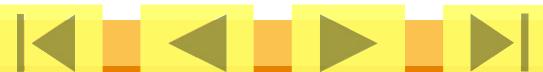


离散数学

第五章：图论一

2025年9月7日星期日



引言

我们所讨论的图(Graph)与人们通常所熟悉的图，例如圆、椭圆、函数图表等是很不相同的。图论中所谓的图是指**某类具体离散事物集合和该集合中的每对事物间以某种方式相联系的数学模型**。

如果我们用**点**表示**具体事物**，用**连线**表示一对**具体事物之间的联系**。那么，一个图就是由一个表示**具体事物的点的集合**和表示**事物之间联系的一些线的集合**所构成，至于**点的位置**和**连线的长短曲直**是无关紧要的。

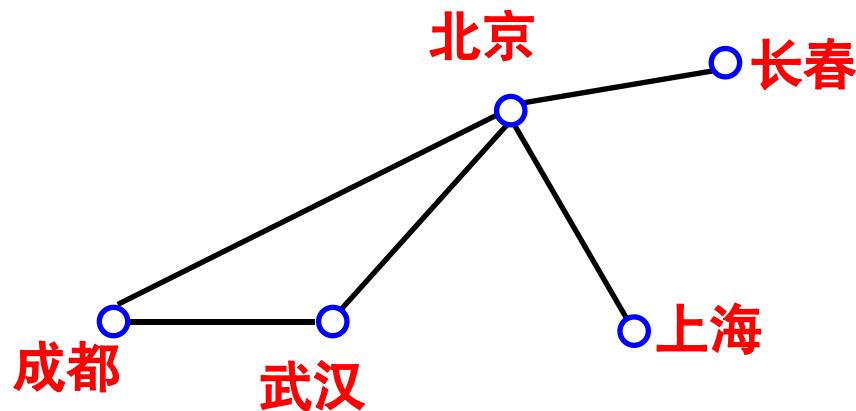
5.1 内容提要

- 1 图的基本概念
- 2 图的表示、分类
- 3 图的性质
- 4 通路与回路
- 5 图的连通性
- 6 图的应用

5.2 图的基本概念

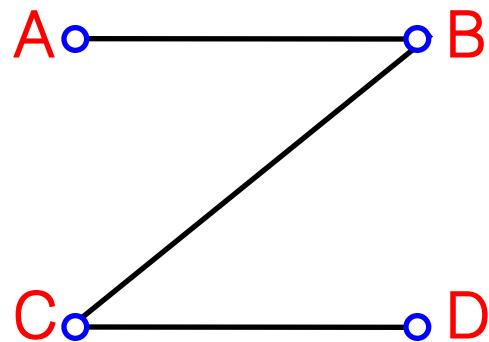
5.2.1 图的定义

例5.2.1 (1) 考虑一张航线地图，图中用点表示城市，当两个城市间有直达航班时，就用一条线将相应的点连接起来。这种航线地图的一部分如下图所示；



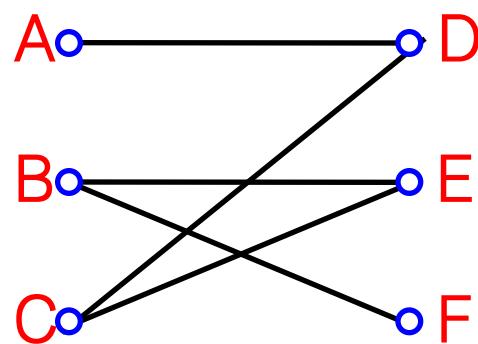
例5.2.1 (2)

假设有4台计算机，分别标记为A、B、C和D，在计算机A和B、C和D以及B和C之间有信息流。这种情形可用下图表示，通常称这种图为**通信网络**；



例5.2.1 (3)

假设有一群人和一组工作，这群人中的某些人能够做这组工作中的某些工作。例如，有3个人A、B和C，3件工作D、E和F，假设A只能做工作D，B能做工作E和F，C能做工作D和E。则这种情形可用下图表示，其中，在人和这个人能够做的工作之间画有线。



基本思想

用图形表示一组对象，其中有些对象对是有联系的。当然，这几个图形也可以表示其它的含义。例如在（3）的图中点A、B、C、D、E和F分别表示6家企业，如果某两家企业有业务往来，则其对应的点之间用线连接起来，这时的图形又反映了这6家企业间的业务关系。

对于这种图形，我们感兴趣的只是有多少个点和哪些结点之间有线连接，至于连线的长短曲直和结点的位置却无关紧要，只要求每一条线都起始于一个点，而终止于另一个点。

定义5.2.1

一个图(Graph)是一个序偶 $\langle V, E \rangle$, 记为 $G = \langle V, E \rangle$, 其中:

- (1) $V = \{v_1, v_2, \dots, v_n\}$ 是有限非空集合, v_i 称为结点(Nodal Point), 简称点(Point), V 称为结点集(Nodal Set)。
- (2) E 是有限集合, 称为边集(Frontier Set)。 E 中的每个元素都有 V 中的结点对与之对应, 称之为边(Edge)。

与边相关的几个概念

定义5.2.1中的结点对即可以是无序的，也可以是有序的。

若边 e 与无序结点对 (u, v) 相对应，则称 e 为无向边(Undirected Edge)，记为 $e = (u, v) = (v, u)$ ，这时称 u 、 v 是边 e 的两个端点(End point)。

若边 e 与有序结点对 $\langle u, v \rangle$ 相对应，则称 e 为有向边(Directed Point)(或弧)，记为 $e = \langle u, v \rangle$ ，这时称 u 为 e 的始点(Initial Point)(或弧尾)， v 为 e 的终点(terminal Point)(或弧头)，统称为 e 的端点。

5.2.2 图的表示

对于一个图 G , 如果将其记为 $G = \langle V, E \rangle$, 并写出 V 和 E 的集合表示, 这称为**图的集合表示**。

而为了描述简便起见, 在一般情况下, 往往只画出它的图形: 用小圆圈表示 V 中的结点, 用由 u 指向 v 的有向线段或曲线表示有向边 $\langle u, v \rangle$, 无向线段或曲线表示无向边 (u, v) , 这称为**图的图形表示**。

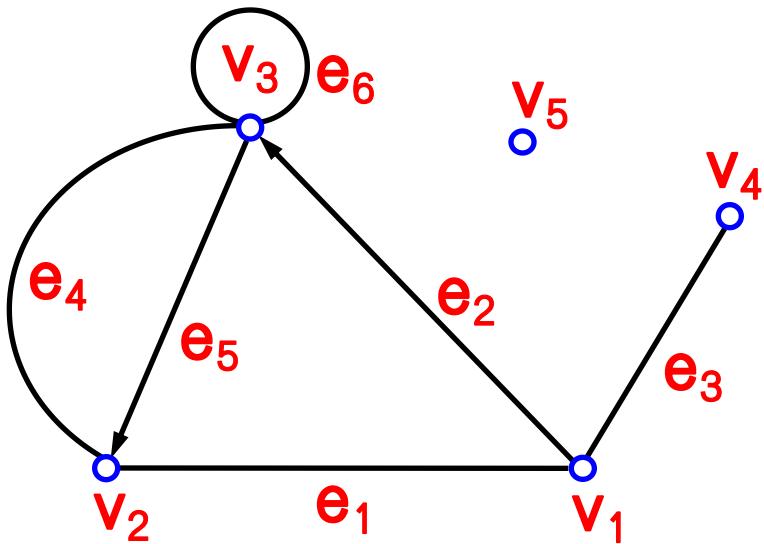
例5.2.2

设图 $G = \langle V, E \rangle$, 这里 $V = \{v_1, v_2, v_3, v_4, v_5\}$, $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$, 其中 $e_1 = (v_1, v_2)$, $e_2 = \langle v_1, v_3 \rangle$, $e_3 = (v_1, v_4)$, $e_4 = (v_2, v_3)$, $e_5 = \langle v_3, v_2 \rangle$, $e_6 = (v_3, v_3)$ 。试画出图G的图形，并指出哪些是有向边，哪些是无向边？

分析 由于 V 中有 5 个结点，因此要用 5 个小圆圈分别表示这 5 个结点，点的具体摆放位置可随意放。而对 E 中的 6 条边，圆括号括起的结点对表示无向边，直接用直线或曲线连接两个端点，尖括号括起的结点对表示有向边，前一个是始点，后一个始终点，用从始点指向终点的又向直线或曲线连接。

解

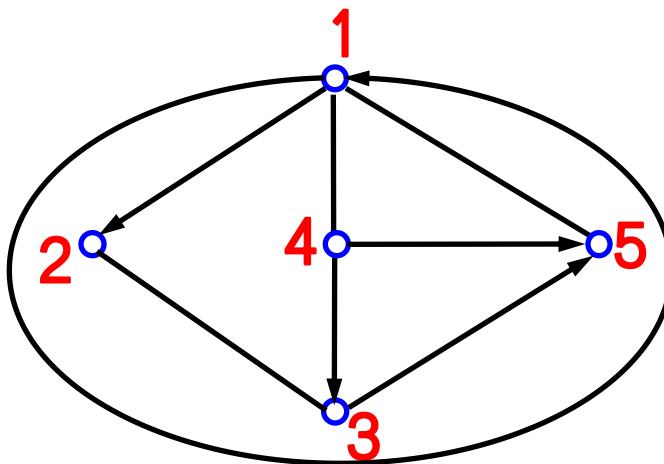
G的图形如下图所示。



G中的 e_1 、 e_3 、 e_4 、 e_6 是无向边， e_2 、 e_5 是有向边。

例5.2.3

设图 $G = \langle V, E \rangle$ 的图形如下图所示，试写出 G 的集合表示。



解析图④的集合表示为记号构成结点集合 $\{1, 2, 3, 4, 5\}$ ，将连接结点对的直线或曲线用圆括号括起该结点对表示无向边，将连接结点对的有向直线或曲线用尖括号括起该结点对表示有向边，这里箭头指向的结点放在后面。

两种描述方法的优缺点

- 用集合描述图的优点是**精确**，但**抽象不易理解**；
- 用图形表示图的优点是**形象直观**，但当图中的结点和边的数目较大时，使用这种方法是很**不方便**的，甚至是**不可能的**。

图的矩阵表示

我们在学习中常常需要分析图并在图上执行各种过程和算法，也许必须用计算机来执行这些算法，因此必须把图的结点和边传输给计算机，由于集合与图形都不适合计算机处理，所以要找到一种新的表示图的方法，这就是图的矩阵表示。

由于矩阵的行和列有固定的次序，因此在用矩阵表示图时，先要将图的结点进行排序，若不具体说明排序，则默认为书写集合V时结点的顺序。

定义5.2.2

设图 $G = \langle V, E \rangle$, 其中 $V = \{v_1, v_2, \dots, v_n\}$, 并假定结点已经有了从 v_1 到 v_n 的次序, 则 n 阶方阵 $A_G = (a_{ij})_{n \times n}$ 称为 G 的邻接矩阵 (Adjacency Matrix), 其中

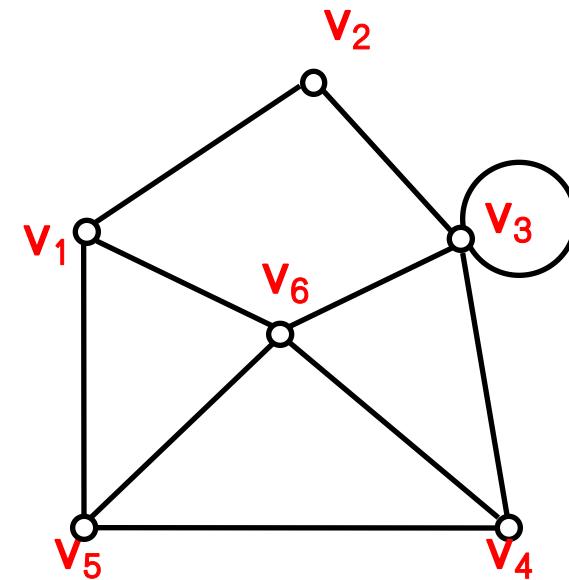
$$a_{ij} = \begin{cases} 1, & \text{若 } (v_i, v_j) \in E \text{ 或 } \langle v_i, v_j \rangle \in E \\ 0, & \text{否则} \end{cases} \quad i, j = 1, 2, 3, \dots, n$$

例5.2.4

试写出下图所示图G的邻接矩阵。

解析若将该图的6个结点排序则为 $v_2, v_3, v_4, v_5, v_6, v_1$ 。写出其邻接矩阵。

初学时可先在矩阵的行与列前分别按结点排序上结点，若第i行前的结点到第j列前的结点有边相连，则在邻接矩阵的第i行第j列元素为1，否则为0。若有结点排 $v_1, v_2, v_3, v_4, v_5, v_6$ ，则可标记如下：



说明

由定义5.2.2可看出，图 $G = \langle V, E \rangle$ 的邻接矩阵依赖于 V 中元素的次序。对于 V 中各元素不同的排序，可得到同一图 G 的不同邻接矩阵。但是， G 的任何一个邻接矩阵可以从 G 的另一邻接矩阵中通过交换某些行和相应的列而得到，其交换过程与将一个排序中的结点交换位置变为另一个排序是一致的。如果我们略去由结点排序不同而引起的邻接矩阵的不同，则图与邻接矩阵之间是一一对应的。因此，我们略去这种由于 V 中元素的次序而引起的邻接矩阵的任意性，只选 V 中元素的任一种次序所得出的邻接矩阵，作为图 G 的邻接矩阵。

例

图中的结点重排次序为
 $v_5v_2v_1v_3v_6v_4$, 得另一个邻接矩阵

$$A_{1G} = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

在邻接矩阵 A_{1G} 中, 如果先交换第1、3行, 而后交换第1、3列; 接着交换第3、4行, 再交换第3、4列; 接着交换第5、6行, 再交换第5、6列; 接着交换第4、5行, 再交换第4、5列。那么就能由邻接矩阵 A_{1G} 得到邻接矩阵 A_G 。

5.2.3 图的操作

定义5.2.3 设图 $G = \langle V, E \rangle$ 。

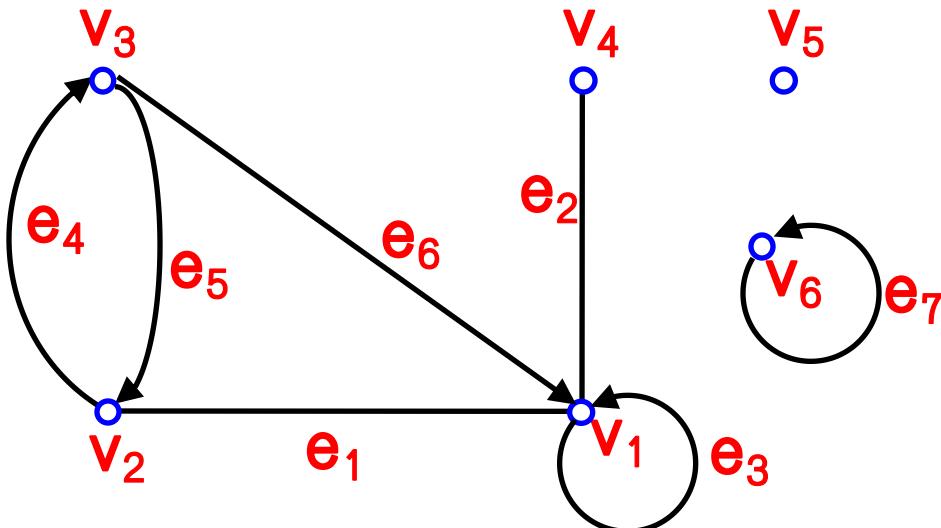
1. 设 $e \in E$, 用 $G - e$ 表示从 G 中去掉边 e 得到的图, 称为 **删除边 e** 。又设 $E' \subseteq E$, 用 $G - E'$ 表示从 G 中删除 E' 中所有边得到的图, 称为 **删除 E'** 。
2. 设 $v \in V$, 用 $G - v$ 表示从 G 中去掉结点 v 及 v 关联的所有边得到的图, 称为 **删除结点 v** 。又设 $V' \subset V$, 用 $G - V'$ 表示从 G 中删除 V' 中所有结点及关联的所有边得到的图, 称为 **删除 V'** 。
3. 设 $e = (u, v) \in E$, 用 $G \setminus e$ 表示从 G 中删除 e , 将 e 的两个端点 u, v 用一个新的结点 w 代替, 使 w 关联除 e 外的 u 和 v 关联的一切边, 称为 **边 e 的收缩**。一个图 G 可以收缩为图 H , 是指 H 可以从 G 经过若干次边的收缩而得到。
4. 设 $u, v \in V$ (u, v 可能相邻, 也可能不相邻), 用 $G \cup (u, v)$ 表示在 u, v 之间加一条边 (u, v) , 称为 **加新边**。

5.2.4 邻接点与邻接边

定义5.2.4 在图 $G = \langle V, E \rangle$ 中，若两个结点 v_i 和 v_j 是边 e 的端点，则称 v_i 与 v_j 互为邻接点(Adjacent Point)，否则 v_i 与 v_j 称为不邻接的；具有公共结点的两条边称为邻接边(Adjacent Edge)；两个端点相同的边称为环(Ring)或自回路(Self-Loop)；图中不与任何结点相邻接的结点称为孤立结点(Isolated Point)；仅由孤立结点组成的图称为零图(Null Graph)；仅含一个结点的零图称为平凡图(Trivial Graph)；含有 n 个结点， m 条边的图，称为(n, m)图。

例5.2.5

试写出下图所示图G的所有结点的邻接点、所有边的邻接边，并指出所有的孤立结点和环。



分析

根据定义5.2.4，如果两个结点间有边相连，那么它们互为邻接点；如果两条边有公共结点，那么它们互为邻接边。

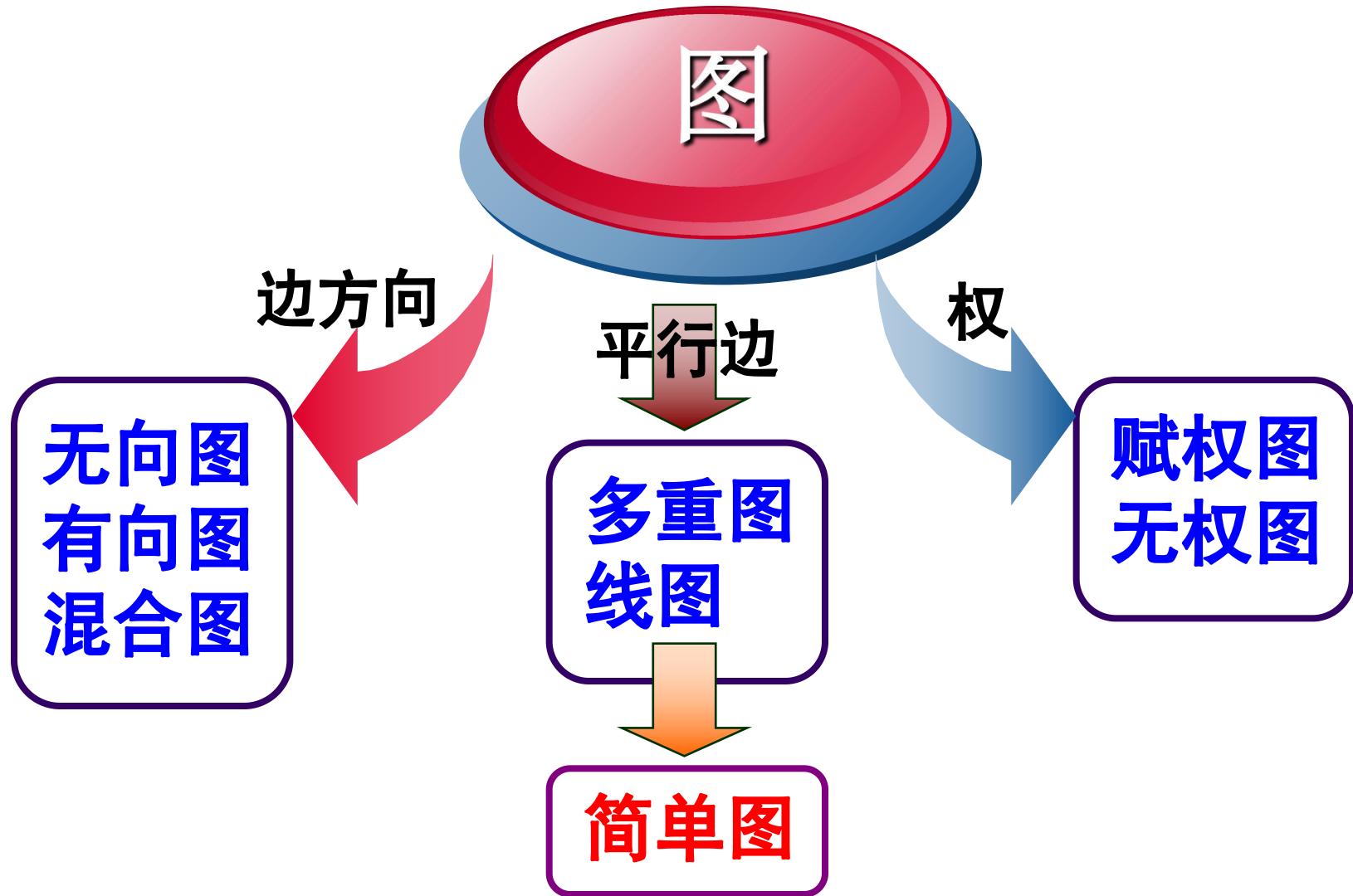
需要注意的是，只要当一个结点处有环时，它才是自己的邻接点；由于一条边有两个端点，在计算邻接边时要把这两个端点都算上，例如 e_2 和 e_4 都是 e_1 的邻接边。所有边都是自己的邻接边。

