第一章 图的基本概念

提起图论，大家会很自然地想起四色问题，七桥问题等一些有趣的问题。作为一个学科来讲，图论起源于 1736 年欧拉（Euler）的第一篇图论论文，该文解决了哥尼斯堡的七桥问题；1936 年，科尼格（Konig）出版第一本图论书籍，在以后几十年中，图论在理论和应用方面逐步发展起来，特别是近 40 多年来，由于计算机的广泛应用，图论的发展更是迅速。图论给含有二元关系的系统提供了数学模型，因而它在许多科学领域中具有越来越重要的地位。

数据结构课程中也涉及图论的某些内容，但其侧重点在数据结构和图的一些算法。本课程则是要较为系统地介绍图的基础知识和一些比较典型的应用。

§1.1 图的概念

在集合论部分，我们已经知道，离散对象之间的关系可以用关系图来表示，这种图就是我们下面要定义的有向图。另外，我们所讲的图还包括无向图，为此，我们来看下面两个例子，以作为我们引入的概念所来自的背景。

例 1.1 （循环赛问题）A、B、C、D 4 支足球队进行循环赛，为了解当前各队的胜负情况，可以用结点表示球队，用有向边 (u,v) 表示球队 u 胜球队 v，如图 1.1 所示。此图表示的信息是，A 胜 B、C、D，B 胜 C，D 胜 C，而 B 和 D 还没有进行比赛。

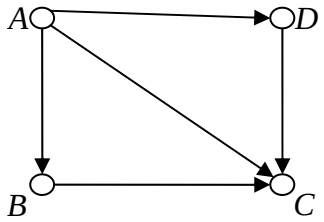

图 1.1

例 1.2 （交通问题）为说明 A，B，C，D，E 四个城市间的铁路交通状况，可用结点表示城市，用无向边 (u,v) 表示城市 u 与城市 v 间有直达列车。如图 1.2 所示，这是五个城市之间的一种列车调度情况。

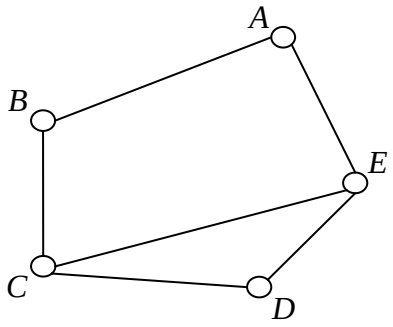

图 1.2

由结点和边组成的对象就称为图，它表示了某种二元关系。

大家知道，二元关系是用有序对来表示的。如果二元关系是对称的，则该关系可简单地用无序对（即(a,b)和(b,a)表示同一个对象）来表示。如例 1.2 中，两个城市之间有直达列车就可以用无序对来表示。

针对这种情况，我们引入无序积的概念。

定义 1.1 设A和B是两个非空集合，所有$a\in A$和$b\in B$所组成的无序对构成的集合，称为A和B的无序积，记作$A\&B$。无序积$A\&B$的任意一个子集称为A和B的一个二元关系，当$A=B$时，A和B的二元关系称为A上的二元关系。

注：虽然$A\times A$和$A\&A$的子集都称为A上的二元关系。但是在具体讨论的时候，将不至于发生混淆。

下面来给出图的定义。

定义 1.2 设V是非空集合，E是$V\times V$（$V\&V$）的一个多重子集，称偶对(V,E)为有向图（无向图），记为$D(V,E)$（$G(V,E)$）。V中的元素称为顶点或结点，V称为结点集；E中的元素称为有向边（无向边），简称边，E称为边集。有向图和无向图统称为图。用G泛指一个图。

注 1: 我们分别用$V(G)$和$E(G)$表示图G的结点集和边集，当我们说结点$v\in G$时意味着$v\in V(G)$，而说边$e\in G$意味着$e\in E(G)$。如果$V(G)$和$E(G)$都是有限集，则称G为有限图；否则称G为无限图。本课程只限于讨论有限图，今后，如果不加特殊说明，就认为

$$V=\{v_1,\cdots,v_n\},\quad E=\{e_1,\cdots,e_m\},$$

即结点数$|V|=n$，边数$|E|=m$，结点数n称为图G的阶。

注 2: 人们总是用图形来表示一个图，即用小圆圈（或实心点）表示结点，用结点之间的线段表示无向边，用有向线段表示有向边。如图 1.1, 1.2 所示。

注 3: 若 $E(G)=\varnothing$，则称 G 为零图，n 阶零图记为 N_n，特别称 N_1 为平凡图。为了后面方便表示图的运算，我们引入一个无实际意义的图：空图，即结点集为空集的图，记作 $\varnothing$。

