斯堡 7 桥简化

[1]在第二章欧拉回路中介绍

早期的图论与数学游戏有密切的联系: 周游世界问题、渡河问题、三家三井问题...20 世纪后，图论的应用渗透到其它学科领域，如物理、化学、运筹学、博弈论、计算机网络、社会学、语言学等等。对于基础图论来说，不要求事先掌握高深的数学工具，只需要有集合论和线性代数的基本概念，即可进行学习。

1.2 图的基本概念及定义

定义 1.2.1. 二元组 $G = (V(G), E(G))$ 称为图。其中 $V(G)$ 是非空集，称为结点集；$E(G)$ 为 $V(G)$ 各结点之间边的集合，称为边集。

常用 $G = (V, E)$ 表示图。
当 V,E 都是有限集合时，称 G 为有限图。
当 V 或 E 是无限集合时，称 G 为无限图。
一般情况下，给定 $G = V, E$, 如不加特殊说明，认为 $V = \{v_1, v_2, v_3, \cdots, v_n\}$, $E = \{e_1, e_2, e_3, \cdots, e_m\}$, 即结点数 $|V| = n$，边数 $|E| = m$。

若图中的边为有向的，则称为有向图。
若图中的边为无向的，则称为无向图。
若图中既有有向边，又有无向边，则称为混合图。

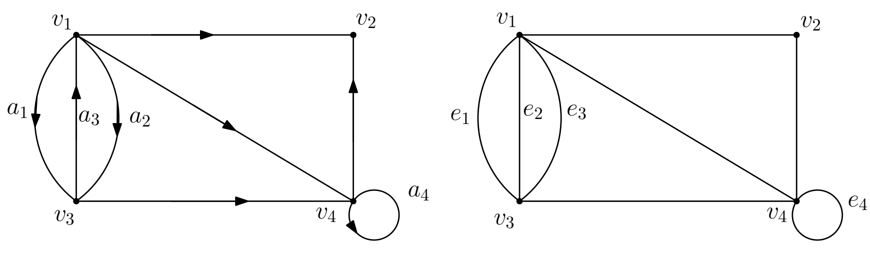


图 1.3

图的边可用 $e_k = (v_i, v_j)$ 表示

1. 称 v_i 与 v_j 是相邻结点
2. 称 e_k 分别与 v_i, v_j 相关联

3. 如果 e_k 是有向边，称 v_i 是 e_k 的始点,v_j 是 e_k 的终点，并称为 v_i 是 v_j 的直接前驱,v_j 是 v_i 的直接后继

4. 如果 e_k 是无向边, 称 v_i, v_j 是 e_k 的两个端点

定义 1.2.2. 只与一个结点相关联的边称为自环; 在同一对结点之间可以存在多条边，称为重边; 含有重边的图叫多重图。

定义 1.2.3. $G = (V, E)$ 的某结点所关联的边数称为该结点的度, 用 $d(v)$ 表示。如果 V 带有自环，则自环对 $d(v)$ 的贡献为 2。
有向图中:

- 以结点 V 为始点的边数目称为 V 的正度，记为 $d^+(v)$

- 以结点 V 为终点的边数目称为 V 的负度，记为 $d^-(v)$

显然，有 $d^+(v) + d^-(v) = d(v)$

定义 1.2.4. 任意两结点间最多只有一条边，且不存在自环的无向图称为简单图。
没有任何边的简单图叫空图，记为 N_n。
任何两结点间都有边的简单图称为完全图记为 K_n。(K_n 中每个结点的度为 *n-1*)

性质 1.2.1. 设 $G = (V, E)$ 有 n 个结点，m 条边，则

$$\sum_{v \in V(G)} (d_v) = 2m$$

性质 1.2.2. G 中度为奇数的结点必为偶数个。

性质 1.2.3. 有向图 G 中正度之和等于负度之和。

性质 1.2.4. K_n 中的边数为 $\frac{1}{2}n(n-1)$。

性质 1.2.5. 非空简单图中一定存在度相同的结点。

定义 1.2.5. 如果图 $G = (V, E)$ 的每条边 $e_k = (v_i, v_j)$ 都赋予一个实数 W_k 做为该边的权, 则称 G 为赋权图。如果这些权都是正实数，就称 G 为正权图。