解

图G所有结点的邻接点和孤立结点，所有边的邻接边和环如下表所示。图G既不是平凡图，也不是零图，而是一个(6, 7)图。

结点	邻接点	是否孤立结点	边	邻接边	是否环
v_1	v_1, v_2, v_3, v_4	否	e_1	$e_1, e_2, e_3, e_4, e_5, e_6$	否
v_2	v_1, v_3	否	e_2	e_1, e_2, e_3, e_6	否
v_3	v_1, v_2	否	e_3	e_1, e_2, e_3, e_6	是
v_4	v_1	否	e_4	e_1, e_4, e_5, e_6	否
v_5		是	e_5	e_1, e_4, e_5, e_6	否

5.2.5 图的分类



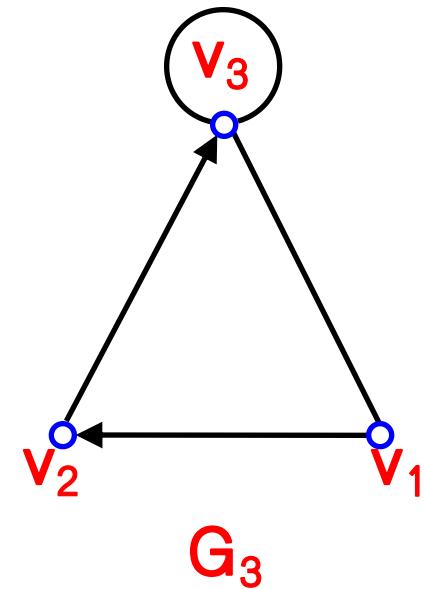
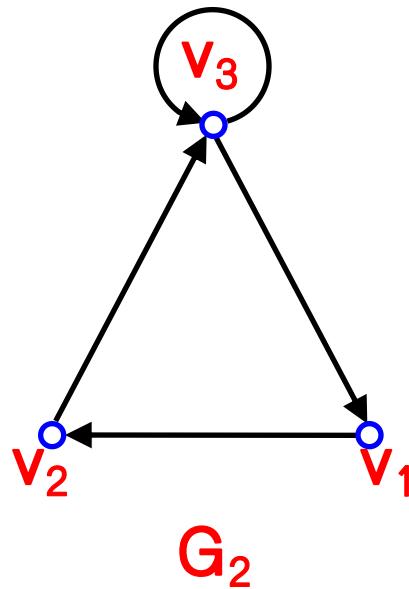
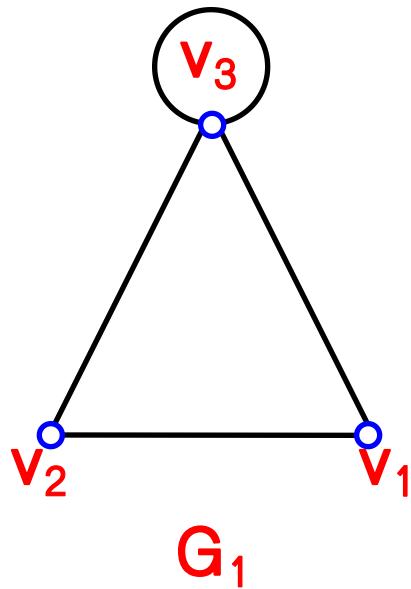
1. 按边有无方向分类

定义5.2.5 每条边都是无向边的图称为**无向图**(Undirected Graph); 每条边都是**有向边**的图称为**有向图**(Directed Graph); 有些边是**无向边**, 而另一些边是**有向边**的图称为**混合图**(Mixed Graph)。

关系中的关系图都是**有向图**, 这时邻接矩阵就是**关系矩阵**。

例5.2.6

试判断下图所示的三个图是无向图、有向图，还是混合图？

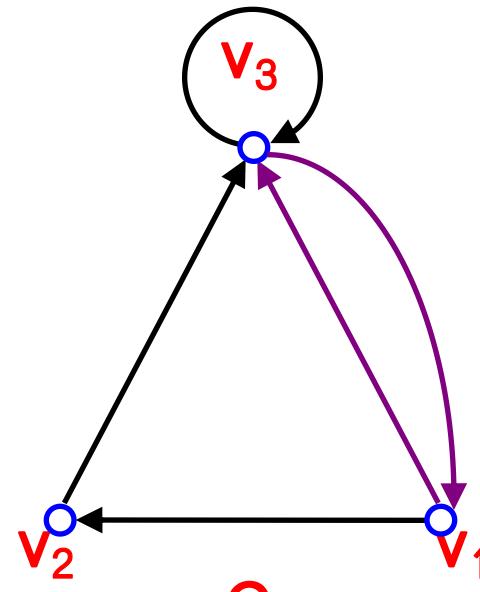


分析 判断无向图、有向图和混合图， G_1 为无向图， G_2 为有向图， G_3 为混合图。
无方向就行了。

说明

我们仅讨论无向图和有向图，至于混合图，我们可将其中的无向边看成方向相反的两条有向边，从而转化为有向图来研究。

例如可将图 G_3 转化为下图。

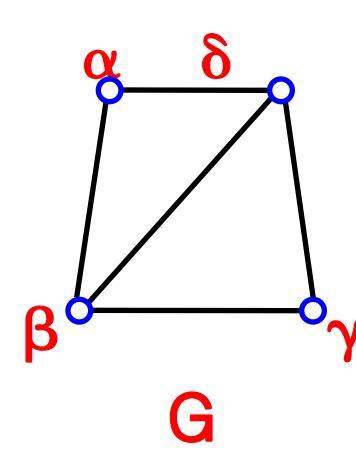
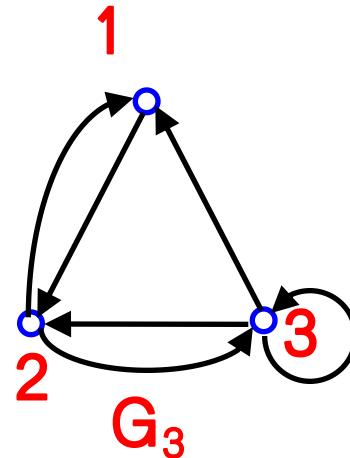
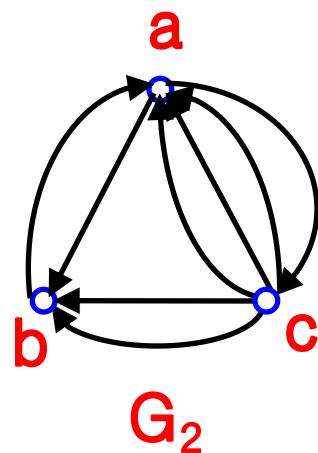
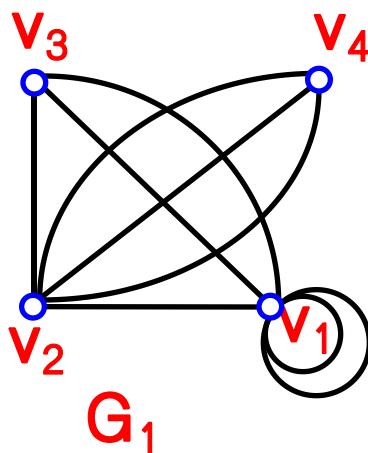


2. 按有无平行边分类

定义5.2.6 在有向图中，两结点间(包括结点自身间)若有**同始点和同终点**的几条边，则这几条边称为**平行边**(Parallel Edge); 在无向图中，两结点间(包括结点自身间)若有几条边，则这几条边称为**平行边**。两结点a、b间相互平行的边的条数称为边(a, b)或a, b的**重数**(Repeated Number)。含有平行边的图称为**多重图**(Multigraph); 非多重图称为**线图**(Line Graph); 无环的线图称为**简单图**(Simple Graph)。

例5.2.7

试判断下图所示的4个图是多重图、线图，还是简单图？并指出多重图中所有平行边的重数。



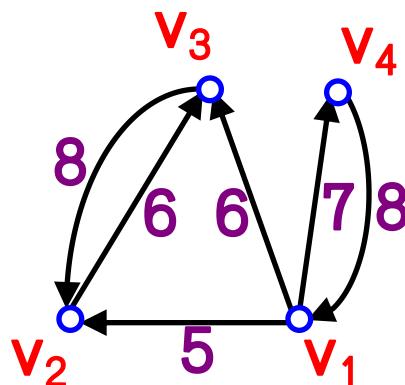
解析 G₁是多重图，仅看有无平行边；G₂是线图，仅看有无平行边；G₃是简单图，仅看有无环。G是简单图。G₁的平行边重数为2，G₂的平行边重数为3，G₃的平行边重数为2，G的平行边重数为3。

3. 按边或结点是否含权分类

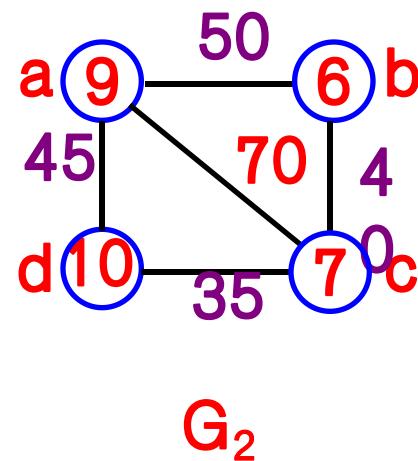
定义5.2.7 赋权图(Weight Graph) G 是一个**三重组** $\langle V, E, g \rangle$ 或**四重组** $\langle V, E, f, g \rangle$ ，其中 **V** 是结点集合， **E** 是边的集合， **f** 是从 **V** 到非负实数集合的**函数**， **g** 是从 **E** 到非负实数集合的**函数**。

例5.2.8

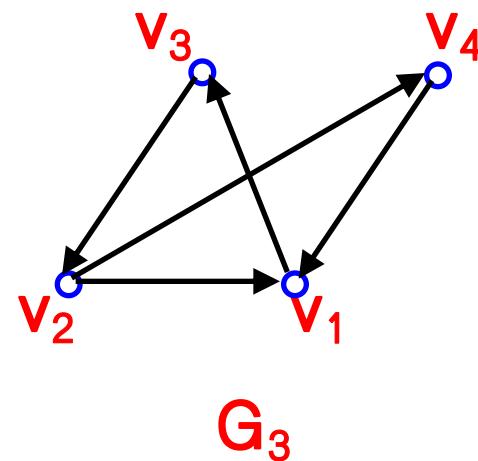
下图所示的图哪个是赋权图，哪个是无权图？是赋权图的请写出相应的函数。



G_1



G_2



G_3

解

分析 中对每条边都赋予了赋权图数值，才是赋权图。而图结点都赋予非负实数值的图就是赋权图，图 G_1 的每条边都赋予了非负实数值，因此图 G_1 是赋权图。图 G_2 的每条边和每个结点都赋予了非负实数值，因此图 G_2 是赋权图。而图 G_3 的边没有赋予非负实数值，因此图 G_3 不是赋权图。

$$g_1(\langle v_3, v_2 \rangle) = 8, \quad g_1(\langle v_4, v_1 \rangle) = 8.$$

$$f_2(a) = 9, \quad f_2(b) = 6, \quad f_2(c) = 7, \quad f_2(d) = 10;$$

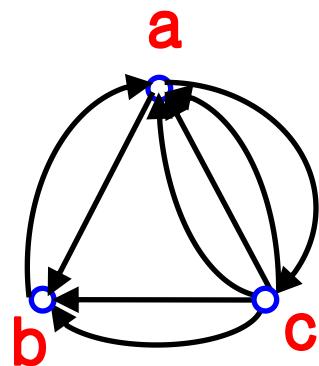
$$g_2((a, b)) = 50, \quad g_2((a, c)) = 70,$$

$$g_2((a, d)) = 45, \quad g_2((b, d)) = 40,$$

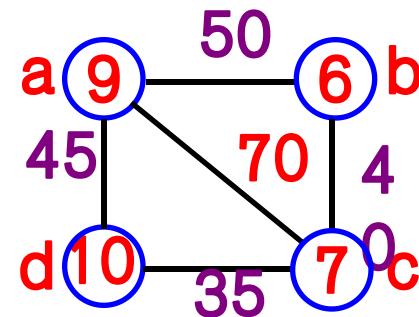
$$g_2((c, d)) = 35,$$

注

我们还可以将上述三种分类方法综合起来对图进行划分。



有向无权多重图



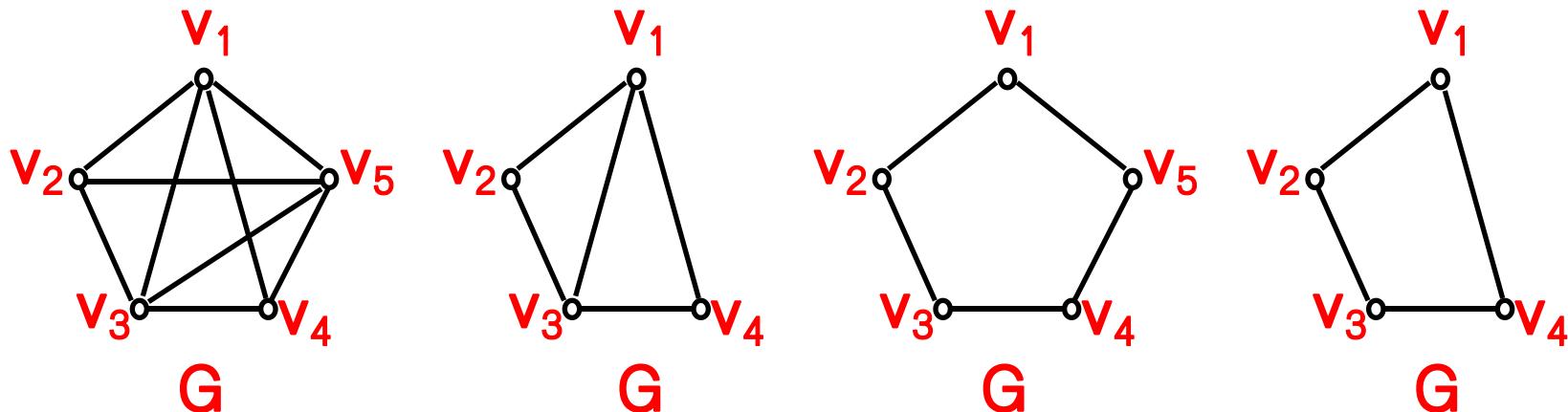
无向赋权简单图

5.2.6 子图与补图

定义5.2.8 设有图 $G = \langle V, E \rangle$ 和图 $G_1 = \langle V_1, E_1 \rangle$ 。

1. 若 $V_1 \subseteq V$, $E_1 \subseteq E$, 则称 G_1 是 G 的**子图**(Subgraph), 记为 $G_1 \subseteq G$ 。
2. 若 $G_1 \subseteq G$, 且 $G_1 \neq G$ (即 $V_1 \neq V$ 或 $E_1 \neq E$), 则称 G_1 是 G 的**真子图**(Proper Subgraph), 记为 $G_1 \subset G$ 。
3. 若 $V_1 = V$, $E_1 \subseteq E$, 则称 G_1 是 G 的**生成子图**(Spanning Subgraph)。
4. 设 $V_2 \subseteq V$ 且 $V_2 \neq \emptyset$, 以 V_2 为结点集, 以两个端点均在 V_2 中的边的全体为边集的 G 的子图, 称为 V_2 导出的 G 的子图, 简称 V_2 的**导出子图**(Induced Subgraph)。

例5.2.9



判断上图中子图G真子图和生成子图需要判断结
点集和边集是否都是图G的子集。子图G₁是图G的真子图，因为它的顶点集{v₁, v₂, v₃, v₄, v₅}是图G的子集，且它的边集{(v₁, v₂), (v₁, v₃), (v₁, v₄), (v₂, v₃), (v₂, v₄), (v₃, v₄), (v₁, v₅)}也是图G的子集。而G₂、G₃和G₄都不是图G的真子图，因为它们的顶点集都不完全包含在图G的顶点集中。例如，G₂的顶点集{v₁, v₂, v₃, v₄}不包含v₅，G₃的顶点集{v₁, v₂, v₃, v₄, v₅}包含了不属于图G的顶点v₅，G₄的顶点集{v₁, v₂, v₄}不包含v₃。因此，只有G₁是图G的真子图。

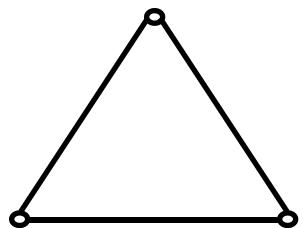
定义5.2.9

设 $G = \langle V, E \rangle$ 为一个具有n个结点的无向简单图，如果G中任意两个结点间都有边相连，则称G为无向完全图(Undirected Complete Graph)，简称G为完全图(Complete Graph)，记为 K_n 。

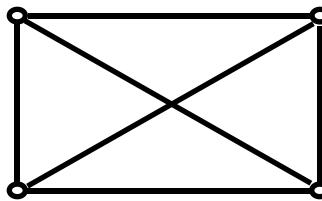
设 $G = \langle V, E \rangle$ 为一个具有n个结点的有向简单图，如果G中任意两个结点间都有两条方向相反的有向边相连，则称G为有向完全图(directed Complete Graph)，在不发生误解的情况下，也记为 K_n 。

对于完全图来说，其邻接矩阵除主对角元为0外，其它元素均为1。

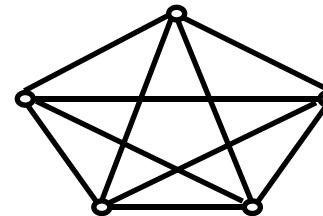
例



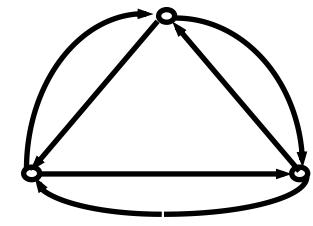
K_3



K_4



K_5



K_3

无向完全图 K_n 的边数为 $C(n,2) = \frac{1}{2}n(n - 1)$

有向完全图 K_n 的边数为 $P(n,2) = n(n - 1)$

定义5.2.10

设 $G = \langle V, E \rangle$ 为简单图， $G' = \langle V, E_1 \rangle$ 为完全图，则称 $G_1 = \langle V, E_1 - E \rangle$ 为 G 的补图 (Complement Graph)，记为 \bar{G} 。

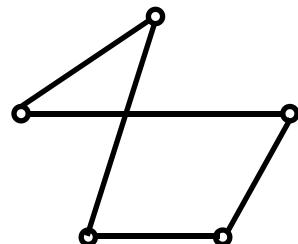
注 在定义5.2.10中，当 G 为有向图时，则 G' 为有向完全图；当 G 为无向图时，则 G' 为无向完全图。

G 的补图也可理解为从结点集 V 的完全图中删除 G 中的边剩下的图，即 G 与其补图 \bar{G} 的结点集是相同的，边集是相对于完全图的边集为全集的补集。

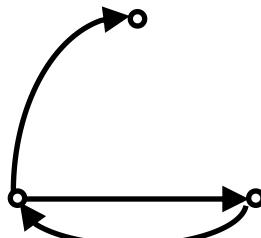
显然，若 $G_1 = \bar{G}$ ，则 $G = \bar{\bar{G}}$ ，即它们互为补图。

例5.2.10

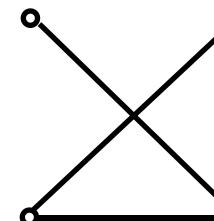
求下图中图(a)、(b)、(c)的补图。



(a)



(b)



(c)

注意，孤立结点一定不要漏了，否则结点集就不同。

分析 上图中图(a)、(b)、(c)的补图分别为

(a)、(b)、(c)对于完全图的边集为全集的补集。具体做的方法，可先补充一些边使其变为完全图，然后再删除原来图中的边得到。

利用邻接矩阵描述补图

若设简单图G的邻接矩阵 $A = (a_{ij})_{n \times n}$, 则它的补图 的邻接矩阵 $= (\bar{a}_{ij})_{n \times n}$ 有:

$$\bar{a}_{ij} = \begin{cases} 1 - a_{ij}, & i \neq j \\ 0, & i = j \end{cases} \quad i, j = 1, 2, 3, \dots, n$$

例5.2.11

证明：在任意6个人的集会上，总会有3个人相互认识或者有3个人互相不认识（假设认识是相互的）。

分析 把6个人作为结点，相互认识的人之间连边，这个问题就转化图的问题。可以利用图及其补图来解这个问题。

证明

把参加集会的人作为结点，相互认识的人之间连边，得到图G，设G'为G的补图，这样问题就转化为证明G或G'中至少有一个完全子图 K_3 。

考虑完全图 K_6 ，结点 v_1 与其余5个结点各有一条边相连，这5条边一定有3条在G或G'中，不妨设有3条边在G中，设这3条边为 (v_1, v_2) 、 (v_1, v_3) 、 (v_1, v_4) 。

考虑结点 v_2, v_3, v_4 。若 v_2, v_3, v_4 在G中无边相连，则 v_2, v_3, v_4 相互不认识；若 v_2, v_3, v_4 在G中至少有一条边相连，例如 (v_2, v_3) ，则 v_1, v_2, v_3 就相互认识。因此，总会有3个人相互认识或者有3个人互相不认识。

5.2.7 结点的度数与握手定理

定义5.2.11 (1) 图 $G = \langle V, E \rangle$ 中以结点 $v \in V$ 为端点的边数(有环时计算两次)称为结点 v 的**度数**(Degree), 简称**度**, 记为 $\deg(v)$.