注 4：将有向图 D 各有向边的箭头都去掉所得无向图 G 称为 D 的基图。

为叙述方便起见，下面再引进一些术语。如果边 $e_k=(v_i,v_j)$，则称 v_i 与 v_j 相邻，e_k 分别与 v_i，v_j 互相关联，v_i，v_j 是 e_k 的端点；如果 e_k 是有向边，则称 v_i 是 e_k 的始点，v_j 是 e_k 的终点，并称 v_i 是 v_j 的直接前趋，v_j 是 v_i 的直接后继。无向图的两条边 e_k 和 e_l 若有一个公共端点，则称它们是相邻的。在图 G 中，只与一个结点相关联的边称为自环；同一对结点间可以存在多条边，称为重边（或平行边），含有重边的图称为多重图；称既不含重边也不含自环的图为简单图。

简单图是我们重点研究的图。

定义 1.3 任何两结点间都有边相连的 n 阶无向简单图称为 n 阶无向完全图，记作 K_n。

由上面讨论可知，图的基本要点之一是结点与边的关联关系，下面介绍一个量度这种关联程度的概念：结点的度数。

定义 1.4 对于图 G 中任意结点 v，v 作为图 G 中边的端点的次数，称为 v（在 G 中）的度数，简称 v 的度，记作 $d_G(v)$，在不发生混淆的情况下，间记为 $d(v)$。

注 1：若 v 带有自环，则自环对 $d(v)$ 的贡献为 2。

注 2：在有向图中，由于各边都是有向边，所以对每个结点 v，还可定义其正度（或称为出度）$d_G^+(v)$（简记为 $d^+(v)$）为以 v 为始点的边的数目；定义其负度（或称为入度）$d_G^-(v)$（简记为 $d^-(v)$）为以 v 为终点的边的数目。显然有 $d_G(v)=d_G^+(v)+d_G^-(v)$。

注 3: 在图 G 中，度为 0 的结点，即没有边关联的结点称为孤立点；度为 1 的结点称为悬挂点；度为奇数的结点称为奇结点；度为偶数的结点称为偶结点。

注 4: 如果无向图 G 的每一个结点都有相同的度，则称 G 是正则的；度为 k 的正则图称为 k 正则图。

下面列出结点度的一些基本性质。

性质 1.1 （Euler, 1736）设图G有n个结点， m条边，则

$$\sum_{i=1}^{n} d(v_i) = 2m \text{。}$$

该性质是图论中的第一个定理，称为图论的基本定理或握手定理。

性质 1.2 图G中奇结点的数目必是偶数。

性质 1.3 有向图D中，正度之和等于负度之和。

$$\sum_{i=1}^{n} d^+(v_i) = \sum_{i=1}^{n} d^-(v_i) = m$$

该性质可看作有向图版的握手定理。

性质 1.4 K_n中结点的度都为$n-1$，边数为$\frac{1}{2}n(n-1)$。

性质 1.5 2阶以上无向简单图中一定存在度相同的结点。

下面再引入一些表示关联的术语。

对于图G的任意结点v，称v的邻点集为v的邻域，记为$N_G(v)$；称$N_G(v)\bigcup\{v\}$为v的闭邻域，记为$\overline{N_G(v)}$；若$S \subseteq V$，则称$N_G(S) = \bigcup_{v\in S} N_G(v)$为$S$的邻域。称

$$\{e \in E(G) : e\text{与}v\text{相关联}\}$$

为v的关联集，记作$I_G(v)$。若G是有向图，则称v的直接后继集为v的外邻域，记作$N_G^+(v)$；称v的直接前趋集为v的内邻域，记作$N_G^-(v)$。显然有$N_G(v) = N_G^+(v)\bigcup N_G^-(v)$。令

$$\Delta(G) = \max\{d(v) : v \in V(G)\}\text{，} \quad \delta(G) = \min\{d(v) : v \in V(G)\}\text{，}$$