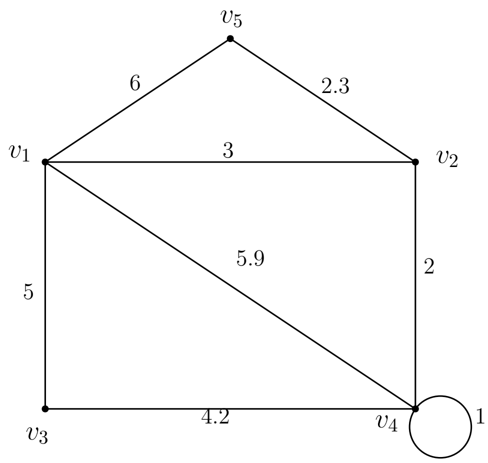


图 1.4

定义 1.2.6. 给定 $G = (V, E)$, 如果存在另一个图 $G' = (V', E')$，满足 V' 包含于 V, 满足 E' 包含于 E，则称 G' 是 G 的一个子图。
如果 $V' = V$，就称 G' 是 G 的支撑子图。
如果 V' 包含于 V，且 E' 包含了在结点子集 V' 之间的所有边，则称 G' 是 G 的导出子图。

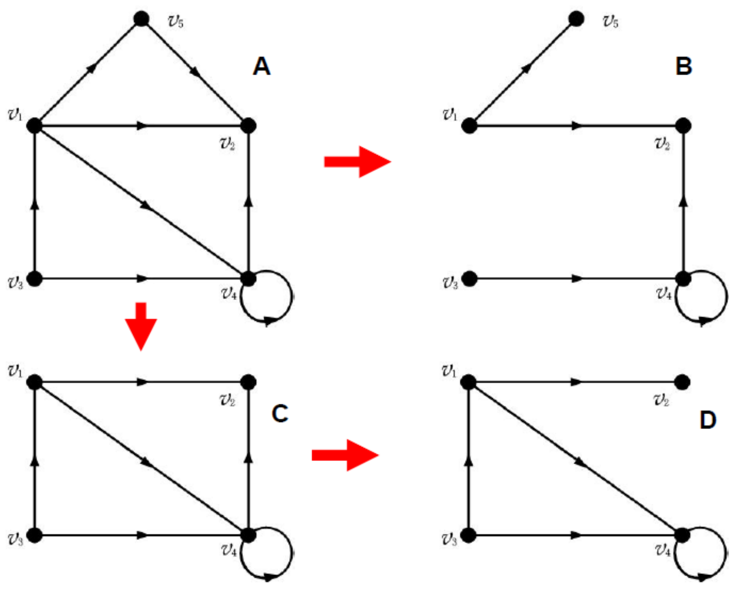


图 1.5: B: 支撑子图 C: 导出子图 D: 子图

推论 1.2.1. 显然，根据上述的定义，图 G 是自身的子图，支撑子图，导出子图。

空图是图 G 的支撑子图。

定义 1.2.7. 称原图 G 和空图都是图 G 的平凡子图。

定义 1.2.8. 给定两个图 $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$。令

$$G_1 \bigcup G_2 = (V_1 \bigcup V_2, E_1 \bigcup E_2) \tag{1.1}$$
$$G_1 \bigcap G_2 = (V_1 \bigcap V_2, E_1 \bigcap E_2) \tag{1.2}$$
$$G_1 \bigoplus G_2 = (V_1 \bigoplus V_2, E_1 \bigoplus E_2) \tag{1.3}$$