(2) 有向图 $G = \langle V, E \rangle$ 中以结点 v 为**始点**的边数称为 v 的**出度**(Out-Degree), 记为 $\deg^+(v)$; 以结点 v 为**终点**的边数称为 v 的**入度**(In-Degree), 记为 $\deg^-(v)$ 。显然, $\deg(v) = \deg^+(v) + \deg^-(v)$.

(3) 对于图 $G = \langle V, E \rangle$, 度数为1的结点称为**悬挂结点**(Hanging Point), 以悬挂结点为端点的边称为**悬挂边**(Hanging Edge).

利用邻接矩阵描述

设图 $G = \langle V, E \rangle$, $V = \{v_1, v_2, \dots, v_n\}$ 的邻接矩阵为

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

若 G 是无向图，则 A 中第 i 行元素是由结点 v_i 所关联的边所决定，其中为 1 的元素数目等于 v_i 的度数，即，

$$\deg(v_i) = \sum_{k=1}^n a_{ik} + a_{ii}$$

$$\deg(v_i) = \sum_{k=1}^n a_{ki} + a_{ii}$$

利用邻接矩阵描述

若G是有向图，则A中第*i*行元素是由结点 v_i 为始点的边所决定，其中为1的元素数目等于 v_i 的出度，即

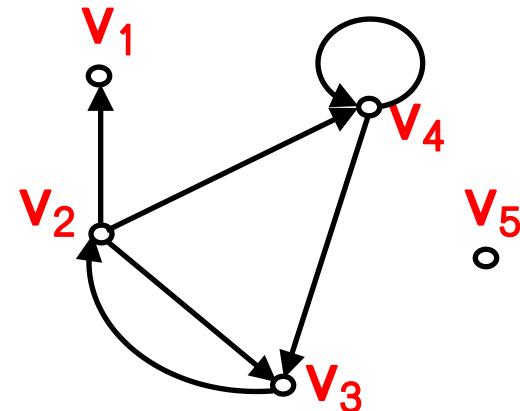
$$\deg^+(v_i) = \sum_{k=1}^n a_{ik}$$

A中第*i*列元素是由结点 v_i 为终点的边所决定，其中为1的元素数目等于 v_i 的入度，即。

$$\deg^-(v_i) = \sum_{k=1}^n a_{ki}$$

例5.2.12

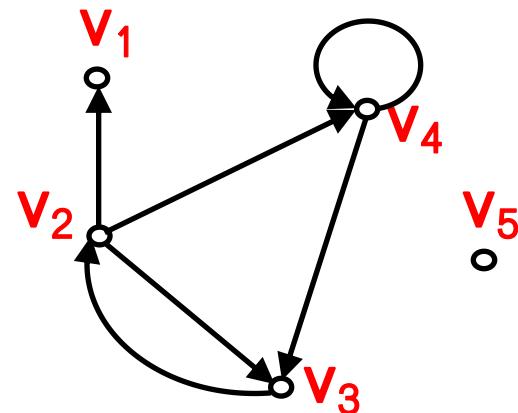
求右图中所有结点的度数、出度和入度，指出悬挂结点和为悬挂边，并用邻接矩阵验证之。



分析 求结点的度数非常简单，只需要数一下以该结点为端点的边数，出度只需要数一下以其为始点的边数，入度只需要数一下以其为终点的边数，无向环算2度，又向环出度和入度各算1度。只有度数为1的才是悬挂结点，以悬挂结点为端点的边才是悬挂边。
悬挂结点 $\langle v_2, v_1 \rangle$ 为悬挂边。

例5.2.12

求右图中所有结点的度数、出度和入度，指出悬挂结点和为悬挂边，并用邻接矩阵验证之。



该图的邻接矩阵为

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

容易验证，上面的计算是正确的。例如A中第二行元素之和为3，即 v_2 的出度为3。

定理5.2.1(握手定理)

图中结点度数的总和等于边数的二倍，即设图 $G = \langle V, E \rangle$ ，则有

$$\sum_{v \in V} \deg(v) = 2|E|$$

分析 由定义5.2.11，结点 v 的度数等于以 v 为端点的边数，而1条边有2个端点(环的2个端点相同)，因此1条边贡献2度。

定理5.2.1(握手定理)

这个结果是图论的第一个定理，它是由欧拉于1736年最先给出的。欧拉曾对此定理给出了这样一个形象论断：如果许多人在见面时握了手，两只手握在一起，被握过手的总次数为偶数。故此定理称为图论的基本定理或握手定理。

证明 因为每条边都有两个端点(环的两个端点相同)，所以加上一条边就使得各结点的度数之和增加2，因此结论成立。

推论5.2.1

图中度数为奇数的结点个数为偶数。

证明 考虑 $|E|$ 为偶数，奇数个奇数之和为奇数且偶数个奇数之和为偶数 $\{v|v \in V \text{ 且 } \deg(v) \text{ 为偶数}\}$ 。

显然， $V_1 \cap V_2 = \emptyset$ ，且 $V_1 \cup V_2 = V$ ，于是

$$\sum_{v \in V} \deg(v) = \sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v) = 2|E|,$$

式中 $2|E|$ 和 $\sum_{v \in V_2} \deg(v)$ （偶数之和为偶数）均为偶数，

因而 $\sum_{v \in V_1} \deg(v)$ 也为偶数。于是 $|V_1|$ 为偶数，即度数

为奇数的结点个数为偶数。

推论5.2.1

今后常称度数为奇数的结点为奇度数结点
(Odd Degree Point), 度数为偶数的结点为偶度数结点(Even Degree Point)。

定理5.2.2

有向图中各结点的出度之和等于各结点的入度之和，
等于边数，即设有向图 $G = \langle V, E \rangle$ ，则有

$$\sum_{v \in V} \deg^+(v) = \sum_{v \in V} \deg^-(v) = |E|$$

证明 因为每条有向边具有一个始点和一个终点（环的始点和终点是同一个结点），因此，每条有向边对应一个出度和一个入度。图 G 中有 $|E|$ 条有向边，则 G 中必产生 $|E|$ 个出度，这 $|E|$ 个出度即为各结点的出度之和， G 中也必产生 $|E|$ 个入度，这 $|E|$ 个入度即为各结点的入度之和。因而，在有向图中，各结点的出度之和等于各结点的入度之和，都等于边数 $|E|$ 。

定理5.2.3

定理：任何简单无向图中点的度数不会互不相同。

证明：(1) 图中没有孤立点，则 n 阶图中点的度数最大为 $n-1$ 。

根据鸽巢原理， n 个点至少有两个点度数一样；

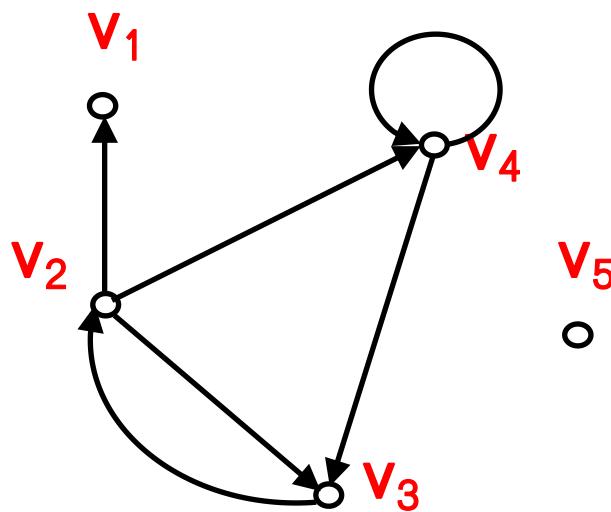
(2) 图中有一个孤立点，则 n 阶图中点的度数最大为 $n-2$ 。根

据鸽巢原理， $n-1$ 个点中至少有两个点度数一样；

(3) 图中有两个孤立点，得证。

定义5.2.12

设 $V = \{v_1, v_2, \dots, v_n\}$ 为图 G 的结点集，称 $(\deg(v_1), \deg(v_2), \dots, \deg(v_n))$ 为 G 的 **度数序列** (Degree Sequence)。



上图的度数序列为 $(1, 4, 3, 4, 0)$ 。

例5.2.14

(1) $(3, 5, 1, 4)$, $(1, 2, 3, 4, 5)$ 能成为图的度数序列吗? 为什么?

(2) 已知图G中有15条边, 2个度数为4的结点, 4个度数为3的结点, 其余结点的度数均小于等于2, 问G中至少有多少个结点? 为什么?

解 分析 由推论3.2.1, 一个图的度数序列中的每一个元素都必须能够成为图的度数序列。利用推论5.2.1, 一定有偶数个奇数。
(2) 图中边数为15, 由握手定理知, G中所有结点的度数之和为30, 2个度数为4的结点, 4个度数为3的结点占去20度, 还剩下10度。若其余全是度数为2的结点, 还需要5个结点来占用这10度, 所以G至少有11个结点。

定理5.2.4

定理：非负整数组 (d_1, d_2, \dots, d_n) 是图的度序列的充分必要条件是： $\sum_{i=1}^n d_i$ 为偶数。

证明：必要条件易证（握手定理）

充分条件证明：按照如下方式作图G：若 d_i 为偶数，则在与之对应的点作 $d_i/2$ 个环；对于剩下的偶数个奇数，两两配对后分别在每配对点间先连一条边，然后在每个顶点画 $(d_j-1)/2$ 个环。该图的度序列就是已知数组。

问题：上述所构造图非简单图，如何判断非负整数组 (d_1, d_2, \dots, d_n) 是简单图的度序列？

可图序列 (Graphic Sequence)

一个非负整数组如果是某简单图的度序列，我们称它为**可图序列**，简称图序列。

关于图序列，主要研究3个问题：

- (1) 存在问题：什么样的整数组是图序列？
- (2) 计数问题：一个图序列对应多少不同构的图？
- (3) 构造问题：如何画出图序列对应的所有不同构图？

研究现状：(1)彻底解决了，(2)解决得不好，(3)没有解决。

可图序列的不同实现

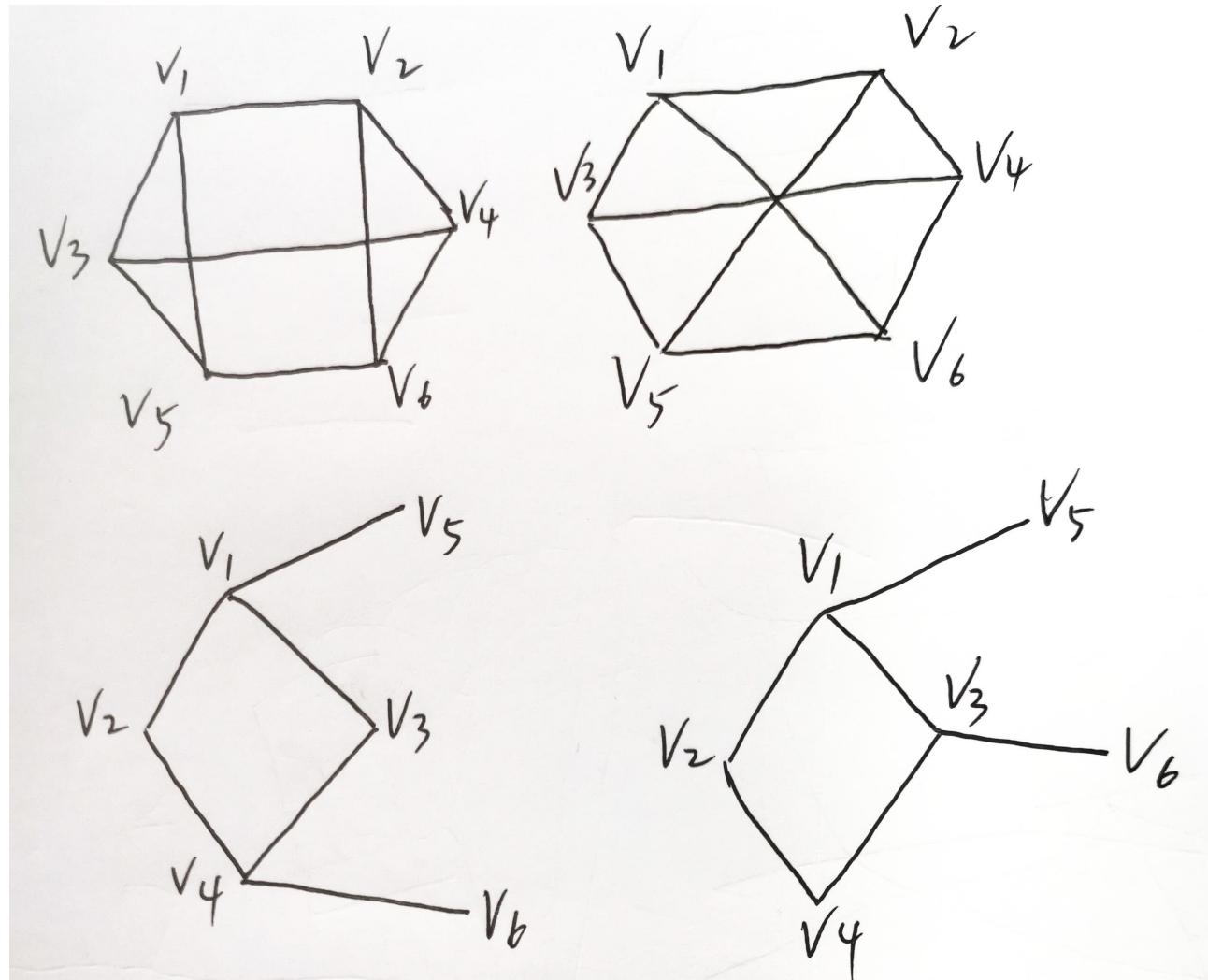
观察：

同一可图序列可有多个不同构的图。

(3,3,3,3,3,3)

与

(3,3,2,2,1,1)



Havel-Hakimi算法

定理5.2.5：非负整数组

$$\pi = (d_1, d_2, \dots, d_n), d_1 \geq d_2 \geq \dots \geq d_n, \sum_{i=1}^n d_i = 2m$$

是图序列的充分必要条件是：

$$\pi_1 = (d_2 - 1, d_3 - 1, \dots, d_{d_1+1} - 1, d_{d_1+2}, \dots, d_n)$$

是图序列。

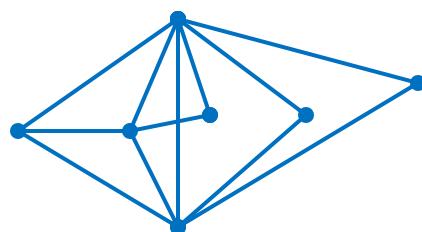
Havel-Hakimi算法举例

问题: $\pi = (6, 5, 4, 3, 2, 2, 2)$ 是否为图序列? 如果是, 作出对应的一个简单图。

解: $\pi_1 = (4, 3, 2, 1, 1, 1)$

$\pi_2 = (2, 1, 0, 0, 1)$

由于 $\pi_2 = (2, 1, 0, 0, 1)$ 是图序列, 所以原序列是图序列。



定理5.2.5证明

证明：必要条件易证（新增点 v_1 与点 v_2, \dots, v_{d_1+1} 分别连接）

充分性证明：设

$$\pi = (d_1, d_2, \dots, d_n), d_1 \geq d_2 \geq \dots \geq d_n, \sum_{i=1}^n d_i = 2m$$

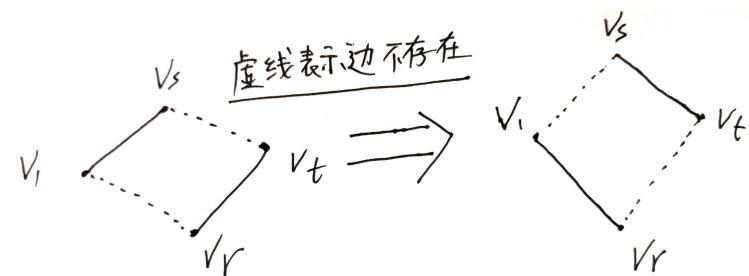
是图序列；现看 v_1 与哪 d_1 个点相连。

(1)如点 v_1 与度数 $d_2, d_3, \dots, d_{d_1+1}$ 的点相连，则得证；

(2)如点 v_1 与任何 $v_s (s > d_1 + 1)$ 相连，可把 v_s 替换成 v_r 与 v_1 连接
($r < d_1 + 2$)

因 $d_r > d_s$ ，则存在点 v_t 与 v_r 相连但不与 v_s 连接；

通过右图变换，不改变新图度序列，
但变 v_1 与 v_r 相连。



5.2.8 图的并交差积联及合成运算

在图论中，将两个或更多的图按照某种方式合并，或者对一个图作某种形式的操作，可以得到很有意义的新图。将图合并或对一个图进行操作，称为图运算。图运算形式很多。

1、图的并运算

设 G_1, G_2 是 G 的两个子图， G_1 与 G_2 并是指由 $V(G_1) \cup V(G_2)$ 为顶点集，以 $E(G_1) \cup E(G_2)$ 为边集组成的子图。记为：

$$G_1 \cup G_2$$

特别是，如果 G_1, G_2 不相交(没有公共顶点)，称它们的并为直接并，可以记为： $G_1 + G_2$

2、图的交运算

设 G_1, G_2 是 G 的两个子图， G_1 与 G_2 交是指由 $V(G_1) \cap V(G_2)$ 为顶点集，以 $E(G_1) \cap E(G_2)$ 为边集组成的子图。记为：

$$G_1 \cap G_2$$

3、图的差运算

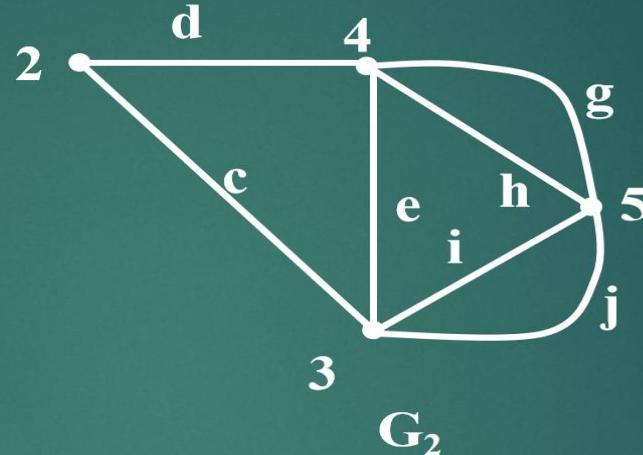
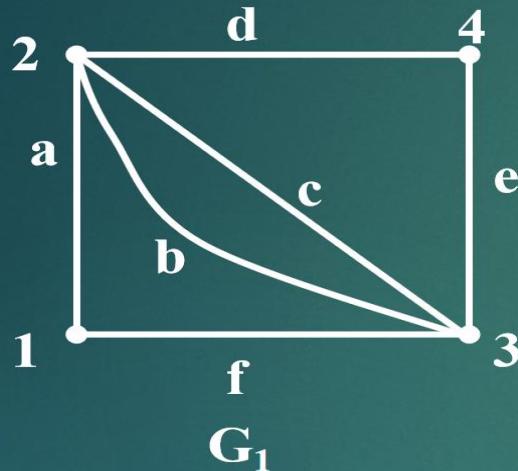
设 G_1, G_2 是两个图， G_1 与 G_2 的差是指从 G_1 中删去 G_2 中的边得到的新图。记为 $G_1 - G_2$.