称$\Delta(G)$，$\delta(G)$分别为G的最大度和最小度。在有向图D中，可类似定义$\Delta^+(D)$，$\delta^+(D)$，$\Delta^-(D)$，$\delta^-(D)$分别为D的最大出度，最小出度，最大入度，最小入度。

例 1.3 证明：任何 6 人当中，要么有 3 人互相认识，要么有 3 人互相不认识。

证明：以其中一人v_0来说，在其余 5 人中，要么他认识的人数至少为 3，要么他不认识的人

数至少为 3。不妨假定为前者，设v_1，v_2，v_3为v_0认识的 3 人。若这 3 人中有两人互相认识，再加上v_0，就构成互相认识的 3 人小组；否则，v_1，v_2，v_3构成互相不认识的 3 人小组。

下面介绍子图的概念。

定义 1.5 设图$G=(V,E)$和$G'=(V',E')$满足：$V'\subseteq V$，$E'\subseteq E$，则称G'是G的一个子图。特别地，若$V'\subset V$或$E'\subset E$，则称G'是G的真子图；如果$V'=V$，则称G'是G的支撑子图或生成子图；如果E'包含了G在结点子集V'之间的所有边，则称G'是G的导出子图，

此时，记$G'=G[V']$。

注：用$G'\subseteq G$和$G'\subset G$分别表示G'是G的子图和真子图。

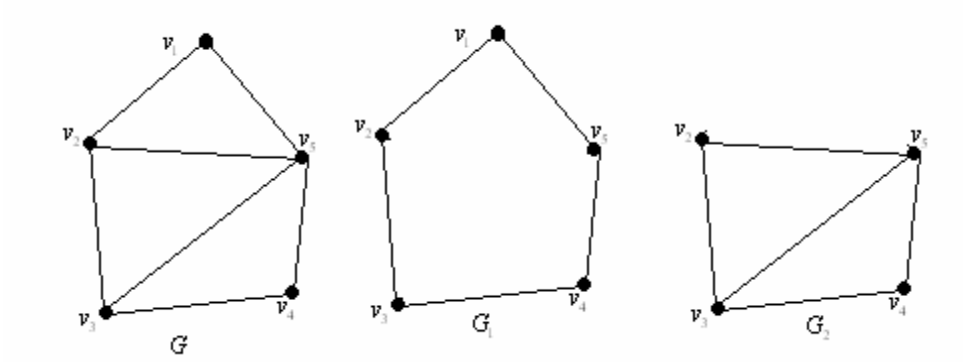

图 1.3

G_1是G的生成子图，G_2是G的导出子图。

G既是自身的生成子图，又是导出子图；零图（与G同阶）是G的生成子图，它们都称为平凡子图。

定义 1.6 给定两个图$G_1=(V_1,E_1)$，$G_2=(V_2,E_2)$，如果存在从V_1到V_2的双射f，使得对任意$u,v\in V_1$，有$(u,v)\in E_1 \Leftrightarrow (f(u),f(v))\in E_2$，则称$G_1$与$G_2$同构。记作$G_1\cong G_2$。