1.1, 1.2, 1.3 分别称为 G_1 和 G_2 的并、交、对称差。

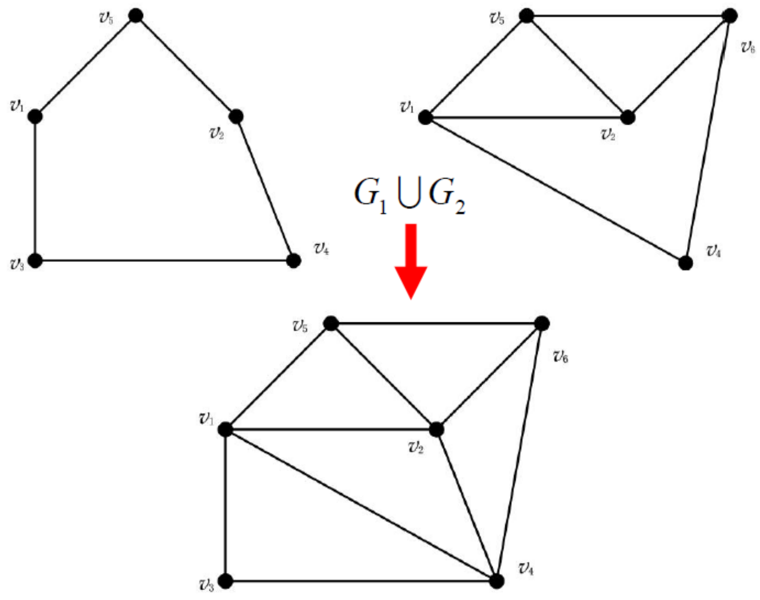


图 1.6: $G_1 \bigcup G_2$

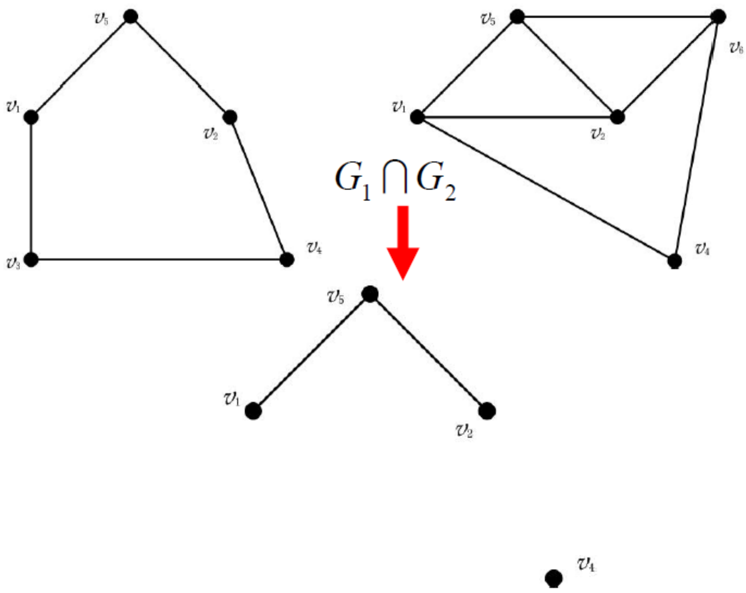


图 1.7: $G_1 \bigcap G_2$

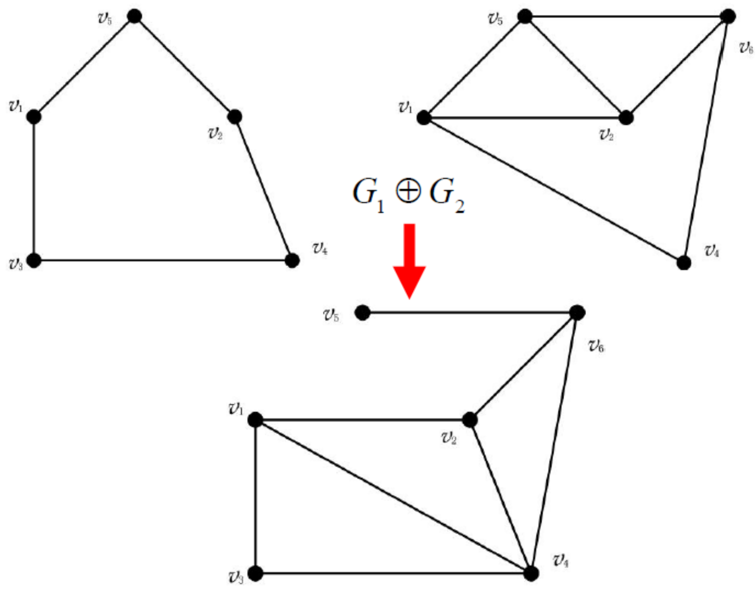


图 1.8: $G_1 \bigoplus G_2$

在图 G 中删掉一个子图 H，指删掉 H 中的各条边，记为$G - H$。对于简单图 G, 称 $K_n - G$ 为 G 的补图，记做$\bar{G}$。

从 G 中删去某个结点 v 及其关联的边所得到的图记做$G-v$。
从 G 中删掉某条特定的边 e, 记做$G-e$。
显然，$G-v$ 是图 G 的导出子图,$G-e$ 是图 G 的支撑子图。