4、图的对称差运算(或环和运算)

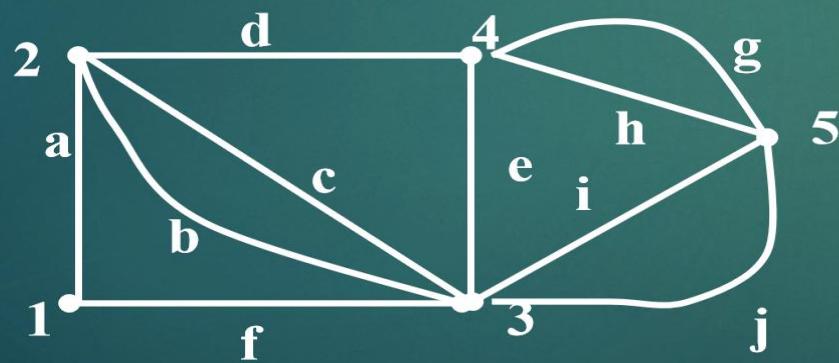
设 G_1, G_2 是两个图， G_1 与 G_2 的对称差定义为：

$$G_1 \Delta G_2 = (G_1 \cup G_2) - (G_1 \cap G_2)$$

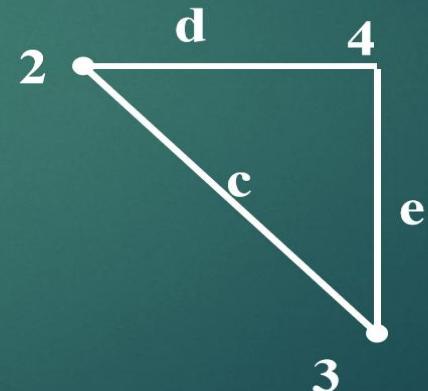
已知 G_1 与 G_2 , 求 $G_1 \cup G_2$ $G_1 \cap G_2$ $G_1 - G_2$ $G_2 - G_1$ $G_1 \Delta G_2$



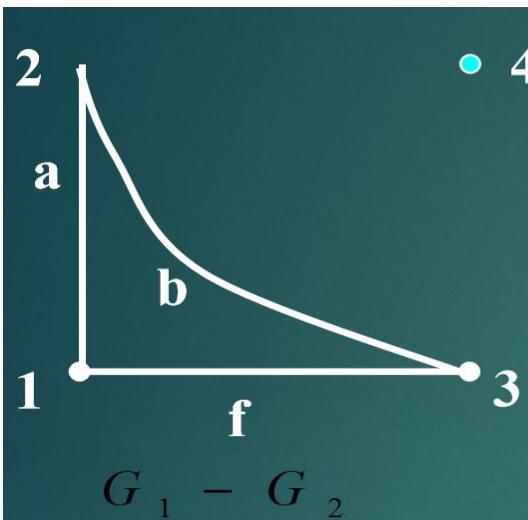
解：由相应运算定义得下面结果：



$G_1 \cup G_2$



$G_1 \cap G_2$



2

$\bullet\; 4$

2

\bullet

1

$G_1 - G_2$

2

a

1

f

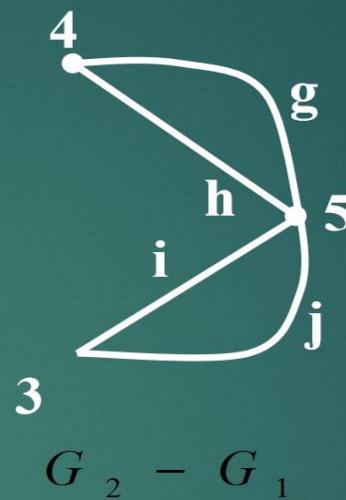
$\bullet\; 4$

$\bullet\; g$

h

$\bullet\; i$

j

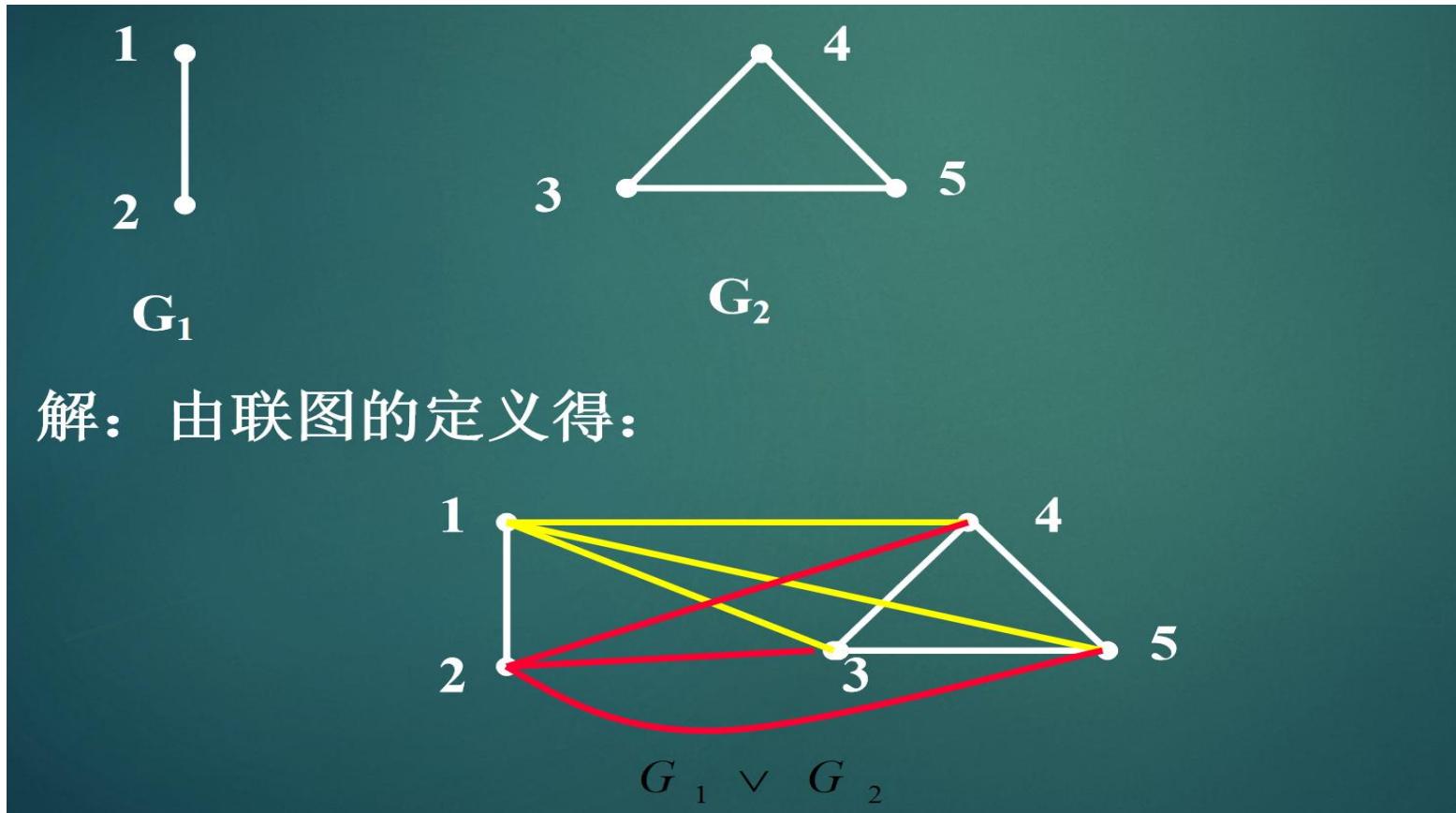


$G_2 - G_1$

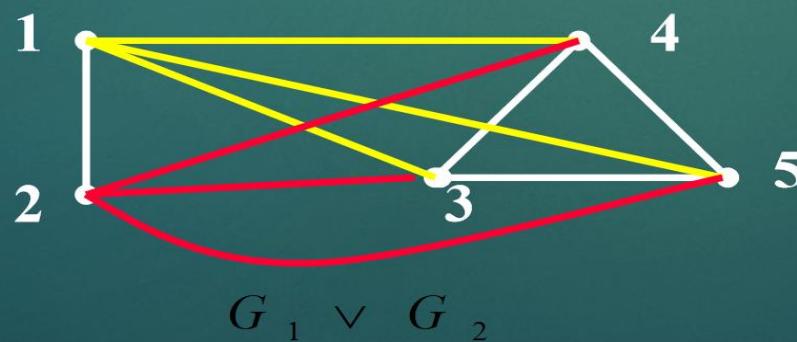
5、图的联运算

设 G_1, G_2 是两个不相交的图，作 G_1+G_2 ，并且将 G_1 中每个顶点和 G_2 中的每个顶点连接，这样得到的新图称为 G_1 与 G_2 的联图。记为：

$$G_1 \vee G_2$$

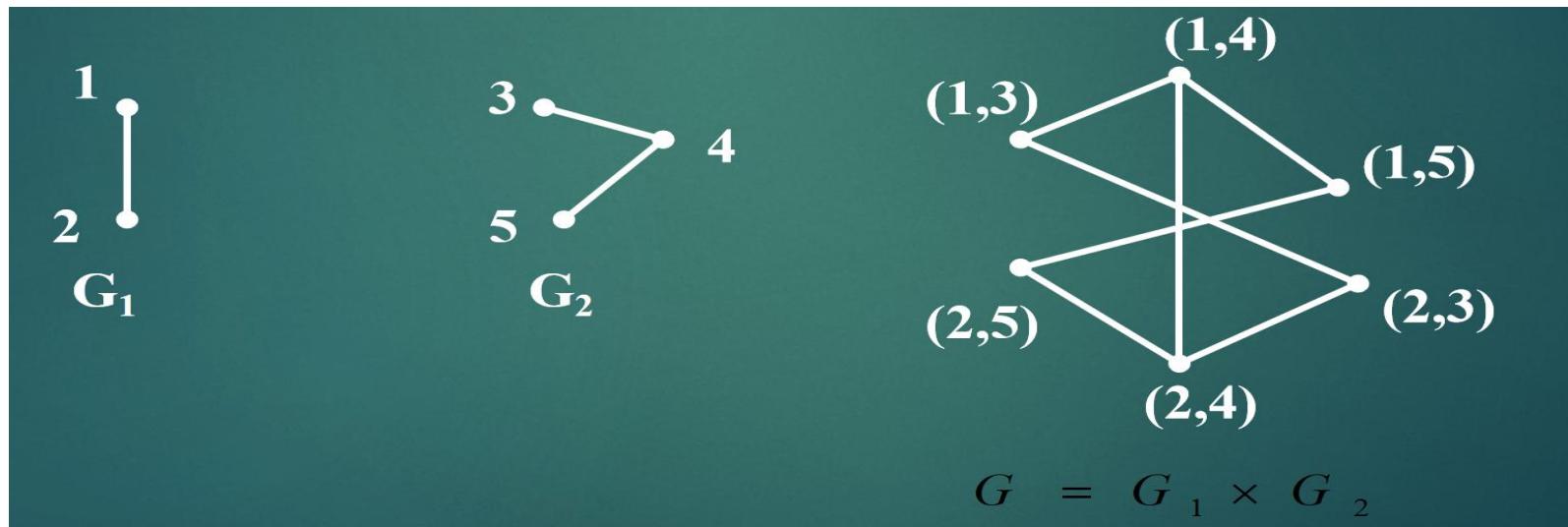


解：由联图的定义得：



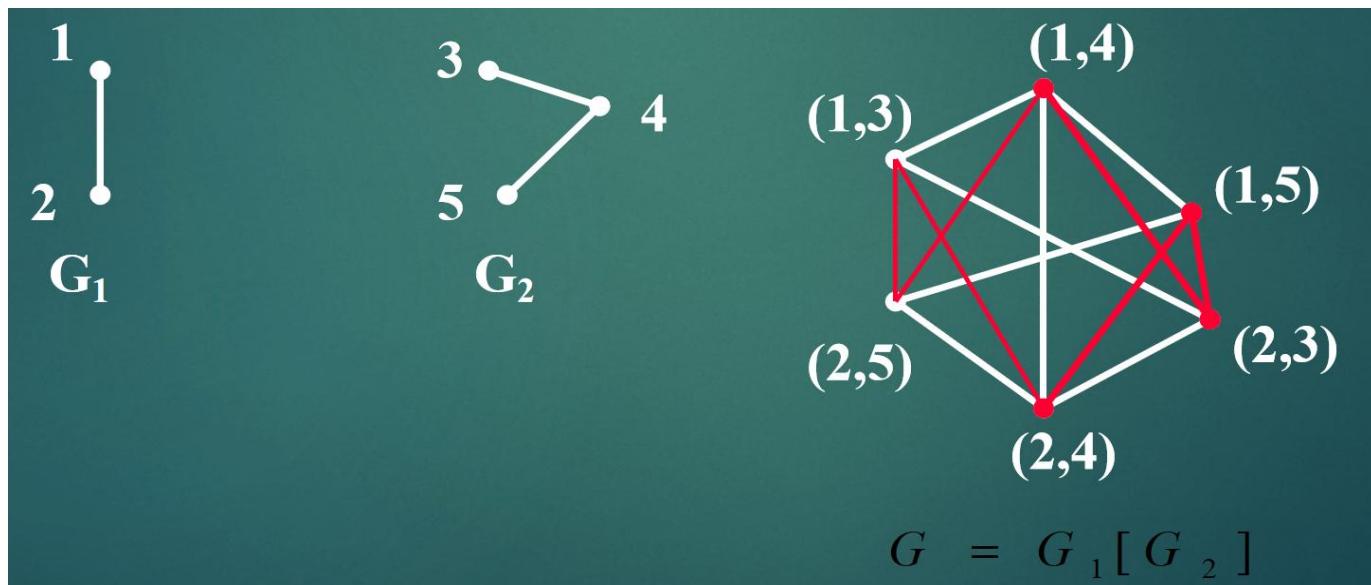
6、图的积图

设 $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, 是两个图。对点集
 $V = V_1 \times V_2$ 的任意两个点 $u=(u_1, u_2)$ 与 $v=(v_1, v_2)$, 当($u_1=v_1$ 和 $u_2 \text{ adj } v_2$)
或($u_2=v_2$ 和 $u_1 \text{ adj } v_1$)时, 把 u 与 v 相连。如此得到的新图称为 G_1 与 G_2 的
积图。记为 $G = G_1 \times G_2$



7、图的合成图

设 $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, 是两个图。对点集
 $V = V_1 \times V_2$ 的任意两个点 $u=(u_1, u_2)$ 与 $v=(v_1, v_2)$, 当 $(u_1 \text{ adj } v_1)$
或 $(u_1=v_1 \text{ 和 } u_2 \text{ adj } v_2)$ 时, 把 u 与 v 相连。如此得到的新图称为 G_1 与 G_2 的
合成图。记为 $G = G_1 [G_2]$

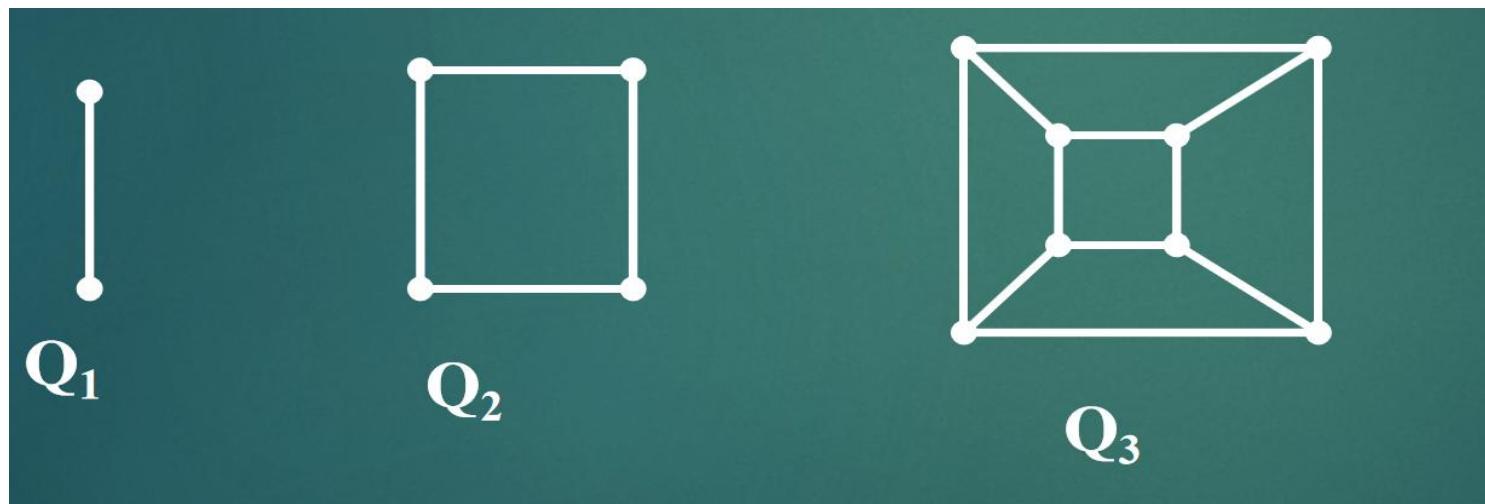


图的积运算是网络构造的常用方法。并行计算机中的网络拓扑常采用所谓的“超立方体”结构。采用该结构可使网络具有较好的可靠性、较小的通信延迟和很好的可扩展性以及便于并行编程等优点。

“超立方体”可以采用积图来递归构造。定义如下：

(1) 1方体 $Q_1 = K_2$

(2) n方体定义为： $Q_n = K_2 \times Q_{n-1}$



“超立方体”常采用下面简单的递归构造方法：

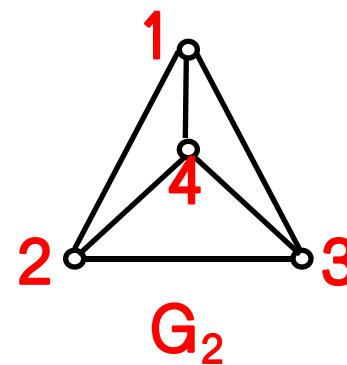
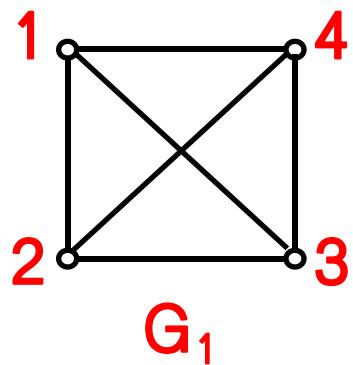
n 方体 Q_n 的顶点可用一个长度为 n 的二进制码来表示。 Q_n 的顶点数目正好等于 2^n 个。

由 $n-1$ 方体 Q_{n-1} 构造 Q_n 的方法是：将 Q_{n-1} 拷贝一个。将原 Q_{n-1} 每个顶点的码前再添加一个零，将拷贝得来的 $n-1$ 方体每个顶点的码前面再添加一个 1。然后在两个 $n-1$ 方体之间连线：当且仅当两个顶点码只有一位对应位数字不同时，该两点连线。如此得到的图即为 n 方体。

5.2.9 图的同构

图是表达事物之间关系的工具，因此，图的最本质的内容是结点和边的关联关系。而在实际画图时，由于结点的位置不同，边的长短曲直不同，同一事物间的关系可能画出不同形状的图来。

例如下图中的两个图 G_1 和 G_2 实际上是同一个图 K_4 。



定义5.2.13

设两个图 $G = \langle V, E \rangle$ 和 $G' = \langle V', E' \rangle$, 如果存在**双射**函数 $g: V \rightarrow V'$, 使得对于任意的 $e = (v_i, v_j)$ (或者 $\langle v_i, v_j \rangle \in E$) 当且仅当 $e' = (g(v_i), g(v_j))$ (或者 $\langle g(v_i), g(v_j) \rangle \in E'$), 并且 e 与 e' 的**重数相同**, 则称 G 与 G' **同构**(Isomorphism), 记为 $G \cong G'$ 。

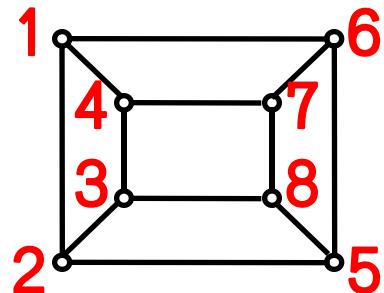
对于同构, 形象地说, 若图的结点可以任意挪动位置, 而边是完全弹性的, 只要在不拉断的条件下, 一个图可以变形为另一个图, 那么这两个图是同构的。