例 1.4 图 1.4 中的两个图同构。

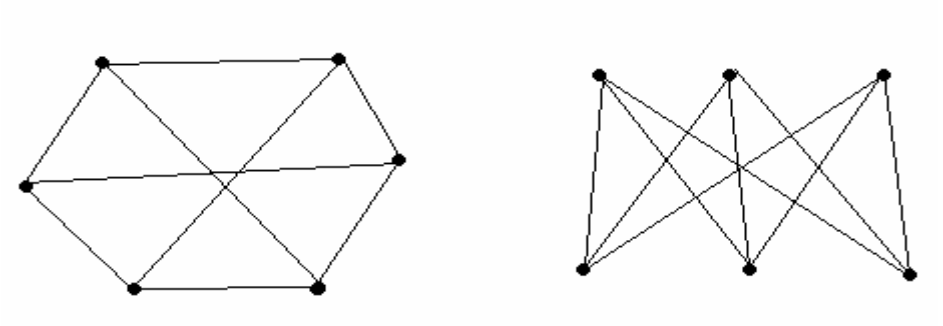

图 1.4

从定义 1.6 知道，若$G_1 \cong G_2$，必满足

（1）$|V(G_1)|=|V(G_2)|$，$|E(G_1)|=|E(G_2)|$；

（2）G_1和G_2结点度的集合（多重的）相同；

（3）G_1和G_2的所有导出子图一一同构。

例 1.5　图 1.5 中的两个图不同构。

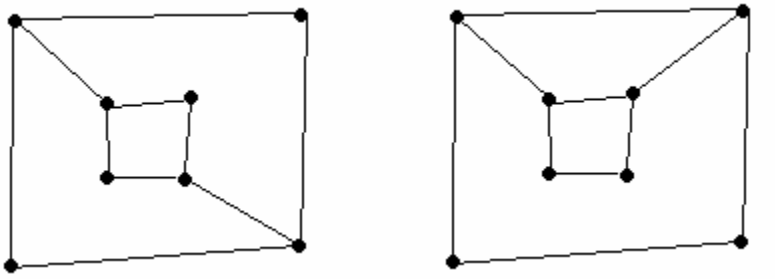

图 1.5

设G是n阶无向简单图，定义它的补图$\overline{G}$如下：$V(\overline{G})=V(G)$，$E(\overline{G})=E(K_n)-E(G)$。

例 1.3 说明，对于任意 6 阶无向简单图G，要么$K_3 \subseteq G$，要么$K_3 \subseteq \overline{G}$。

如果简单无向图G与它的补图$\overline{G}$同构，则称G是自补图。

作业

1. 证明：对于任意一个 9 阶无向简单图G，要么$K_4 \subseteq G$，要么$K_3 \subseteq \overline{G}$。

2. 对无向完全图的每条边任给一方向。得到的有向图称为竞赛图。证明：在任何竞赛图中，有

$$\sum_{i=1}^{n}(d^{+}(v_i))^2=\sum_{i=1}^{n}(d^{-}(v_i))^2$$

3. 若G是自补图，则必有$n=4k$或$n=4k+1$。

现在来看图的运算问题。

定义 1.7 设$G=(V,E)$为一无向图，若$E'\subseteq E$，我们用$G-E'$表示从G中删去E'中所有边得到的图，即$G-E'=(V,E-E')$，此种运算称为删除E'；若$V'\subseteq V$，以$G-V'$表示从G中删去V'中的点及与V'中的点关联的所有边得到的图，即$G-V'=G[V-V']$，此种运算称为删除V'。我们用$G-e$和$G-v$分别简记$G-\{e\}$和$G-\{v\}$。若u，v是G中一对不相邻结点，用$G+(u,v)$表示在u，v之间加一条边，称为加新边。另外，设$V'\subseteq V$，所谓在G中收缩V'，是指在图G中，把V'中的所有点重合为一个点（为方便起见，有时称之为伪点），除V'中结点之间的边外，与V'中的结点相关联的边变为与此新点相关联的边，这样得到的图称为关于V'的收缩图，记为$G\circ V'$。当我们说，在G中收缩子图G'时，意味着在G中收缩$V(G')$，例如在G中收缩边(u,v)，就是指在G中收缩$\{u,v\}$，也记所得的图为$G\circ(u,v)$。

图 1.6

定义 1.8 设$G_1=(V_1,E_1)$，$G_2=(V_2,E_2)$为两图，

（1） 若$V_1\bigcap V_2=\varnothing$，则称$G_1$与$G_2$是不交的；

（2） 若$E_1\bigcap E_2=\varnothing$，则称$G_1$与$G_2$是边不交的或边不重的。

不交的两图必为边不交的，反之不真。

定义 1.9 设 $G_1=(V_1,E_1)$，$G_2=(V_2,E_2)$ 均为无孤立点的图，称以 $E_1\bigcup E_2$（$E_1\bigcap E_2$，E_1-E_2，$E_1\oplus E_2$）为边集，以 $E_1\bigcup E_2$（$E_1\bigcap E_2$，E_1-E_2，$E_1\oplus E_2$）中的边关联的结点的集合为结点集的图为 G_1 和 G_2 的并图（交图，差图，环和），记作 $G_1\bigcup G_2$（$G_1\bigcap G_2$，G_1-G_2，$G_1\oplus G_2$）。