定义 1.2.9. 如 G 为无向图，则

$$\Gamma(v)=\{u|(v,u)\in E\}$$

称为 v 的邻点集。
如 G 为有向图，v 是其中一个结点，则

$$\Gamma^{+}(v)=\{u|(v,u)\in E\}$$

称为 v 的直接后继或者外邻集; 相应的

$$\Gamma^{-}(v)=\{u|(v,u)\in E\}$$

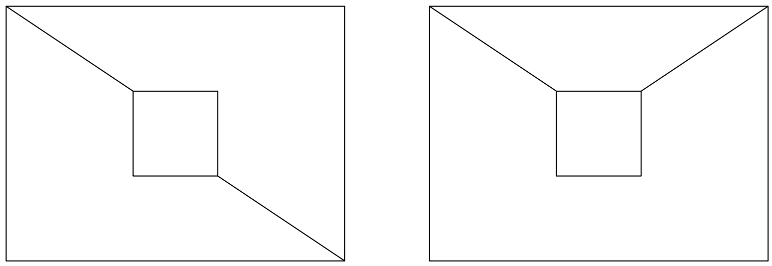
称为 v 的直接前趋集或内邻集。

定义 1.2.10. 两个图 $G_1=(V_1,E_1),G_2=(V_2,E_2)$, 如果 V_1,V_2 之间存在双射 f，而且 $(u,v)\in E_1$，当且仅当 $(f(u),f(v))\in E_2$ 时，称 G_1 和 G_2同构。记做

$$G_1\cong G_2$$

从定义知，若 $G_1\cong G_2$，必须满足:
(1)$|V(G_1)|=|V(G_2)|$,$|E(G_1)|=|E(G_2)|$
(2)G_1 和 G_2 结点度的非增序列相同
(3)G_1 和 G_2 存在同构的导出子图

思考: 如何判定两图同构?



1.3 图的代数表示

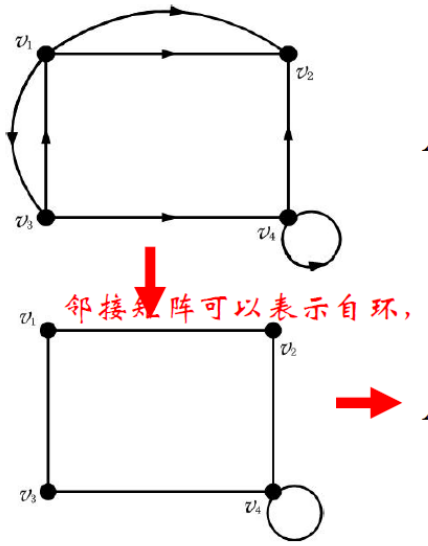
图在计算机中如何表示 (拓扑结构、边权值...)？如何对图进行描述或运算？所以我们需要用代数的方法来描述图!

定义 1.3.1. 表示了结点间的邻接关系的矩阵称为邻接矩阵。

有向图的邻接矩阵 **A** 是一个 n 阶方针，其元素为:

$$\mathbf{A} = [a_{ij}]_{n\times n} \quad a_{ij} = \begin{cases} 1 & (v_i, v_j) \in E \\ 0 & others \end{cases}$$

关联矩阵:关联矩阵表示结点与边之间的关联关系。



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

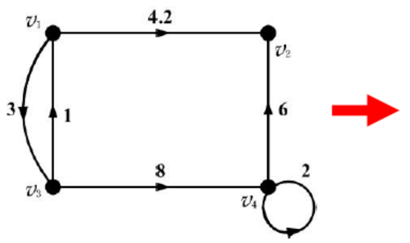
$$A=\begin{bmatrix}0&1&1&0\\1&0&0&1\\1&0&0&1\\0&1&1&1\end{bmatrix}$$

图 1.9

权矩阵:赋权图常用权矩阵 A 进行表示。

其元素为:

$$\mathbf{A}=[a_{ij}]_{n\times n}\quad a_{ij}=\begin{cases}w_{ij} & (v_i,v_j)\in E\\ 0 & others\end{cases}$$



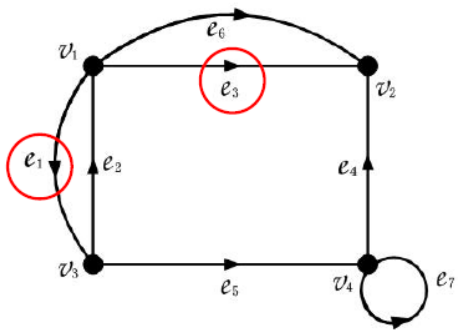
$$A=\begin{bmatrix}0&4.2&3&0\\0&0&0&0\\1&0&0&8\\0&6&0&2\end{bmatrix}$$