两个图同构的必要条件

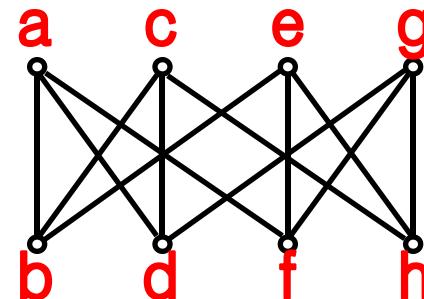
- (1) 结点数目相同;
- (2) 边数相同;
- (3) 度数相同的结点数相同。

例5.2.14

试证明下图中， $G \cong G'$ 。



G



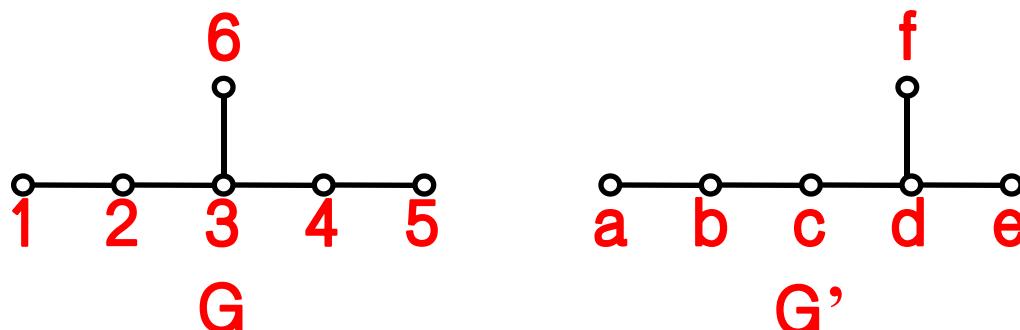
G'

证明 欲证两图同构，双射函数做到满足要求的结点集之间的双射函数。现在还没有好的办法，只有凭经验去试。
令 $f(1)=a, f(2)=b, f(3)=c, f(4)=d, f(5)=e, f(6)=f, f(7)=g, f(8)=h$

容易验证， f 满足定义5.2.13，所以 $G \cong G'$ 。

例5.2.15

证明下图中， G 与 G' 不同构



分析 证明两个图不同构，通常用反证法。

证明 假设 $G \cong G'$ ，双射函数为 f 。由定义5.2.13， v 与 $f(v)$ 的度数一定相同，因此有 $f(3)=d$ 。 G 中3与一个度数为1的结点6邻接，而 G' 中d与两个度数为1的结点e、f邻接，矛盾。

例5.2.15

注意 图同构的三个必要条件不是充分条件。在上图的 G 与 G' 两个图，虽然满足以上三个条件，但不同构。

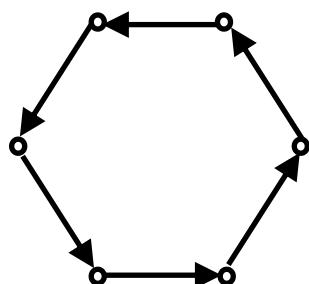
寻找一种简单而有效的方法来判断图的同构，是图论中一个重要而未解决的问题。

5.2.10 图的难点

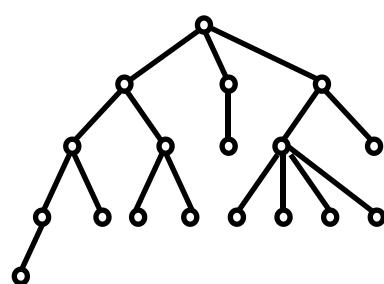
- 图是由两个集合构成的，可以利用集合的有关知识来研究它，如子图、完全图、补图等；
- 图的计算机表示就是它的邻接矩阵，实际中的图都是很大的，可能有成千上万的结点和边，用手工处理是很难想象的；
- 判断图的同构，是图论中一个重要而未解决的问题。现在还没有好的办法，只有凭经验按定义去试；
- 握手定理是图论的基本定理，很多理论都是以它为基础的，必须熟练掌握，并能灵活运用。

5.2.11 图的应用-通讯网络

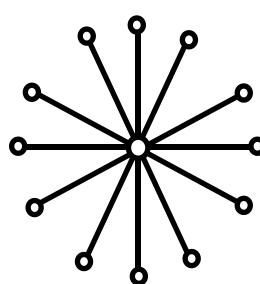
通讯网络中最重要的整体问题之一是网络的结构形式。通讯网络是一个强连通的有向图，根据用途和各种性能指标有着不同的结构形式，下图给出了一些典型的结构。



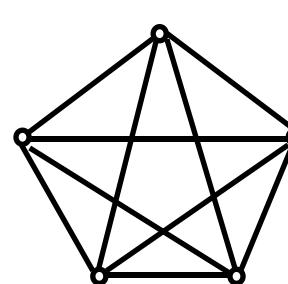
(a)



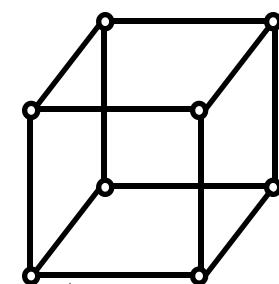
(b)



(c)



(d)



(e)

环型网络 (Ring Network)

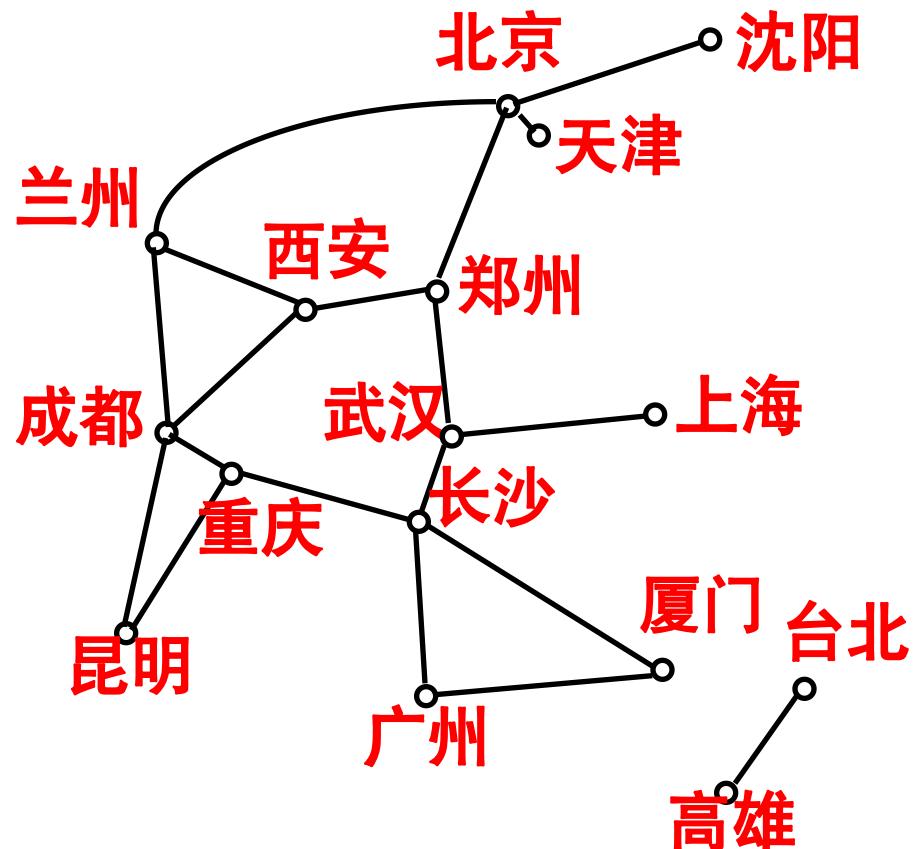
树型网络 (Tree Network)

分布式 (Distributed) 网络

立方体 (Cube) 型网络

5.3 通路、回路与连通性

右图是中国铁路交通图的一部分，如果一个旅客要从成都乘火车到北京，那么他一定会经过其它车站；而旅客不可能从成都乘火车到达台北。这就引出了图的通路与连通的概念。



5.3.1 通路与回路

通路与回路是图论中两个重要的基本概念。本小节所述定义一般来说既适合有向图，也适合无向图，否则，将加以说明或分开定义。

定义5.3.1

给定图 $G = \langle V, E \rangle$ 中结点和边相继交错出现的序列

$$\Gamma = v_0 e_1 v_1 e_2 v_2 \dots e_k v_k.$$

- 若 Γ 中边 e_i 的两端点是 v_{i-1} 和 v_i (G 是有向图时要求 v_{i-1} 与 v_i 分别是 e_i 的始点和终点)， $i=1,2,\dots,k$ ， 则称 Γ 为结点 v_0 到结点 v_k 的通路(Entry)。

v_0 和 v_k 分别称为此通路的始点和终点，统称为通路的端点。

通路中边的数目 k 称为此通路的长度(Length)。

当 $v_0 = v_n$ 时，此通路称为回路(Circuit)。

定义5.3.1

2. 若通路中的所有边互不相同，则称此通路为简单通路(Simple Entry)或一条迹；若回路中的所有边互不相同，则称此回路为简单回路(Simple Circuit)或一条闭迹。
3. 若通路中的所有结点互不相同（从而所有边互不相同），则称此通路为基本通路(Basic Entry)或者初级通路、路径；若回路中除 $v_0=v_k$ 外的所有结点互不相同（从而所有边互不相同），则称此回路为基本回路(Basic Circuit)或者初级回路、圈。

说明

1. 回路是通路的特殊情况。因而，我们说某条通路，它可能是回路。但当我们说一基本通路时，一般是指它不是基本回路的情况。
2. 基本通路（回路）一定是简单通路（回路），但反之不真。因为没有重复的结点肯定没有重复的边，但没有重复的边不能保证一定没有重复的结点。
3. 在不会引起误解的情况下，一条通路 $v_0e_1v_1e_2v_2\dots e_nv_n$ 也可以用边的序列 $e_1e_2\dots e_n$ 来表示，这种表示方法对于有向图来说较为方便。在线图中，一条通路 $v_0e_1v_1e_2v_2\dots e_nv_n$ 也可以用结点的序列 $v_0v_1v_2\dots v_n$ 来表示。

例5.3.1

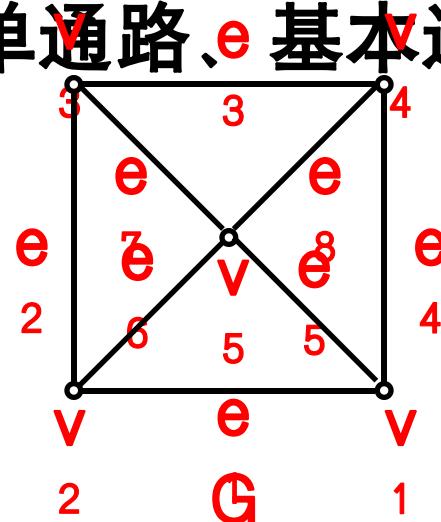
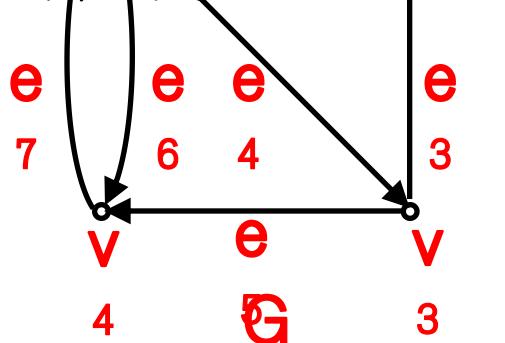
判断下图G₁中的回路 $v_3e_5v_4e_7v_1e_4v_3e_3v_2e_1v_1e_4v_3$ 、 $v_3e_3v_2e_2v_2e_1v_1e_4v_3$ 、 $v_3e_3v_2e_1v_1e_4v_3$ 是否是简单回路、基本回路？图G₂中的通路

$v_1e_1v_2e_6v_5e_7v_3e_2v_2e_6$

$v_5e_8v_4$ 、 $v_1e_5v_5e_7v_3e_2v_2e_6v_5e_8v_4$ 、

$v_1e_1v_2e_6v_5e_7v_3e_2v_2e_6$ 是否是简单通路、 e 基本通路？

并求其长度。



分析

判断一条通（回）路是否是简单通（回）路、基本通（回）路，主要是看它有无重复的边、结点。在图 G_1 中， $v_3e_5v_4e_7v_1e_4v_3e_3v_2e_1v_1e_4v_3$ 中有重复的边 e_4 ，因此它不是简单回路，也不可能的基本回路； $v_3e_3v_2e_2v_2e_1v_1e_4v_3$ 虽然没有重复的边，但有重复的结点 v_2 ，因此只能是简单回路，而不是基本回路；而 $v_3e_3v_2e_1v_1e_4v_3$ 中既没有重复的边，也没有重复的结点，因此既是基本回路，也是简单回路；在图 G_2 中， $v_1e_1v_2e_6v_5e_7v_3e_2v_2e_6v_5e_8v_4$ 中有重复的边 e_6 ，因此它不是既不是简单通路，也不是基本通路； $v_1e_5v_5e_7v_3e_2v_2e_6v_5e_8v_4$ 虽然没有重复的边，但有重复的结点 v_5 ，因此只能是简单通路，但不是基本通路； $v_1e_1v_2e_6v_5e_7v_3e_3v_4$ 中既没有重复的边，也没有重复的结点，因此既是基本通路，也是简单通路。至于通（回）路的长度就是其包含的边的数目，这只需要数一数就行了。

解

在图G₁中， $v_3e_5v_4e_7v_1e_4v_3e_3v_2e_1v_1e_4v_3$ 是一条长度为6的回路，但既不是简单回路，也不是基本回路； $v_3e_3v_2e_2v_2e_1v_1e_4v_3$ 是一条长度为4的简单回路，但不是基本回路； $v_3e_3v_2e_1v_1e_4v_3$ 是一条长度为3的基本回路，也是简单回路；在图G₂中， $v_1e_1v_2e_6v_5e_7v_3e_2v_2e_6v_5e_8v_4$ 是一条长度为6的通路，但既不是简单通路，也不是基本通路； $v_1e_5v_5e_7v_3e_2v_2e_6v_5e_8v_4$ 是一条长度为5的简单通路，但不是基本通路； $v_1e_1v_2e_6v_5e_7v_3e_3v_4$ 是一条长度为4的基本通路，也是简单通路。

说明

在图 G_1 中，简单回路 $v_3e_3v_2e_2v_2e_1v_1e_4v_3$ 既可以用边的序列 $e_3e_2e_1e_4$ 来表示，也可以用结点的序列 $v_3v_2v_2v_1v_3$ 来表示；

在图 G_2 中，简单通路 $v_1e_5v_5e_7v_3e_2v_2e_6v_5e_8v_4$ 既可以用边的序列 $e_5e_7e_2e_6e_8$ 来表示，也可以用结点的序列 $v_1v_5v_3v_2v_5v_4$ 来表示。

定理5.3.1

设 $G = \langle V, E \rangle$ 为线图, $V = \{v_1, v_2, \dots, v_n\}$, $A = (a_{ij})_{n \times n}$ 为 G 的邻接矩阵, $A_m a_{ij}^{(m)} \rangle_{n \times n}$ 。则

- $a_{ij}^{(m)}$ 为从结点 v_i 到结点 v_j 长度为 m 的通路数目;
- $a_{ii}^{(m)}$ 为结点 v_i 到自身的长度为 k 的回路数目;
- $\sum_{i=1}^n \sum_{j=1}^n a_{ij}^{(m)}$ 为 G 中长度为 m 的通路 (含回路) 总数。

分析

观察图G的邻接矩阵 $A = (a_{ij})_{n \times n}$, 我们发现, a_{ij} 表示从结点 v_i 到结点 v_j 长度为 1 的通路数目, 而 A 中所有元素之和为 A 中长度为 1 的通路 (包括回路) 数目 (若 G 是有向图, 它也是边的数目; 若 G 是无向图, 它是边的数目的二倍减去 G 中自回路的数目, 因为当 $v_i \neq v_j$ 时, 一条边 (v_i, v_j) 即是一条从 v_i 到 v_j 的长度为 1 的通路, 也是一条从 v_j 到 v_i 的长度为 1 的通路, 而 (v_i, v_i) 只是一条长度为 1 的通路, 而不能再看作两条)。

分析(续1)

下面寻找G中长度为2的通路（包含回路）数目。首先计算从结点 v_i 到结点 v_j 的长度为2的通路数目。注意到从 v_i 到 v_j 长度为2的通路，中间必经过一结点 v_k 。对于任意的 $k(1 \leq k \leq n)$ ，若存在通路 $v_i v_k v_j$ ，必有 $a_{ik} = 1$ 且 $a_{kj} = 1$ ，即 $a_{ik} \cdot a_{kj} = 1$ 。反之，若不存在通路 $v_i v_k v_j$ ，则必有 $a_{ik} = 0$ 或 $a_{kj} = 0$ ，即 $a_{ik} \cdot a_{kj} = 0$ 。于是从结点 v_i 到 v_j 长度为2的通路总数为：

$$a_{11}a_{1j} + a_{12}a_{2j} + \cdots + a_{in}a_{nj} = \sum_{k=1}^n a_{ik}a_{kj}$$

分析(续2)

由矩阵的乘法规则可知, $\sum_{k=1}^n a_{ik} a_{kj}$ 恰为 A^2 中第 i 行第 j 列的元素。因而在矩阵

$$\left(a_{ij}^{(2)}\right)_{n \times n} = A^2 = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \times \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

中, $\sum_{i=1}^n \sum_{j=1}^n a_{ij}^{(2)}$ 为 G 中长度为 2 的通路 (含回路) 总数,
主对角线上元素之和 $\sum_{i=1}^n a_{ii}^{(2)}$ 为 G 中长度为 2 的回路总数。

分析(续3)

类似地，若从结点 v_i 到结点 v_j 存在长度为3的通路，中间必经过一结点 v_k ，使得从 v_i 到 v_k 存在长度为1的通路，从 v_k 到 v_j 存在长度为2的通路。因而， $a_{ik}=1$ 且 $a_{kj}^{(2)} \geq 1$ ，即 $a_{ik} a_{kj}^{(2)} \geq 1$ 。若 $a_{ik}=0$ 或 $a_{kj}^{(2)} = 0$ ，从而 $a_{ik} a_{kj}^{(2)} = 0$ ，则从 v_i 经过 v_k 到 v_j 没有长度为3的通路。于是从结点 v_i 到 v_j 长度为3的通路总数为：

$$a_{i_1} a_{1j}^{(2)} + a_{i_2} a_{2j}^{(2)} + \cdots + a_{in} a_{nj}^{(2)} = \sum_{k=1}^n a_{ik} a_{kj}^{(2)}$$

分析(续4)

这正是 $A \cdot A^2 = A^3$ 中第*i*行第*j*列的元素 $a_{ij}^{(3)}$ 。

而 $\sum_{i=1}^n \sum_{j=1}^n a_{ij}^{(3)}$ 为G中长度为3的通路（含回路）总数，
主对角线上元素之和 $\sum_{i=1}^n a_{ii}^{(3)}$ 为G中长度为3的回路
总数。

证明

对m用数学归纳法。

(1) 当 $m = 1$ 时，显然成立。

(2) 设 $m = k$ 时，定理成立。

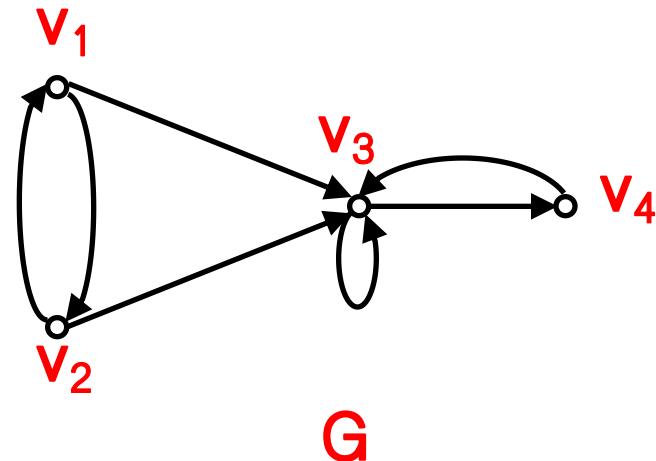
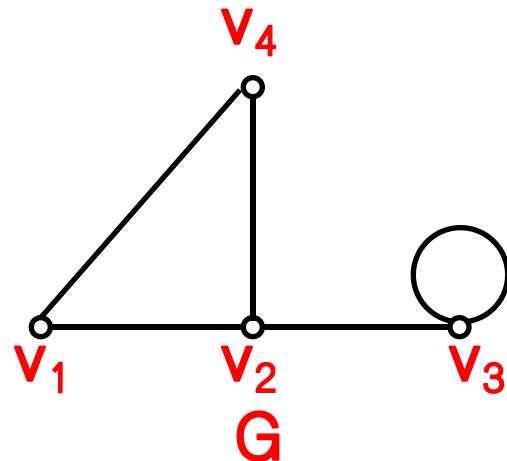
(3) 证明 $m = k + 1$ 时定理成立。