注 1: 有 $G_1\oplus G_2=G_1\bigcup G_2-G_1\bigcap G_2$ 等类似集合运算的一些恒等式。

注 2: $G-G=G\oplus G=\varnothing$。

定义 1.10 设 $G_1=(V_1,E_1)$，$G_2=(V_2,E_2)$ 为两个不交的无向图，称以 $V=V_1\bigcup V_2$ 为结点集，以 $E_1\bigcup E_2\bigcup\{(v_1,v_2):v_1\in V_1,v_2\in V_2\}$ 为边集的图为 G_1 和 G_2 的联图，记作 G_1+G_2。

注 1: 从定义 1.10 不难看出，$K_{n_1}+K_{n_2}=K_{n_1+n_2}$；

注 2: G_1+G_2 的结点数为 n_1+n_2，边数为 $m_1+m_2+n_1n_2$。

定义 1.11 设 $G_1=(V_1,E_1)$，$G_2=(V_2,E_2)$ 为两个无向图，称以 $V_1\times V_2$ 为结点集，以

$$\{((u_1,u_2),(v_1,v_2)):u_1=v_1\text{且}(u_2,v_2)\in E_2,\text{或}u_2=v_2\text{且}(u_1,v_1)\in E_1\}$$

为边集的图为 G_1 和 G_2 的积图，记作 $G_1\times G_2$。

注 1；$G_1\times G_2$ 的结点数为 n_1n_2，边数为 $n_1m_2+n_2m_1$；

注 2: 用 $0,1$ 分别表示 K_2 的两个结点，令 $Q_1=K_2$，$Q_i=Q_{i-1}\times K_2$（$i\geq 2$），则称 Q_k 为 $k-$ 方体图，易知 Q_k 中有 2^k 个结点，$k2^{k-1}$ 条边。

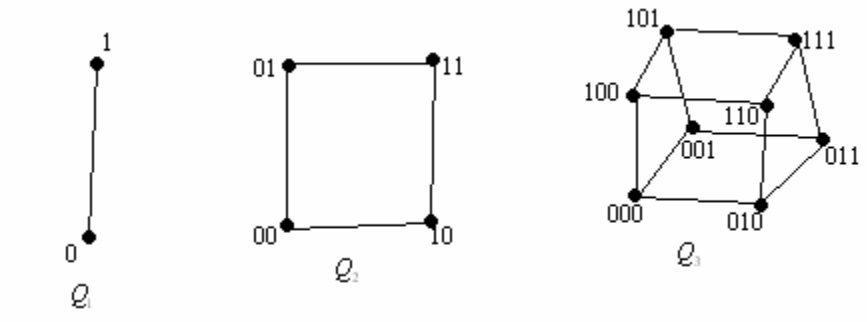

图 1.7

作为本节的结束，我们给出赋权图的定义。

在把一个实际问题抽象化为一个图的问题时，有许多原因需要在图的结点或（和）边上，标注一些附加的信息。例如，表示城市间公路交通的图，每一条边上可以写上一个数，来表示边连接的两个城市之间的距离。我们也可以在每个结点上标一个数，用以表示该城市的人数。在表示球类比赛的图中，每一条边可标注两球队之间的比分和比赛日期。如此等等，需要引入赋权图的概念。

定义 1.12 对于图 $G=(V,E)$，设 f 是 V 上的实函数，g 是 E 上的实函数，称有序三元组 $G_1=(V,E,f)$，$G_2=(V,E,g)$，$G_3=(V,E,f,g)$ 为赋权图。