图 1.10

关联矩阵:关联矩阵表示结点与边之间的关联关系。

有向图 G 的关联矩阵 $\mathbf{B}$ 是 $n \times m$ 的矩阵，当给定结点和边的编号之后，其元素

$$\mathbf{B} = [b_{ij}]_{n\times m} \quad b_{ij} = \begin{cases} 1 & e_j = (v_i, v_k) \in E \\ -1 & e_j = (v_k, v_i) \in E \\ 0 & others \end{cases}$$

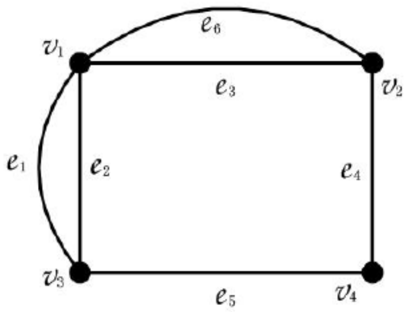


$$B = \begin{array}{cc} \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{bmatrix} 1 & -1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & -1 & 0 & -1 & 0 \\ -1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & ? \end{bmatrix} \\ & \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{matrix} \end{array}$$

图 1.11

无向图 G 的关联矩阵 $\mathbf{B}$ 是 $n \times m$ 的矩阵，当给定结点和边的编号之后，其元素

$$\mathbf{B} = [b_{ij}]_{n\times m} \quad b_{ij} = \begin{cases} 1 & e_j\text{与}v_i\text{关联} \\ 0 & others \end{cases}$$



$$B=\begin{array}{cc} \begin{matrix} v_1\\ v_2\\ v_3\\ v_4 \end{matrix} & \begin{bmatrix} 1&1&1&0&0&1\\ 0&0&1&1&0&1\\ 1&1&0&0&1&0\\ 0&0&0&1&1&0 \end{bmatrix}\\ & \begin{matrix} e_1&e_2&e_3&e_4&e_5&e_6 \end{matrix} \end{array}$$

图 1.12

关联矩阵的性质（有向图）：
(1) 每列只有两个非零元：1、-1
(2) 第 i 行非零元的数目恰为结点 v_i 的度，其中 1 的数目为其正度，-1 的数目为其负度
关联矩阵的性质（无向图）：
(1) 每列只有一个非零元：1
(2) 第 i 行 1 的数目恰为结点 v_i 的度
关联矩阵能够表示重边，但不能表示自环。

邻接矩阵、权矩阵、关联矩阵可以表示重边，但无法表示自环

邻接矩阵、权矩阵、关联矩阵的优点是一旦写出代数表达式，则可得到确定图，且非常直观。缺点是 (1) 不能表示自环 (2) 在计算机上存储邻接矩阵与关联矩阵时，将占据较大的存储空间并可能增加计算复杂度。所以需要引入**边列表、正向表、逆向表、邻接表**。

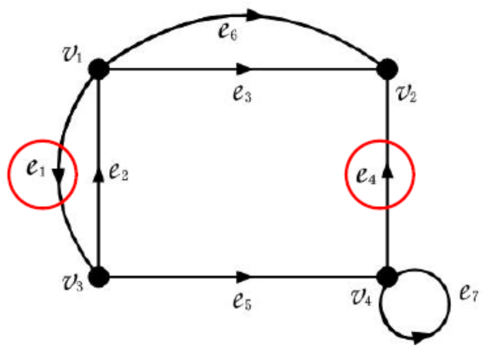
边列表由两个 m 维向量 A 和 B 组成。若 $e_k=(v_i,v_j)$, 则

$$A(k)=i, B(k)=j$$

如果 G 为赋权图，则再增加一个 m 维变量 Z，若 e_k 的权值为 w_k, 则令

$$Z(k)=w_k$$

如图 1.13 所示是边列表，边列表的实质是关联矩阵的压缩形式并克服了其缺点。



$$\begin{array}{l}
A:\begin{pmatrix}1 & 3 & 1 & 4 & 3 & 1 & 4\end{pmatrix}\\
B:\begin{pmatrix}3 & 1 & 2 & 2 & 4 & 2 & 4\end{pmatrix}\\
\quad\;\; \begin{matrix}e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7\end{matrix}
\end{array}$$