因为 $(a_{ij}^{(k+1)})_{n \times n} = A^{k+1} = A \cdot A^k = \left(\sum_{p=1}^n a_{ip} \cdot a_{pj}^{(k)} \right)_{n \times n}$ ，故 $a_{ij}^{(k+1)} = \sum_{p=1}^n a_{ip} \cdot a_{pj}^{(k)}$

而 a_{ip} 是结点 v_i 到 v_p 长度为1的通路数目， $a_{pj}^{(k)}$ 是结点 v_p 到 v_j 长度为 k 的通路数目，故 $a_{ip} \cdot a_{pj}^{(k)}$ 是从结点 v_i 经过 v_p 到结点 v_j 的长度为 $k + 1$ 的通路数目，那么 $\sum_{p=1}^n a_{ip} \cdot a_{pj}^{(k)}$ 是从结点 v_i 到结点 v_j 的长度为 $k + 1$ 的通路数目。

例5.3.2

求下图中图 G_1 和 G_2 的从结点 v_1 到结点 v_3 长度为2和3的通路数目及所有长度为2和3的通路数目。



分析 利用定理5.3.1¹，求图中长度为 m 的通路数目，只需要先写出图的邻接矩阵²，然后计算邻接矩阵的 m 次方即可。

解

在图中， G_1 是无向线图， G_2 是有向线图，它们的邻接矩阵分别为：

$$A(G) = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

$$A(G_2) = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

解(续1)

下面计算邻接矩阵的幂,

$$(A(G))^2 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 3 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix}$$

$$a_{13}^{(2)} = 1$$

$$\sum_{i=1}^4 \sum_{j=1}^4 a_{ij}^{(2)} = 21$$

$$\sum_{i=1}^4 a_{ii}^{(2)} = 9$$

因而 G_1 中从结点 v_1 到结点 v_3 长度为2通路数目为1,
长度为2的通路 (含回路) 总数为21, 其中9条为回路。

解(续2)

$$(A(G_2))^2 = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

$$a_{13}^{(2)} = 2$$

$$\sum_{i=1}^4 \sum_{j=1}^4 a_{ij}^{(2)} = 13$$

$$\sum_{i=1}^4 a_{ii}^{(2)} = 4$$

G_2 中从结点 v_1 到结点 v_3 长度为2通路数目为2，长度为2的通路（含回路）总数为13，其中5条为回路。

解(续3)

$$(A(G))^3 = A(G) \cdot (A(G))^2 = \begin{pmatrix} 2 & 4 & 2 & 3 \\ 4 & 3 & 4 & 4 \\ 2 & 4 & 3 & 2 \\ 3 & 4 & 2 & 2 \end{pmatrix}$$

$$a_{13}^{(3)} = 2 \quad \sum_{i=1}^4 \sum_{j=1}^4 a_{ij}^{(3)} = 48 \quad \sum_{i=1}^4 a_{ii}^{(3)} = 10$$

因而 G_1 中从结点 v_1 到结点 v_3 长度为3的通路数目为2, 长度为3的通路(含回路)总数为48, 其中10条为回路。

解(续4)

$$(A(G_2))^3 = A(G_2) \cdot (A(G_2))^2 = \begin{pmatrix} 0 & 1 & 4 & 2 \\ 1 & 0 & 4 & 2 \\ 0 & 0 & 3 & 2 \\ 0 & 0 & 2 & 1 \end{pmatrix}$$

$$a_{13}^{(3)} = 4$$

$$\sum_{i=1}^4 \sum_{j=1}^4 a_{ij}^{(3)} = 22$$

$$\sum_{i=1}^4 a_{ii}^{(3)} = 4$$

G_2 中从结点 v_1 到结点 v_3 长度为3的通路数目为4，
长度为3的通路（含回路）总数为22，其中4条为
回路。

定义5.3.2

在图 $G = \langle V, E \rangle$ 中, $v_i, v_j \in V$ 。

1. 如果从 v_i 到 v_j 存在通路, 则称 v_i 到 v_j 是可达的, 否则称 v_i 到 v_j 不可达。规定: 任何结点到自己都是可达的。
2. 如果 v_i 到 v_j 可达, 则称长度最短的通路为从 v_i 到 v_j 的短程线(Geodesic);
3. 从 v_i 到 v_j 的短程线的长度称为从 v_i 到 v_j 的距离(Distance), 记为 $d(v_i, v_j)$ 。如果 v_i 到 v_j 不可达, 则通常记为 $d(v_i, v_j) = \infty$ 。

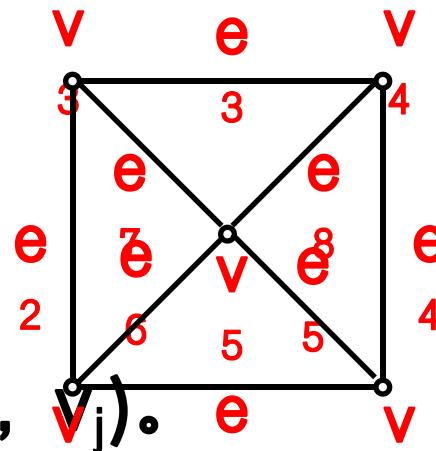
说明

$d(v_i, v_j)$ 满足下列性质：

(1) $d(v_i, v_j) \geq 0$.

(2) $d(v_i, v_i) = 0$;

(3) $d(v_i, v_k) + d(v_k, v_j) \geq d(v_i, v_j)$.



对于无向图，一定有若 v_i 到 v_j 可达，则 v_j 到 v_i 可达；也有 $d(v_i, v_j) = d(v_j, v_i)$ 。

对于有向图， v_i 到 v_j 可达，不一定有 v_j 到 v_i 可达；也不一定有 $d(v_i, v_j) = d(v_j, v_i)$ 。

例如在上图 G_1 中， $d(v_1, v_2) = 2$, $d(v_2, v_1) = 1$, $d(v_4, v_1) = d(v_1, v_4) = 1$, $d(v_2, v_4) = 2$, $d(v_4, v_2) = 3$; 在上图 G_2 中， $d(v_1, v_3) = 2$, $d(v_3, v_4) = 1$, $d(v_2, v_4) = 2$ 。

定理5.3.2

在一个具有 n 个结点的图中，如果从结点 v_i 到结点 v_j ($v_i \neq v_j$) 存在一条通路，则从 v_i 到 v_j 存在一条长度不大于 $n-1$ 的通路。

分析 通路的长度为序列中的结点数减1，如果结点不重复，最多 n 个，因此通路长度最多 $n-1$ ；如果结点有重复，则在重复的结点间构成一条回路，删除这条回路，剩下的仍然是从结点 v_i 到结点 v_j 的通路。一直删下去，直到无重复结点为止，这样定理就得证了。

证明

设 $v_{i_0} v_{i_1} \cdots v_{i_k}$ 为从 v_i 到 v_j 的长度为 k 的一条通路，其中 $v_{i_0} = v_i$, $v_{i_k} = v_j$ ，此通路上有 $k+1$ 个结点。若 $k \leq n-1$ ，这条通路即为所求。若 $k > n-1$ ，则此通路上的结点数 $k+1 > n$ ，由鸽笼原理知，必存在一个结点在此通路中不止一次出现，设 $v_{i_s} = v_{i_t}$ ，其中， $0 \leq s < t \leq k$ 。去掉 v_{i_s} 到 v_{i_t} 中间的通路，至少去掉一条边，得通路 $v_{i_0}, v_{i_1}, \dots, v_{i_s}, v_{i_{t+1}}, \dots, v_{i_k}$ ，此通路比原通路的长度至少少 1。如此重复进行下去，必可得一条从 v_i 到 v_j 的长度不大于 $n-1$ 的通路。

几个结论

推论5.3.1 在一个具有 n 个结点的图中，如果从结点 v_i 到结点 v_j ($v_i \neq v_j$) 存在一条通路，则从 v_i 到 v_j 存在一条长度不大于 $n-1$ 的基本通路。

定理5.3.3 在一个具有 n 个结点的图中，如果存在经过结点 v_i 回路，则存在一条经过 v_i 的长度不大于 n 的回路。

推论5.3.2 在一个具有 n 个结点的图中，如果存在经过结点 v_i 回路，则存在一条经过 v_i 的长度不大于 n 的基本回路。

利用邻接矩阵判断可达

利用定理5.3.2和定理5.3.3，我们可以通过计算图的邻接矩阵及其幂的方法来判断从 v_i 到 v_j 是否可达，以及从 v_i 到 v_j 的距离。

设矩阵 $B_n = A + A^2 + A^3 + \dots + A^n$

则 B^n 中的元素

$$b_{ij}^{(n)} = a_{ij}^{(1)} + a_{ij}^{(2)} + \dots + a_{ij}^{(n)} = \sum_{m=1}^n a_{ij}^{(m)}$$

表示图G中从结点 v_i 到结点 v_j 的长度小于等于n的通路总数，若 $i = j$ ， $b_{ii}^{(n)}$ 为G中结点 v_i 到自身的长度小于等于n的回路总数。

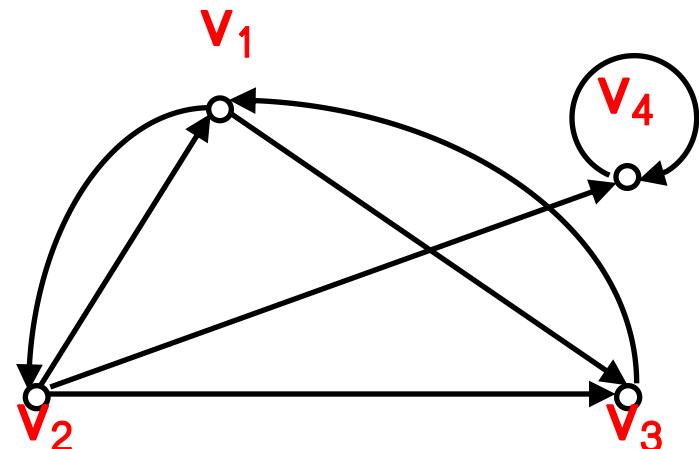
定理5.3.4

设 $G = \langle V, E \rangle$ 为线图, $V = \{v_1, v_2, \dots, v_n\}$, $A = (a_{ij})_{n \times n}$ 为 G 的邻接矩阵, $A^m = (a_{ij}^{(m)})_{n \times n}$, $m=1, 2, \dots, n$; $B^n = (b_{ij}^{(n)})_{n \times n} = A + A^2 + A^3 + \dots + A^n$ 。则有: 如果 $a_{ij}^{(n)} > 0$, 那么从 v_i 到 v_j 可达, 否则不可达; 并且

$$d(v_i, v_j) = \begin{cases} \infty, & \text{如果所有 } a_{ij}^{(1)}, a_{ij}^{(2)}, \dots, a_{ij}^{(n)} \text{ 均为0} \\ k, & \text{否则, } k = \min\{m \mid a_{ij}^{(m)} \neq 0, m=1, 2, \dots, n\} \end{cases}$$

例5.3.3

判断右图中图G中结点之间的可达关系，并求任两结点间的距离。



分析 利用定理5.3.4，先写出图的邻接矩阵A，然后计算A的幂即可。

解

在图中，G的邻接矩阵及其2、3、4次幂分别为：

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

$$A^2 = \begin{pmatrix} 2 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A^3 = \begin{pmatrix} 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 2 \\ 2 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A^4 = \begin{pmatrix} 4 & 1 & 3 & 3 \\ 3 & 2 & 3 & 3 \\ 1 & 2 & 2 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

解(续)

从而有

$$B_4 = A + A^2 + A^3 + A^4 = \begin{pmatrix} 7 & 4 & 7 & 5 \\ 7 & 4 & 7 & 7 \\ 4 & 3 & 4 & 2 \\ 0 & 0 & 0 & 4 \end{pmatrix}$$

故从 v_1 到 v_1, v_2, v_3, v_4 都是可达的；从 v_2 到 v_1, v_2, v_3, v_4 都是可达的；从 v_3 到 v_1, v_2, v_3, v_4 都是可达的；从 v_4 到 v_4 都是可达的，从 v_4 到 v_1, v_2, v_3 都是不可达的。并且有

$$\begin{aligned} d(v_1, v_2) &= d(v_1, v_3) = d(v_2, v_1) = d(v_2, v_3) = d(v_2, v_4) \\ &= d(v_3, v_1) = 1, d(v_1, v_4) = d(v_3, v_2) = 2, d(v_3, v_4) = 3, \\ d(v_4, v_1) &= d(v_4, v_2) = d(v_4, v_3) = \infty. \end{aligned}$$

定义5.3.3

设 $G = \langle V, E \rangle$ 是一个线图，其中 $V = \{v_1, v_2, \dots, v_n\}$ ，并假定结点已经有了从 v_1 到 v_n 的次序，称 n 阶方阵 $P = (p_{ij})_{n \times n}$ 为图 G 的 可达性矩阵 (Accessibility Matrix)，其中

$$p_{ij} = \begin{cases} 1, & \text{当 } v_i \text{ 到 } v_j \text{ 至少存在一条非零长度 的通路} \\ 0, & \text{否则} \end{cases}$$
$$i, j = 1, 2, 3, \dots, n$$

说明

逆向图的可达矩阵是称的，而有向图的可达性矩阵则不一定对称。

与邻接矩阵不同，可达性矩阵不能给出图的完整信息，但由于它简便，在应用上还是很重要的。

如果我们知道矩阵 B^n ，则只需将其中的非零元素写成1，就可得到可达性矩阵，即

$$p_{ij} = \begin{cases} 1, & b_{ij}^{(n)} \neq 0 \\ 0, & b_{ij}^{(n)} = 0 \end{cases} \quad i, j = 1, 2, 3, \dots, n$$

定理5.3.5

设 $G = \langle V, E \rangle$ 为线图， A 、 P 分别是 G 的邻接矩阵和可达性矩阵，则有

$$P = A \vee A^{(2)} \vee A^{(3)} \vee \cdots \vee A^{(n)} = \bigvee_{i=1}^n A^{(i)}$$

这里， $A^{(i)}$ 表示做矩阵布尔乘法的 i 次幂。

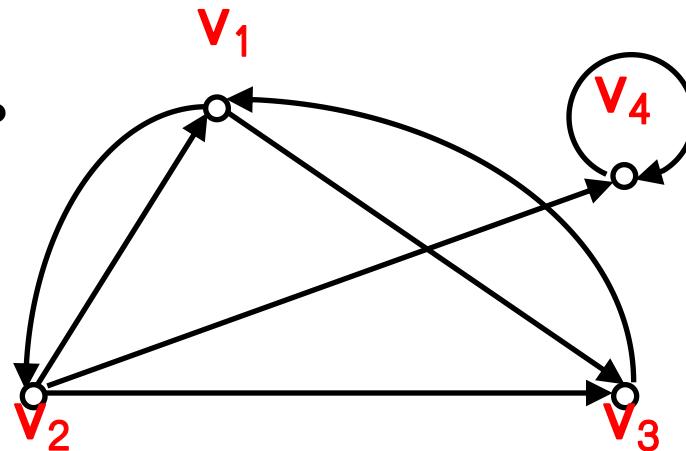
例5.3.4

求右图中图G中的可达性矩阵。

解 在图5.3.4中， G 的邻接矩阵的邻接矩阵 A 布尔乘法^{2、3、4次幂}的^{2、3、4次幂}，然后做布尔加即可。

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

$$A^{(2)} = A \odot A = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



例5.3.4(续1)

$$A^{(3)} = A^{(2)} \odot A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A^{(4)} = A^{(3)} \odot A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

例5.3.4(续2)

于是该图的可达性矩阵为：

$$P = A^{(1)} \vee A^{(2)} \vee A^{(3)} \vee A^{(4)} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

这与我们利用B4求得的结果完全一致。

5.3.2 无向图的连通性

定义5.3.4 若无向图G中的任何两个结点都是可达的，则称G是连通图(Connected Graph)，否则称G是非连通图(Unconnected Graph)或分离图(Separated Graph)。

无向完全图 K_n ($n \geq 1$) 都是连通图，而多于一个结点的零图都是非连通图。

利用邻接矩阵A和可达性矩阵P，显然有：

非平凡无向线图G是连通图当且仅当它的可达性矩阵P的所有元素均为1。

定理5.3.6

无向图 $G = \langle V, E \rangle$ 中结点之间的可达关系 R 定义如下：

$$R = \{ \langle u, v \rangle | u, v \in V, u \text{ 到 } v \text{ 可达} \},$$

则 R 是 V 上的等价关系。

分析 利用等价关系的定义，很容易证明 R 是自反、对称、传递的。

证明

(1) 对任意 $v \in V$, 由于规定任何结点到自身总是可达的, 因此 $\langle v, v \rangle \in R$, 故 R 是自反的;

(2) 对任意 $u, v \in V$, 若 $\langle u, v \rangle \in R$, 则 u 到 v 可达, 即存在从 u 到 v 的通路, 由于 G 是无向图, 因此该通路也是从 v 到 u 的通路, 从而 v 到 u 可达, 即 $\langle v, u \rangle \in R$, 故 R 是对称的;

(3) 对任意 $u, v, w \in V$, 若 $\langle u, v \rangle \in R$, $\langle v, w \rangle \in R$, 则 u 到 v 可达, v 到 w 可达, 即存在从 u 到 v 的通路和从 v 到 w 的通路, 于是存在从 u 经过 v 到 w 的通路, 即 u 到 w 是可达的, 即 $\langle u, w \rangle \in R$, 故 R 是传递的。

说明

利用等价关系的特点，即等价关系可以导致集合的划分，因此对于任何无向图的结点集都存在一种划分，使得每个划分块中的结点都彼此可达，而两个不同划分块中的结点都不可达。

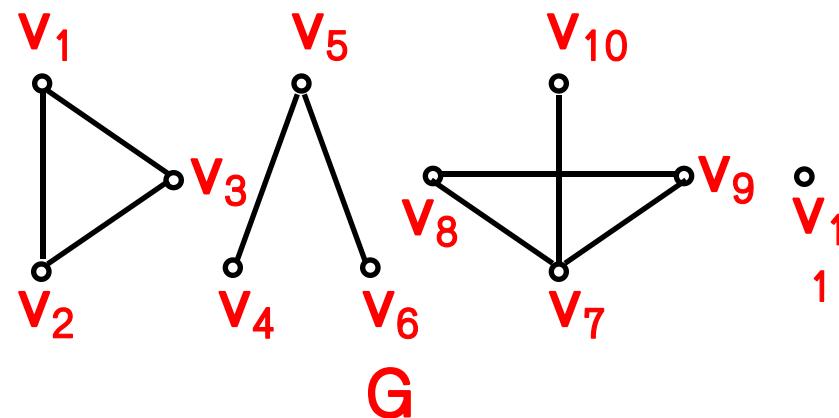
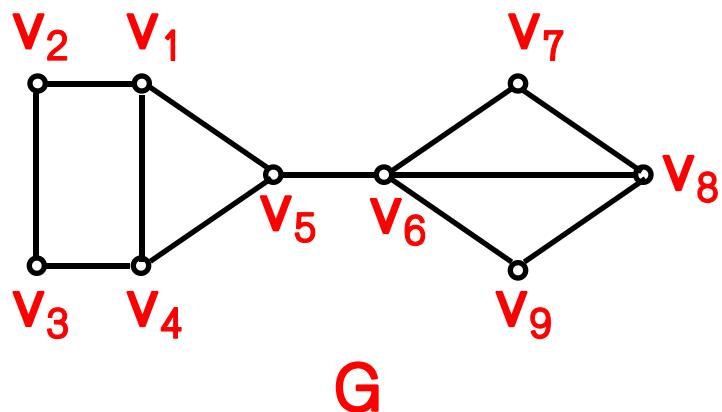
定义5.3.5

无向图 $G = \langle V, E \rangle$ 中结点之间的可达关系 R 的每个等价类导出的子图都称为 G 的一个连通分支(Connected Component)。用 $p(G)$ 表示 G 中的连通分支个数。

显然，无向图 G 是连通图当且仅当 $p(G) = 1$ ；每个结点和每条边都在且仅在一个连通分支中。

例5.3.5

判断下图中图 G_1 和 G_2 的连通性，并求其连通分支个数。



分析解 本题中图很简单，并且给出了图形，很容易看出 G_1 是连通图，所以 $p(G_1) = 1$ 。
在上图中 G_2 是非连通图。容易看出， G_2 中可达关系的等桥类为 $\{v_1, v_2, v_3\}$, $\{v_4, v_5, v_6\}$, $\{v_7, v_8, v_9, v_{10}\}$, $\{v_{11}\}$ ，它们导出的子图即为 G_2 的4个连通分支。