注：f 给每个结点赋权，g 给每条边赋权。我们往后主要讨论带边权的赋权图。边权都为正数的赋权图称为正权图。边权物理意义：长度，时间，费用，容量。

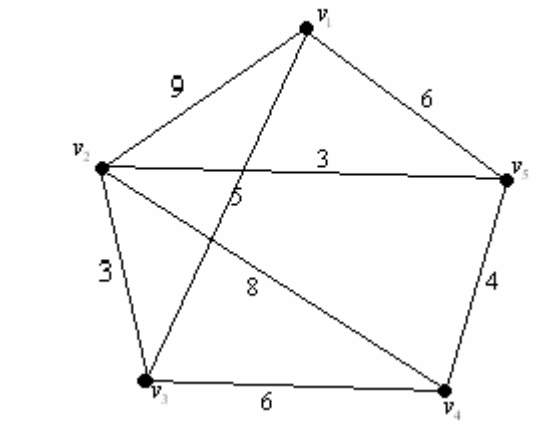

图 1.8

§1.2 图的代数表示

在对图G进行描述或运算时，需要采用代数方法进行表示，常用的表示方法有：

1.2.1 邻接矩阵

邻接矩阵表示图的结点之间的邻接关系。邻接矩阵是一个n阶方阵$A=(a_{i,j})_{n\times n}$，其元素

$$a_{i,j}=\begin{cases}1, & \text{if } (v_i,v_j)\in E\\ 0, & \text{otherwise}\end{cases}。$$

邻接矩阵可表示自环，但无法表示重边。对有向图来说，邻接矩阵的第i行的1元数目正是v_i的正度$d^+(v_i)$，而第j列的1元数目是v_j的负度$d^-(v_j)$；对无向图来说，邻接矩阵是对称矩阵，第i行（列）的1元数目正是v_i的度$d(v_i)$。

1.2.2 权矩阵

赋权图$G=(V,E,g)$（无重边）常用权矩阵$W=(w_{i,j})_{n\times n}$表示。其中

$$w_{i,j}=\begin{cases}g((v_i,v_j)), & \text{if } (v_i,v_j)\in E\\ 0\ (\text{or }\infty), & \text{otherwise}\end{cases}$$

$g((v_i,v_j))$是边(v_i,v_j)的权。

1.2.3 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$B=(b_{i,j})_{n\times m}$，其中

$$b_{i,k}=\begin{cases}1, & \text{if 结点}v_i\text{是边}e_k\text{的始点}\\ -1, & \text{if 结点}v_i\text{是边}e_k\text{的终点}\\ 0, & \text{if结点}v_i\text{不是边}e_k\text{的端点}\end{cases}。$$

关联矩阵具有下列性质

（1） 每列只有两个非零元：1和—1。

（2） 第i行1元的数目恰是$d^+(v_i)$，—1元数目恰是$d^-(v_j)$。

（3） 关联矩阵能够表示重边，但不能表示自环。

类似可定义无向图的关联矩阵，但其中不含—1元。

用邻接矩阵和关联矩阵表示图十分直观。但在使用计算机对图进行各种操作时，采用邻接矩阵或关联矩阵作为输入形式将占据较大的存储空间并可能增加计算复杂度。为克服这些缺陷，产生了图的一些其他表示方法。

1.2.4 边列表

边列表是对关联矩阵的列进行压缩的结果。它由两个 m 维向量 A 和 B 组成。当对 G 的结点和边分别编号后，若 $e_k=(v_i,v_j)$，则 $A(k)=i$， $B(k)=j$，即 $A(k)$ 存放第 k 条边的始点编号，$B(k)$ 存放第 k 条边的终点编号。如果 G 是赋权图，则再增加一个 m 维向量 C，若 e_k 的权是 w_k，则 $C(k)=w_k$。