图 1.13

赋权图只需增加权值向量，如果 G 是赋权图，则再增加一个 m 维向量 Z，若 e_k 的权是 w_k, 则令 $Z(k)=w_k$。例如图 1.14 的边列表形式是：

$$A:(4\quad 4\quad 1\quad 2\quad 2\quad 2\quad 4) \tag{1.4}$$

$$B:(1\quad 1\quad 2\quad 2\quad 4\quad 3\quad 3) \tag{1.5}$$

$$Z:(5\quad 3\quad 4\quad 6\quad 7\quad 2\quad 4) \tag{1.6}$$

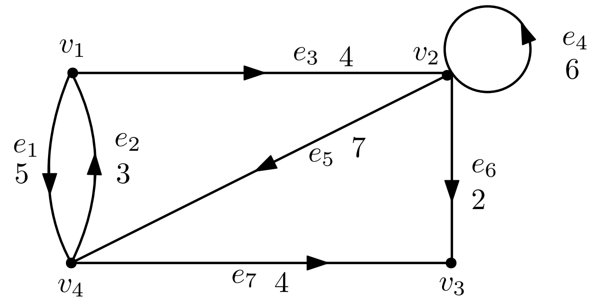


图 1.14

类似的可以得到无向图的边列表。

正向表: 当对 G 的结点与边进行编号后，正向表将每个节点的直接后继集中在一起存放。

有向图的正向表由一个 $(n+1)$ 维向量 A, 一个 m 维向量 B 组成。$A(i)$ 表示结点 v_i 的第一个后继在 B 中的地址，B 中存放这些后继结点的编号，$A(n+1)=m+1$。

对赋权图，用 m 维向量 Z 存放权，$Z(k)=w_k$。

有向图正向表如图 1.15 所示:

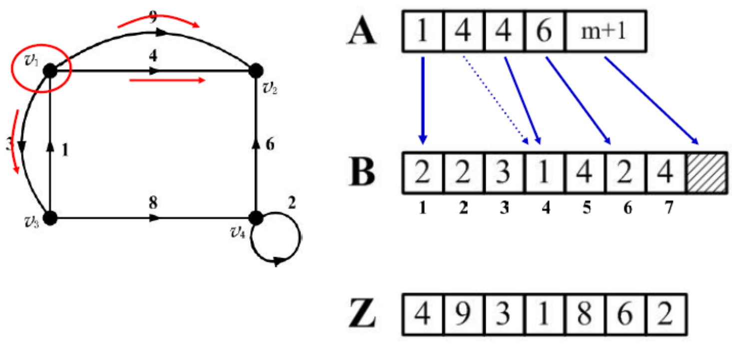


图 1.15

正向表存在下述关系:

1.$d^+(v_i)=A(i+1)-A(i)$

2. $A(i)=\sum_{j=1}^{i-1} d^+(v_j)+1$

3. 从 $B(A(i))$ 到 $B(A(i+1)-1)$ 的任意一个值，都是 v_i 的直接后继。

无向图的正向表结构:

B 向量中存放的是相应邻结点的编号，因此为 2m 维。相应地，Z 向量也变成 2m 维。

无向图正向表如图 1.16 所示:

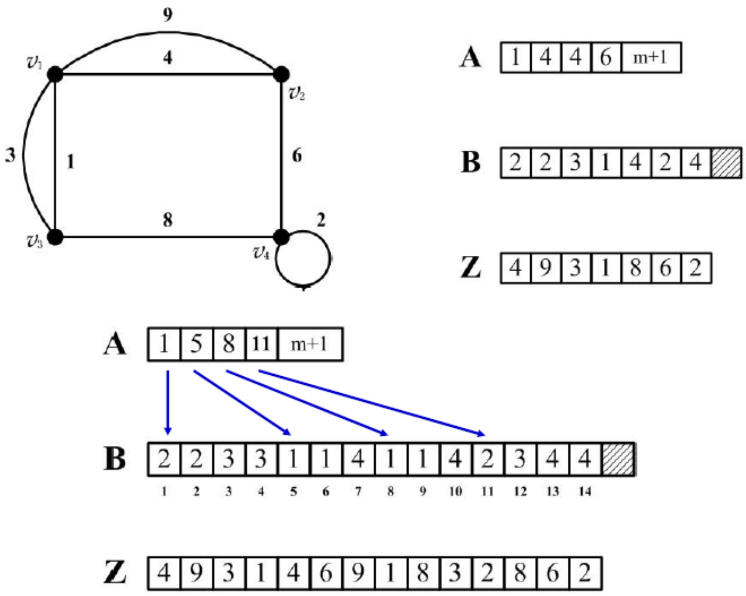


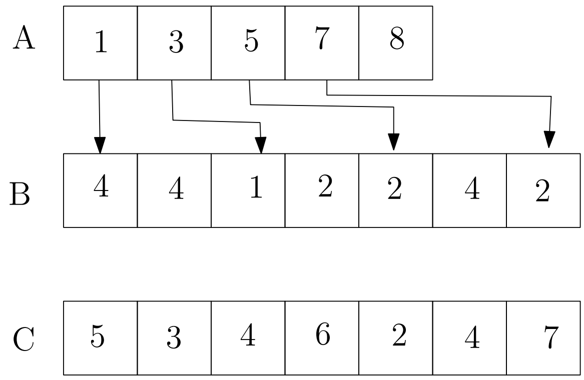
图 1.16

逆向表：与正向表相反，逆向表是将每个结点的直接前趋集中在一起存放。

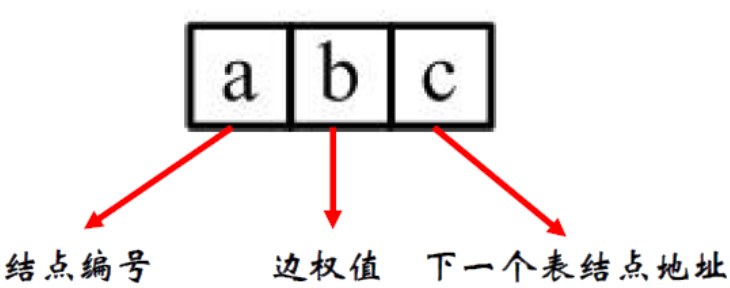
逆向表实质是对有向图邻接矩阵的列进行压缩的结果。

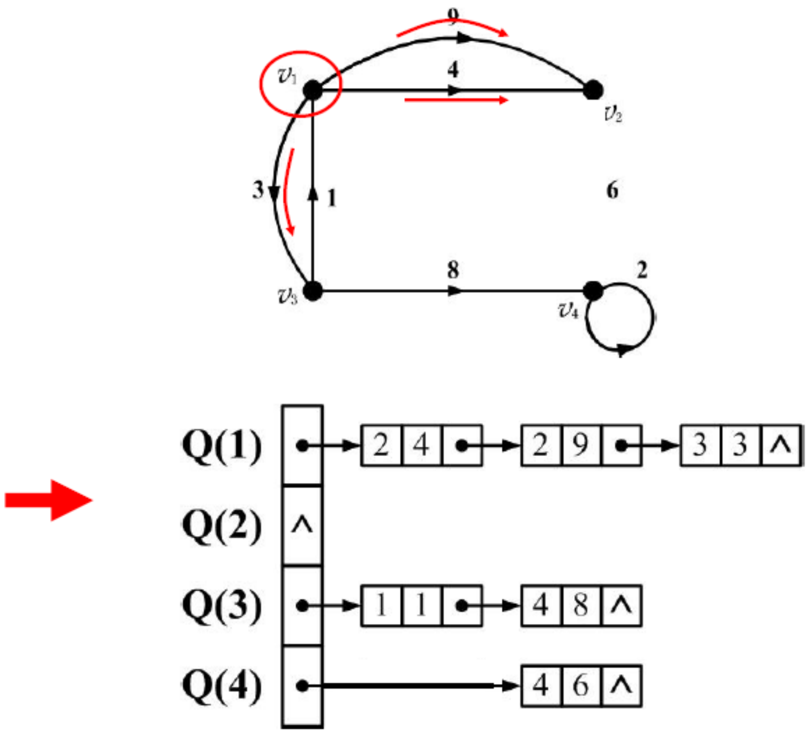
思考：正（逆向表优缺点是什么?）

例如图 1.14 的逆向表是:



邻接表：采用单链表结构表示一个图。其中基本单元为表结点，如下所示：





9
4
v_1
v_2
3
1
6
8
2
v_3
v_4
Q(1)
2 4
2 9
3 3 ∧
Q(2)
∧
Q(3)
1 1
4 8 ∧
Q(4)
4 6 ∧