5.3.3 有向图的连通性

定义5.3.5

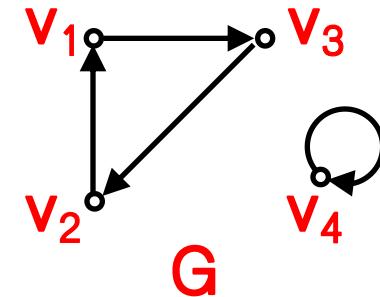
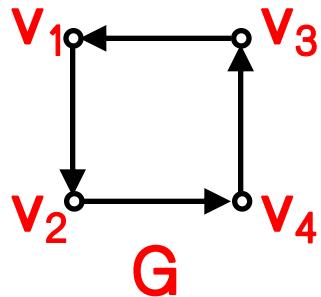
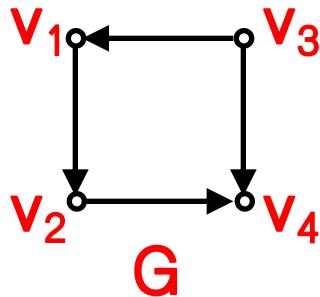
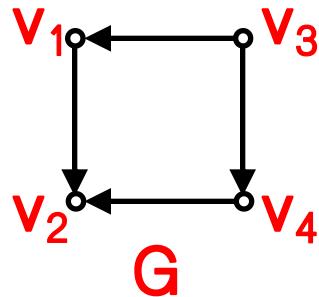
若有向图G是强连通图，则它必是单向连通图；若有向图G是单向连通图，则它必是（弱）连通图。但是上述二命题的逆均不成立。

1. 略

2. 表G中任何一对结点之间至少有一个结点到另一个结点是可达的，则称G是单向连通图 (Unilaterally Connected Graph);
3. 若G中任何一对结点之间都是相互可达的，则称G是强连通图(Strongly Connected Graph)。

例5.3.6

判断下图中4个图的连通性。



分析 先看略去图中所有有向边的方向得无向图，容易看出G₁、G₂、G₃是连通的有向图，G₁是弱连通的有向图。再看有向连通图中必然的可达(弱连通图)到v₄不强连通图不自然所以G₁是弱连通图。G₂中任何一对结点之间至少有一个结点到另一个结点是可达的，所以G₂是单向连通图(当然它也是弱连通图)；G₃中任何一对结点之间至少有一个结点到另一个结点是可达的，所以G₃是强连通图(当然它也是单向连通图和弱连通图)。

定理5.3.7

有向图G是强连通图的充分必要条件是G中存在一条经过所有结点的回路。

分析 只需要利用回路中的任二点相互可达和强连通的定义即可证明。

必要性：设G是强连通图，那么G中任二结点均是相互可达的。不妨设G中的结点为 v_1, v_2, \dots, v_n ，因为 v_i 到 v_{i+1} 是可达的， $i=1, 2, \dots, n-1$ ，且 v_n 到 v_1 是可达的，所以 v_i 到 v_{i+1} 存在通路， $i=1, 2, \dots, n-1$ ，且 v_n 到 v_1 存在通路。让这些连通首尾相接，则得一回路C。显然所有结点均在该回路中出现。

定理5.3.7

有向图G是强连通图的充分必要条件是G中存在一条经过所有结点的回路。

分析 只需要利用回路中的任二点相互可达和强连通的定义即可证明。

证明 充分性：如果G中存在一条经过所有结点的回路C，则G中任意二结点均在回路C上，所以G中任二结点都是相互可达的，因而G是强连通图。

利用A和P判断有向图的连通性

1. 有向线图G是强连通图当且仅当它的可达性矩阵P的所有元素均为1;
2. 有向线图G是单向连通图当且仅当它的可达性矩阵P及其转置矩阵 P^T 经过布尔并运算后所得的矩阵 $P' = P \vee P^T$ 中除主对角元外其余元素均为1;
3. 有向线图G是弱连通图当且仅当它的邻接矩阵A及其转置矩阵 A^T 经布尔并运算所得的矩阵 $A' = A \vee A^T$ 作为邻接矩阵而求得的可达性矩阵 P' 中所有元素均为1。

定义5.3.6

在有向图 $G = \langle V, E \rangle$ 中，设 G' 是 G 的子图，如果

- (1) G' 是强连通的 (单向连通的、弱连通的)；
- (2) 对任意 $G'' \subseteq G$, 若 $G' \subseteq G''$, 则 G'' 不是强连通的 (单向连通的、弱连通的)；

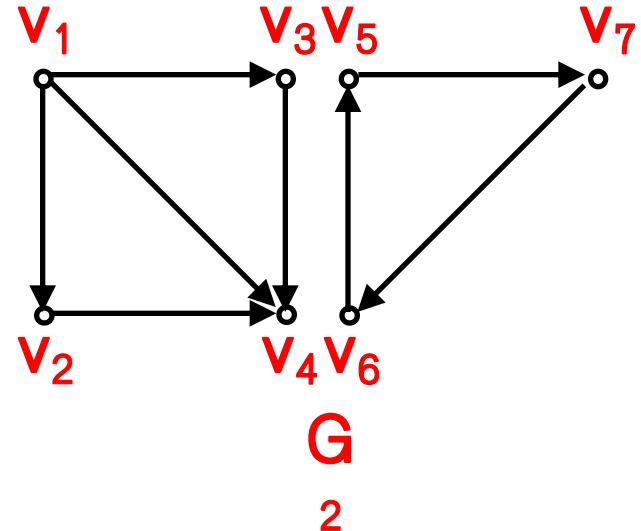
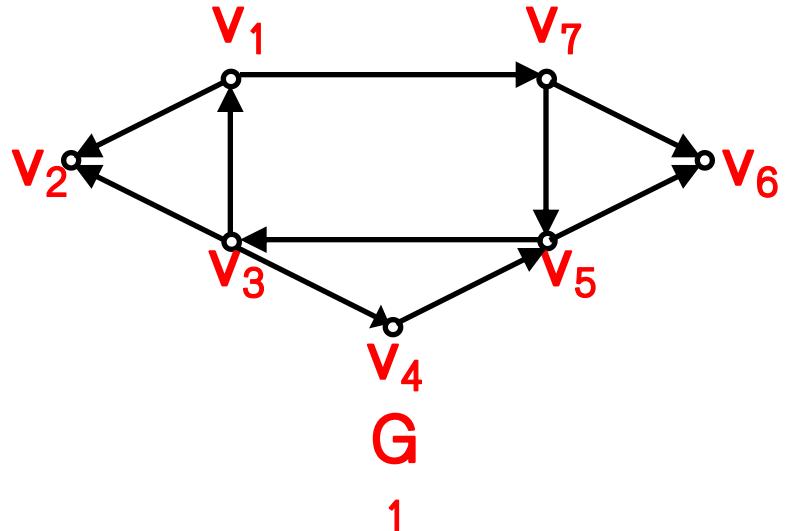
那么称 G' 为 G 的强连通分支 (单向连通分支、弱连通分支) (Strongly/Unilaterally/weakly Connected Component), 或称为强分图 (单向分图、弱分图)。

注

1. 如果不考虑边的方向，弱连通分支对应相应的无向图的连通分支。
2. 注意把握（强、单向、弱）连通分支的**极大性特点**，即任意增加一个结点或一条边就不是（强、单向、弱）连通的了。

例5.3.7

求下面2个图的强、单向和弱连通分支。



分析 由定义从某个结点开始逐渐增加结点，看它们导出的子图是否是强（单向或弱）分图。

解

在图 G_1 中，由 $\{v_2\}$, $\{v_6\}$ 和 $\{v_1, v_3, v_4, v_5, v_7\}$ 导出的子图都是强连通分支；由 $\{v_1, v_2, v_3, v_4, v_5, v_7\}$ 和 $\{v_1, v_3, v_4, v_5, v_6, v_7\}$ 导出的子图都是单向连通分支； G_1 本身为弱连通分支。

在图 G_2 中，由 $\{v_1\}$, $\{v_2\}$, $\{v_3\}$, $\{v_4\}$ 和 $\{v_5, v_6, v_7\}$ 导出的子图都是强连通分支；由 $\{v_1, v_2, v_3\}$, $\{v_1, v_3, v_4\}$ 和 $\{v_5, v_6, v_7\}$ 导出的子图都是单向连通分支；由 $\{v_1, v_2, v_3, v_4\}$ 和 $\{v_5, v_6, v_7\}$ 导出的子图都是弱连通分支。

一个关系

若在有向图 $G = \langle V, E \rangle$ 的结点集 V 上定义二元关系 R 为： $\langle v_i, v_j \rangle \in R$ 当且仅当 v_i 和 v_j 在同一强（弱）连通分支中， $\forall v_i, v_j \in V$ 。显然， R 是一个等价关系。

因为每一个结点 v_i 和自身总在同一强（弱）连通分支中，且 v_i 和自身在同一个强（弱）连通分支中。又若结点 v_i 和 v_j 在同一强（弱）连通分支中，结点 v_j 和 v_k 在同一强（弱）连通分支中，则 v_i 和 v_k 相互可达，故 v_i 和 v_k 在同一强（弱）连通分支中，所以 R 是传递的。

三个定理

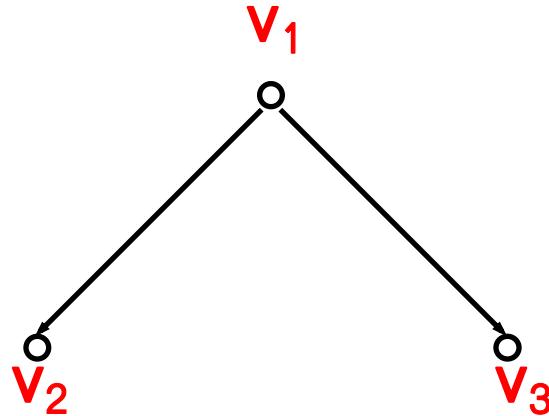
定理5.3.8 在有向图 $G = \langle V, E \rangle$ 中，它的每一个结点位于且仅位于一个强（弱）连通分支中。

定理5.3.9 在有向图 $G = \langle V, E \rangle$ 中，它的每一个结点至少位于一个单向连通分支中。

定理5.3.10 在有向图 $G = \langle V, E \rangle$ 中，它的每一条边至多在一个强连通分支中；至少在一个单向连通分支中；在且仅在一个弱连通分支中。

例

对于“两结点在同一单向连通分支中”这一关系，虽然它是自反的，对称的，但它不是传递的。



上图中， v_2 和 v_1 在同一单向连通分支中， v_1 和 v_3 在同一单向连通分支中，但 v_2 和 v_3 不在同一单向连通分支中。

5.3.4 通路、回路与连通性的难点

1. 要注意区分简单通路与基本通路，简单回路与基本回路；
2. 图的计算机表示就是它的邻接矩阵，我们可以利用邻接矩阵来计算任意两结点间固定长度的通路数目和距离；利用可达性矩阵来判断图的连通性；
3. 计算图的连通分支，特别是有向图的单向分图，是比较难的问题，只有凭经验按定义去试。

5.3.5 通路、回路与连通性的应用

1. 渡河问题

例5.3.7 一个摆渡人要把一只狼、一只羊和一捆菜运过河去。由于船很小，每次摆渡人至多只能带一样东西。另外，如果人不在旁时，狼就要吃羊，羊就要吃菜。问这人怎样才能将它们运过河去？

解 用F表示摆渡人，W表示狼，S表示羊，C表示菜。

若用FWSC表示人和其它三样东西在河的原岸的状态，这样原岸全部可能出现的状态为以下16种。

解(续1)

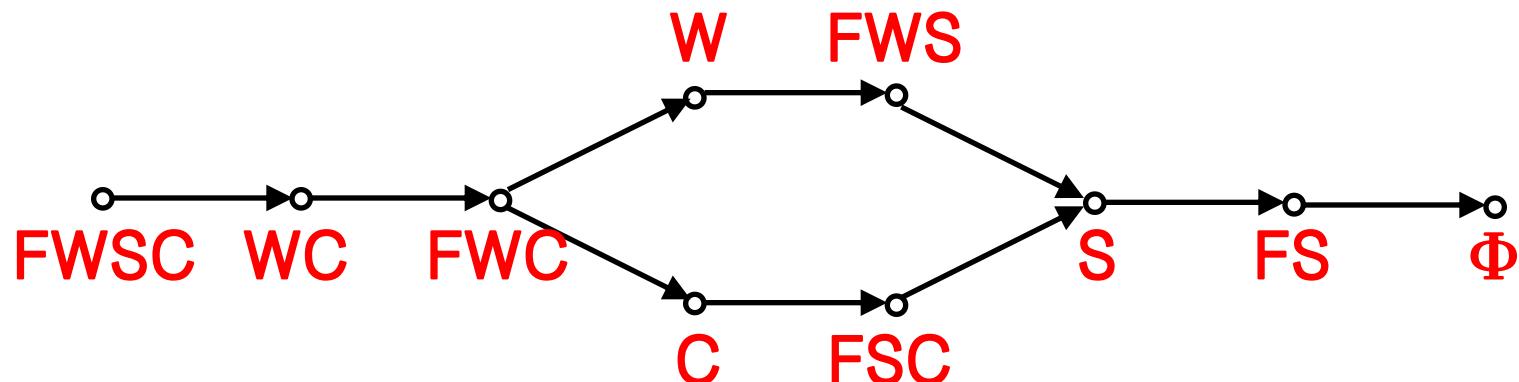
FWSC FWS FWC FSC WSC FW FS
FC
WS WC SC F W S C Φ

这里 Φ 表示原岸什么也没有，即人、狼、羊、菜都已运到对岸去了。

根据题意我们知道，这16种情况中有6种是不允许的，它们是：WSC、FW、FC、WS、SC、F。如FC表示人和菜在原岸，而狼和羊在对岸，这当然是不行的。因此，允许出现的情况只有10种。

解(续2)

以这10种状态为结点，以摆渡前原岸的一种状态与摆渡一次后仍在原岸的状态所对应的结点之间的联线为边做有向图G，如图

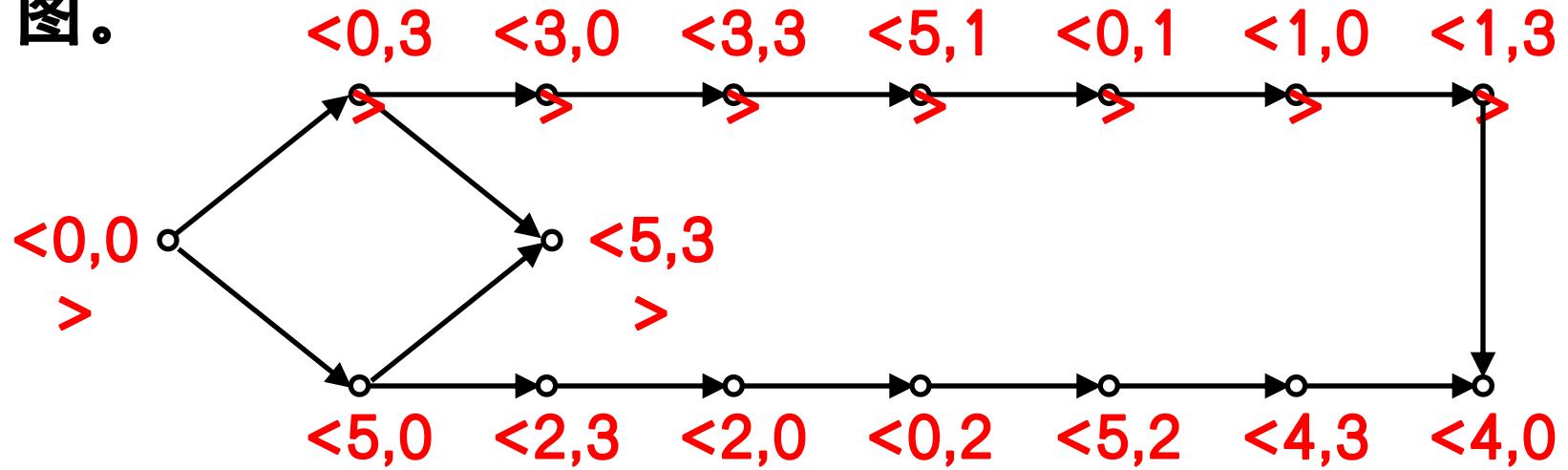


图中给出了两种方案，方案为从FWSC到 Φ 的不同基本通路，它们的长度均为7，按图中所指的方案，摆渡人只要摆渡7次就能将它们全部运到对岸，并且羊和菜完好无损。

2. 均分问题

例5.3.8 有3个没有刻度的桶a、b和c，其容积分别为8升、5升和3升。假定桶a装满了酒，现要把酒均分成两份。除3个桶之外，没有任何其它测量工具，试问怎样均分？

解 用 $\langle B, C \rangle$ 表示桶b和桶c装酒的情况，可得下图。



解

由此可得两种均分酒的方法：

1. a倒满c→c倒入b→a倒满c→c倒满b→b倒入a→ c倒入b→a倒满c→c倒入b；
2. a倒满b→b倒满c→c倒入a→b倒入c→a倒满b→ b倒满c→c倒入a。

3、无向赋权图的最短通路

在赋权图中，边的权也称为边的**长度**，一条通路的**长度**指的就是这条通路上各边的**长度之和**。从结点 v_i 到 v_j 的**长度最小的通路**，称为 v_i 到 v_j 的**最短通路**。

(1) 求给定两结点间的最短通路—Dijkstra算法

如何求出简单无向赋权图 $G = \langle V, E \rangle$ 中从结点 v_1 到 v_n 的最短通路，目前比较好的算法是由Dijkstra在1959年提出的，称为**Dijkstra算法**，其基本思想是：

将结点集合 V 分为两部分：一部分称为**具有P（永久性）标号的集合**，另一部分称为**具有T（暂时性）标号的集合**。所谓**结点v的P标号**是指从 v_1 到 v 的最短通路的长度；而**结点u的T标号**是指从 v_1 到 u 的某条通路的长度。首先将 v_1 取为P标号，其余结点为T标号，然后逐步将具有T标号的结点改为P标号。当结点 v_n 也被改为P标号时，则找到了从 v_1 到 v_n 的一条最短通路。

算法5.3.1 Dijkstra算法

- I. **初始化：** 将 v_1 置为P标号， $d(v_1) = 0$ ， $P = \{v_1\}$ ， $v_i \in V$, $i \neq 1$ ，置 v_i 为T标号，即 $T = V - P$ 且

$$d(v_i) = \begin{cases} w(v_1, v_i), & \text{若 } (v_1, v_i) \in E \\ \infty & \text{若 } (v_1, v_i) \notin E \end{cases}$$

- II. **找最小：** 寻找具有最小值的T标号的结点。若为 v_k ，则将 v_k 的T标号改为P标号，且 $P = P \cup \{v_k\}$ ， $T = T - \{v_k\}$ 。

算法5.3.1 Dijkstra算法(续)

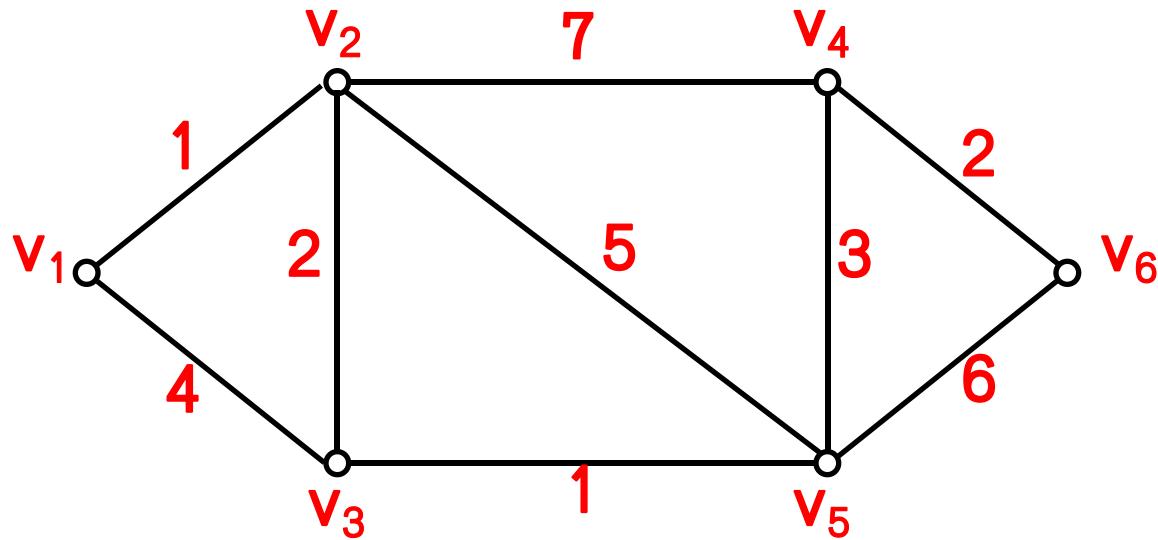
III. 修改：修改与 v_k 相邻的结点的T标号值。 $v_i \in V$,

$$d(v_i) = \begin{cases} d(v_k) + w(v_k, v_i), & \text{若 } d(v_k) + w(v_k, v_i) < d(v_i) \\ d(v_i) & \text{否则} \end{cases}$$