边列表可以表示自环。

例 2.1 图 1.9 所示赋权有向图的边列表表示如下：

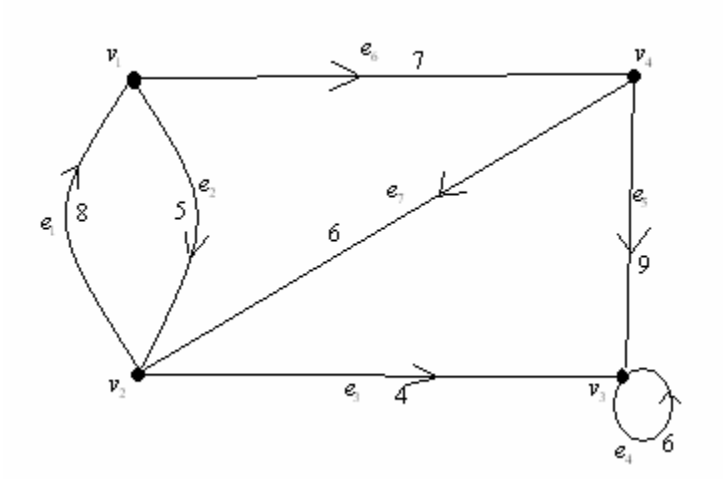

图 1.9

$A=(2,\ 1,\ 2,\ 3,\ 4,\ 1,\ 4)$，

$B=(1,\ 2,\ 3,\ 3,\ 3,\ 4,\ 2)$，

$C=(8,\ 5,\ 4,\ 6,\ 9,\ 7,\ 6)$。

无向图的边列表表示不唯一。

1.2.5 正向表

正向表是对邻接矩阵的行进行压缩的结果，它的特点是将每个结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和一个m维向量B组成。当对G的结点编号后，向量B集中存放每个结点直接后继的编号，$A(i)$表示结点v_i的第一个直接后继在B中存放的位置，约定$A(n+1)=m+1$（起哨兵作用）。如果如果G是赋权图，则再增加一个m维向量C，用以存放对应边的权值。

例 2.2 图 1.9 所示赋权有向图的正向表表示如下：

$A=(1,\ 3,\ 5,\ 6,\ 8)$，

$B=(2,\ 4,\ 1,\ 3,\ 3,\ 2,\ 3)$，

$C=(5,\ 7,\ 8,\ 4,\ 6,\ 6,\ 9)$。

在正向表中，有下列数量关系

（1）$d^+(v_i)=A(i+1)-A(i)$。

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

（3）当$A(i)=A(i+1)$时，说明v_i无直接后继；当$A(i)<A(i+1)$时，则从分量$B(A(i))$到分量$B(A(i+1)-1)$的每一个值都是v_i的直接后继。

由于无向图的边没有方向性，所以B中集中存放的是邻接点的编号，因而B是$2m$维向量，A依然是$n+1$维向量，但需约定$A(n+1)=2m+1$。

类似可定义反向表，是对邻接矩阵的列进行压缩的结果。

1.2.6 正向邻接表

正向邻接表采用单链表结构表示一个图，每个结点的所有外邻点组成一个单链表。每个外邻点用一个表结点表示，表结点的结构如下：

a	b	c

它包括三个域，外邻接点域a，数据域b存放对应边的权值，指针域c存放表结点指针，指向下一个外邻点。n维头指针数组H的分量$H(i)$指向结点v_i的第一个外邻点对应的表结

点。

例 2.3 图 1.9 所示赋权有向图的正向邻接表如图 1.10 所示。

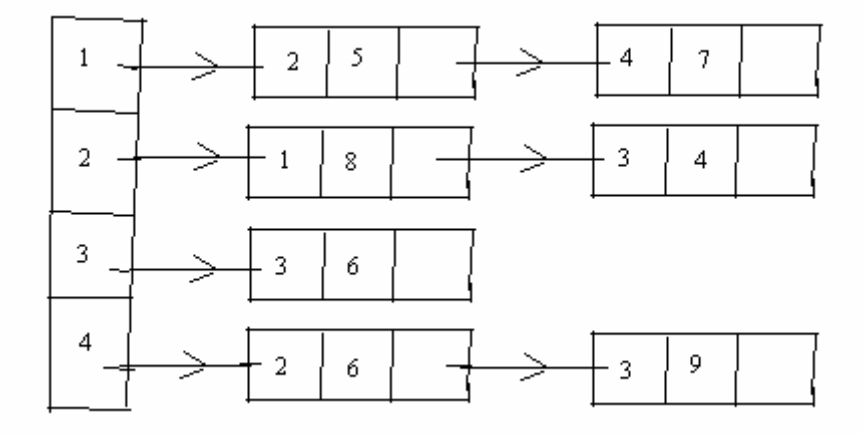

图 1.10

类似定义反向邻接表。

邻接表的特点是使用灵活，譬如要从图G中删去某条边时，只要删除对应的表结点就可以了；若要增加某条边，也只需增加一个表结点，而不需要大的变动。

边列表，正向表，正向邻接表等都能表示重边和自环，而且也都只占据较小的存储空间。邻接矩阵，关联矩阵，边列表，正向表，正向邻接表之间都可以互相转换。为简单起见，本科程主要采用邻接矩阵和正向邻接表两种形式的数据结构。

作业

编写有向图G的邻接矩阵与关联矩阵，邻接矩阵与正向表，关联矩阵与边列表，邻接矩阵与正向邻接表之间互相转化的程序。