IV. 重复(II)和(III)，直到 v_n 改为P标号为止。

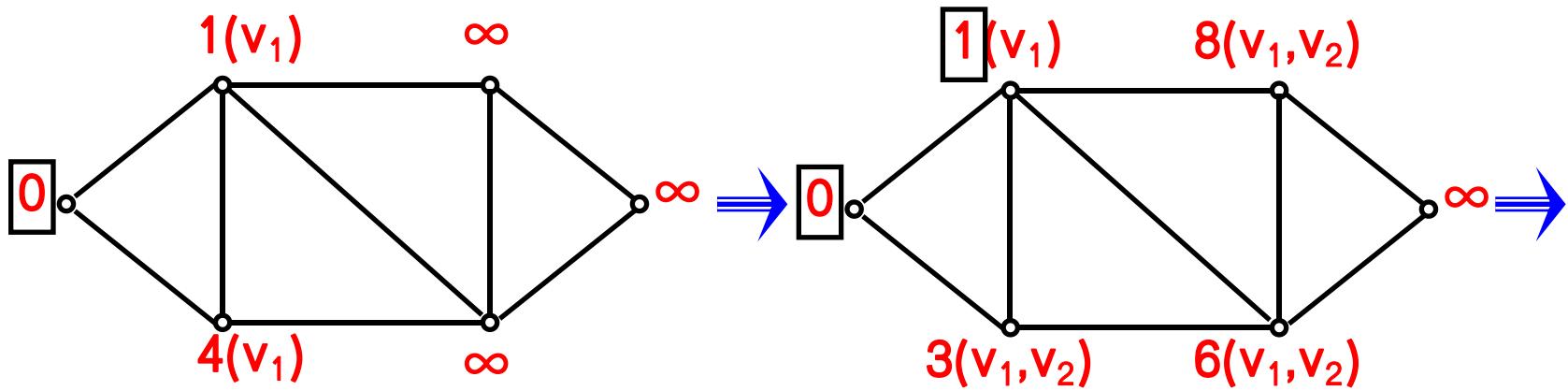
例5.3.9

试求简单无向赋权图中 v_1 到 v_6 的最短通路。

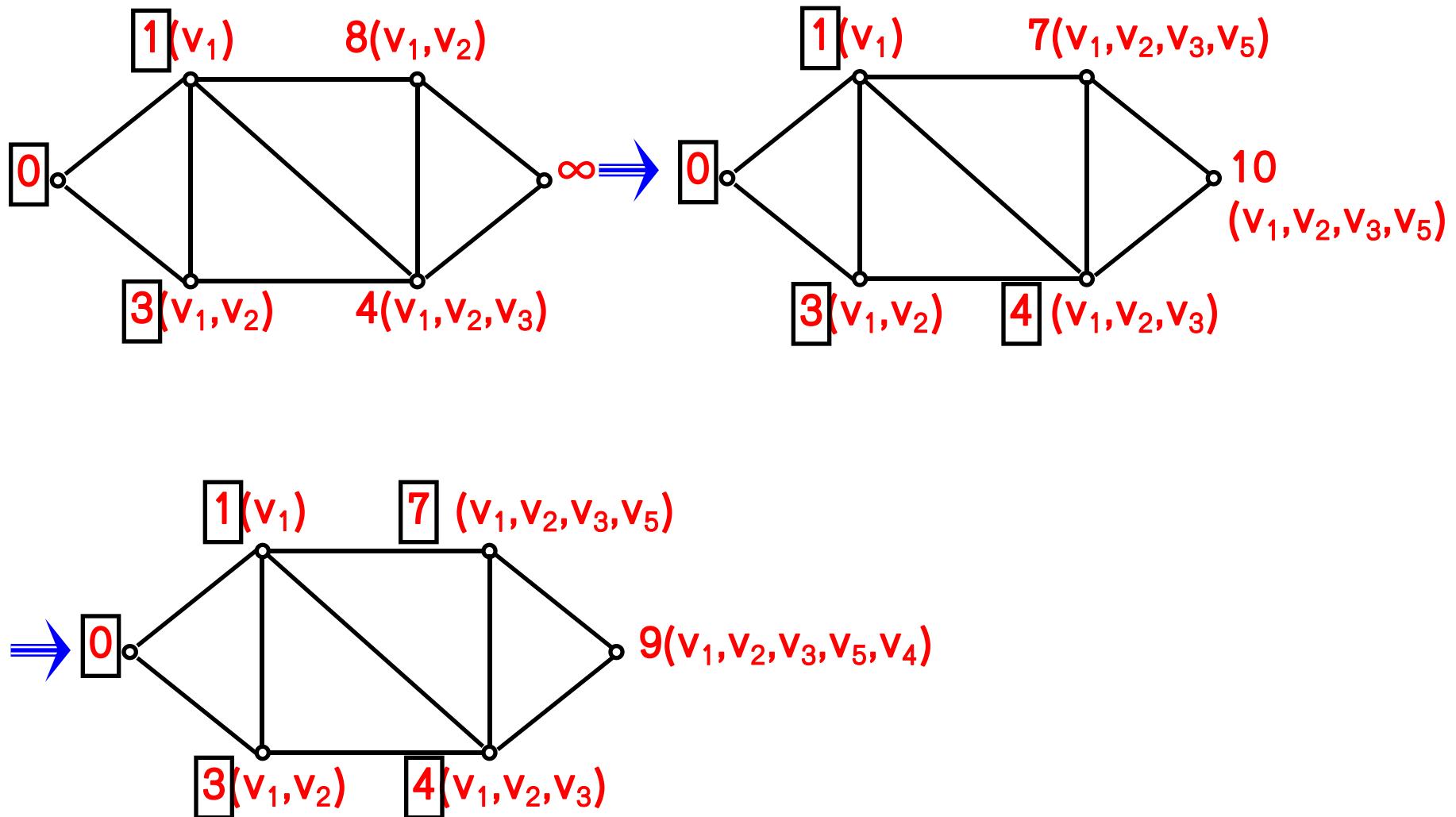


解

根据Dijkstra算法，有如下图所示的求解过程。故 v_1 到 v_6 的最短通路为 $v_1v_2v_3v_5v_4v_6$ ，其长度为9。实际上，也求出了 v_1 到所有结点的最短通路，如 v_1 到 v_5 的最短通路为 $v_1v_2v_3v_5$ ，其长度为4，等等。



解(续)



证明1：从T中选最小值的 $d(u)$ 满足 $d(u) \geq \delta(s, u)$

$\delta(s, u)$ 为s到u的距离

证明概要：

- (1) $d(u)$ 初始值为 ∞ , 且只有找到一个从s到u的通路时其值才会减小;
- (2)s到u的通路长度不会小于 $\delta(s, u)$

下面证明 $d(u) \leq \delta(s, u)$,从而说明 $d(u) = \delta(s, u)$ 。

证明2：从T中选最小值的 $d(u)$ 满足 $d(u) \leq \delta(s, u)$

$\delta(s, u)$ 为s到u的距离

反证法证明：假定以上结论不成立。设节点u为从T集合中移除的第一个节点，使得 $d(u) > \delta(s, u)$ ；

设s到u的最短路径如右图所示（pq代表优先队列），y已在P集合，z还没有放入P集合且z为y邻居，则有以下观察：

(1) $d(y) = \delta(s, y)$ // u是T中第一个节点使得 $d(u) > \delta(s, u)$ ，因此 $d(y) \leq \delta(s, y)$, 且 $d(y) \geq \delta(s, y)$

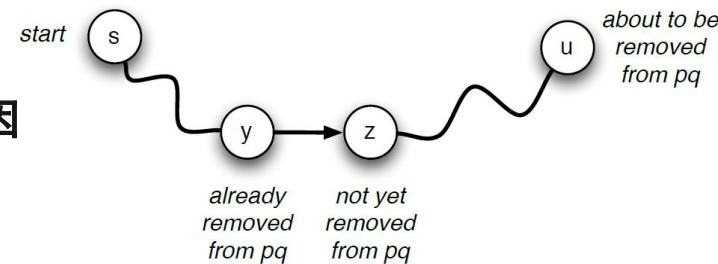
(2) $d(z) \leq d(y) + w(y, z)$ // 因为移除y触发边的松弛操作

(3) $d(u) \leq d(z)$ // 因为u先于z从T中移除

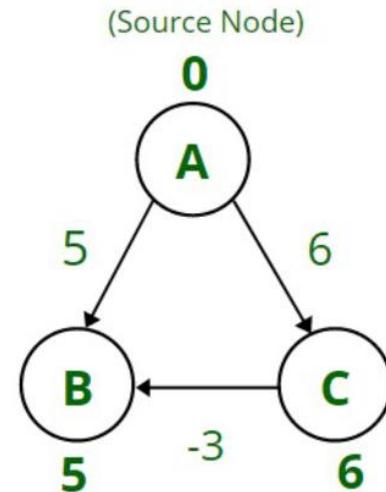
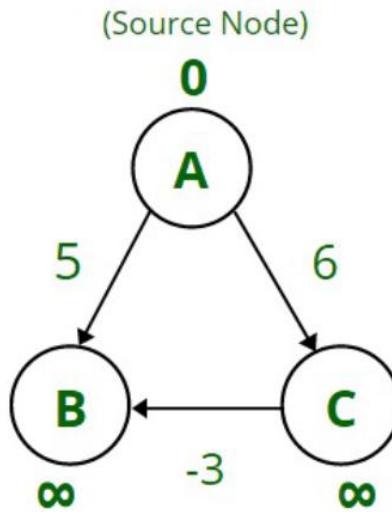
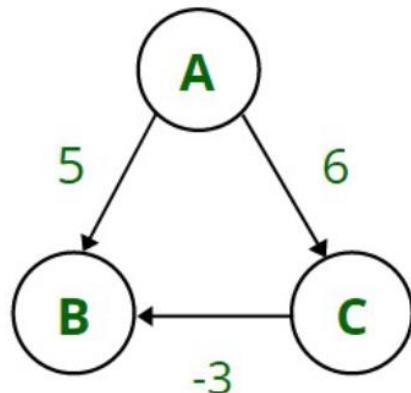
(4) $\delta(s, z) = \delta(s, y) + w(y, z)$ 且 $\delta(s, u) = \delta(s, z) + \delta(z, u)$

由以上观察，且因为图中没有负权值边，可得：

$d(u) \leq d(z) \leq d(y) + w(y, z) = \delta(s, z) \leq \delta(s, z) + \delta(z, u) = \delta(s, u)$, 和假设矛盾。



Dijkstra算法不适用于负权图示例



Dijkstra算法不适用于负权图：

依据上页证明，存在最短路径 $A \rightarrow C \rightarrow B$ ；

$u=B, z=C, s=y=A$, 则 $d(u) \leq d(z) \leq d(y)+6 = \delta(s,z) \leq \delta(s,z)+\delta(z,u)$ 不成立

(2) 求任意两点间最短通路—矩阵相乘算法

考虑从结点*i*到结点*j*的一条最短路径*p*。假定*p*至多包含*m*条边，且*m*为有限值。

- 如果*i=j*, 则*p*中不包含任何边, 所以*p*的权重等于0;
- 如果*i≠j*, 则将路径*p*分解为 $i \xrightarrow{p'} k \rightarrow j$, 其中
 - *p'* 至多包含*m-1*条边, 并为*i*到*k*的一条最短路径,
 - 且 $\delta(i,j) = \delta(i,k) + w_{kj}$ 。

递归求解

设 $l_{ij}^{(m)}$ 是从结点i到结点j的至多包含m条边的任意路径中的最小权重。

➤ 当m=0时，表示从结点i到结点j中间没有边的最短路径，

所以有

$$l_{ij}^{(0)} = \begin{cases} 0 & \text{if } i = j , \\ \infty & \text{if } i \neq j . \end{cases}$$

➤ 当m≥1时，可以在 $l_{ij}^{(m-1)}$ 的基础上计算从i到j的最多由m条边组成的任意路径的最小权重 $l_{ij}^{(m)}$ 。

递归求解

$l_{ij}^{(m)}$ 的递归定义如下:

$$\begin{aligned} l_{ij}^{(m)} &= \min \left(l_{ij}^{(m-1)}, \min_{1 \leq k \leq n} \{ l_{ik}^{(m-1)} + w_{kj} \} \right) \\ &= \min_{1 \leq k \leq n} \{ l_{ik}^{(m-1)} + w_{kj} \}. \end{aligned}$$

- 结点间的最短路径权重 $\delta(i,j)$

如果图G不包含权重为负值的环路，则对于每一对结点i和j，如果 $\delta(i,j) < \infty$ ，则从i到j之间存在一条最短路径。并且，由于最短路径是简单路径，其中至多包含n-1条边，因此有:

$$\delta(i,j) = l_{ij}^{(n-1)}$$

且: $\delta(i,j) = l_{ij}^{(n-1)} = l_{ij}^{(n)} = l_{ij}^{(n+1)} = \dots$

自底向上计算最短路径权重

(1) $I_{ij}^{(1)}$ 表示从结点i到结点j中间至多包含1条边的路径的最小权重,

$$I_{ij}^{(1)} = W_{ij}$$

(2) 自底向上计算方法:

根据输入矩阵 $W=(w_{ij})$, 计算 $L(1)=(I_{ij}^{(1)})$, 然后根据递归关系式计算

$L(2)$ 、 \dots 、 $L(n-1)$ 。其中,

$$L^1 = (W_{ij}) = W$$

$$L^m = I_{ij}^{(m)}, \quad m=1, 2, \dots, n-1.$$

(3) $L(n-1)$ 即是最后的最短路径权重结果矩阵, 有 $I_{ij}^{(n-1)} = \delta(i, j)$

最短路径和矩阵乘法

计算最短路径结果矩阵的过程和矩阵乘法在整体框架上是一致的

一次“ L 与 W 的乘 $L \cdot W$ ”完成一次从 $L^{(m-1)}$ 向 $L^{(m)}$ 的计算。

EXTEND-SHORTEST-PATHS(L, W)

```
1   $n = L.\text{rows}$ 
2  let  $L' = (l'_{ij})$  be a new  $n \times n$  matrix
3  for  $i = 1$  to  $n$ 
4      for  $j = 1$  to  $n$ 
5           $l'_{ij} = \infty$ 
6          for  $k = 1$  to  $n$ 
7               $l'_{ij} = \min(l'_{ij}, l_{ik} + w_{kj})$ 
8  return  $L'$ 
```

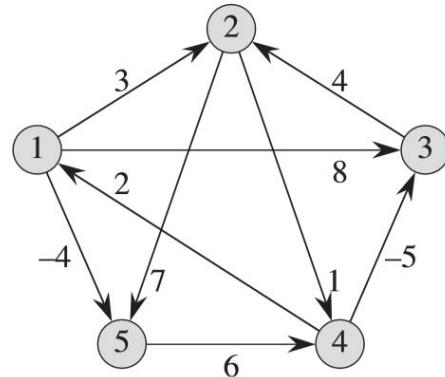
SQUARE-MATRIX-MULTIPLY(A, B)

```
1   $n = A.\text{rows}$ 
2  let  $C$  be a new  $n \times n$  matrix
3  for  $i = 1$  to  $n$ 
4      for  $j = 1$  to  $n$ 
5           $c_{ij} = 0$ 
6          for  $k = 1$  to  $n$ 
7               $c_{ij} = c_{ij} + a_{ik} \cdot b_{kj}$ 
8  return  $C$ 
```

$$\begin{aligned} L^{(1)} &= L^{(0)} \cdot W = W, \\ L^{(2)} &= L^{(1)} \cdot W = W^2, \\ L^{(3)} &= L^{(2)} \cdot W = W^3, \\ &\vdots \\ L^{(n-1)} &= L^{(n-2)} \cdot W = W^{n-1}. \end{aligned}$$

最短路径矩阵乘法示例

矩阵序列: $L^{(1)} \rightarrow L^{(2)} \rightarrow L^{(3)} \rightarrow L^{(4)}$ 。而任何 $m \geq 4$, $L^{(m)} = L^{(4)}$ 。



$$L^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad L^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 2 & -4 \\ 3 & 0 & -4 & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & \infty & 1 & 6 & 0 \end{pmatrix}$$

$$L^{(3)} = \begin{pmatrix} 0 & 3 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix} \quad L^{(4)} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

(3) 任意两点间最短通路—Floyd-Warshall算法

算法5.3.2 Floyd-Warshall算法：

从矩阵 $D^{(0)} = (w_{ij})_{n \times n}$ (这里 $w_{ij} = w(v_i, v_j)$, 称为图的长度矩阵) 开始, 依次构造出 n 个矩阵 $D^{(1)}$ 、 $D^{(2)}$ 、...、 $D^{(n)}$, 这里 n 为图中结点的个数。第 k 个矩阵 $D^{(k)} = (d_j^{(k)})$ 的元素 $d_j^{(k)}$ 表示从结点 v_i 到 v_j 而中间结点仅属于 v_1 到 v_k 的 k 个结点的所有通路中的最短通路长度。

若已知 $D^{(k-1)} = (d_j^{(k-1)})$, 则 $D^{(k)} = (d_j^{(k)})$ 的元素规定为

$$d_j^{(k)} = \min\{d_j^{(k-1)}, d_{kj}^{(k-1)} + d_{kj}^{(k-1)}\}$$

Floyd-Warshall算法

运算过程从 $k = 1$ 开始，让 i 和 j 分别取遍从 1 到 n 的所有值，然后 k 增加 1，如此反复进行，直到 $k = n$ 为止。这时 $D^{(n)} = (d_{ij}^{(k)})$ 的元素就是从 v_i 到 v_j 的最短通路长度。

算法的正确性是显然的。Floyd-Warshall 算法求出了任意两点间最短通路的长度，从而很容易得出相应的最短通路。

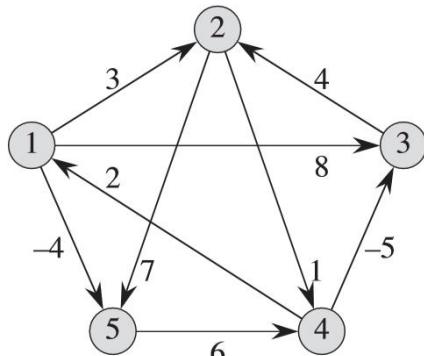
Floyd-Warshall算法伪代码

FLOYD-WARSHALL(W)

```
1   $n = W.rows$ 
2   $D^{(0)} = W$ 
3  for  $k = 1$  to  $n$ 
4      let  $D^{(k)} = (d_{ij}^{(k)})$  be a new  $n \times n$  matrix
5      for  $i = 1$  to  $n$ 
6          for  $j = 1$  to  $n$ 
7               $d_{ij}^{(k)} = \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$ 
8  return  $D^{(n)}$ 
```

- $W_{n \times n}$: 权重邻接矩阵;
- $D_{n \times n}$: 最短路径权重矩阵
- FLOYD-WARSHALL自底向上地完成D矩阵的计算。

例5.3.10 (Floyd-Warshall算法运行过程)



FLOYD-WARSHALL(W)

```

1  n = W.rows
2  D(0) = W
3  for k = 1 to n
4      let D(k) = (dij(k)) be a new n × n matrix
5      for i = 1 to n
6          for j = 1 to n
7              dij(k) = min (dij(k-1), dik(k-1) + dkj(k-1))
8  return D(n)
```

$$D^{(0)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad D^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$D^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad D^{(3)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$D^{(4)} = \begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

$$D^{(5)} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

故点2到点5的最短通路长度为-1，其最短通路为2-4-1-5，其余类似。

4、立方时间求可达矩阵/有向图传递闭包

给定有向图 $G = (V, E)$, 图 G 的传递闭包 $G^* = (V, E^*)$, 其中 $E^* = \{(i, j) : \text{如果图 } G \text{ 中包含一条从结点 } i \text{ 到结点 } j \text{ 的路径}\}$ 。

求有向图的传递闭包:

方法一: 给 E 中每条边赋权重 1, 然后运行 FLOYD-Warshall 算法,
可以在 $\Theta(n^3)$ 求出权重路径矩阵 D 。在 D 中若 $d_{ij} < n$, 则表示
存在一条从结点 i 到结点 j 的路径; 否则 $d_{ij} = \infty$ 。

方法二: 定义矩阵 $T = \{t_{ij}\}$, 若存在一条从结点 i 到结点 j 的路径, $t_{ij} = 1$,
否则 $t_{ij} = 0$ 。

立方时间计算矩阵T

对FLOYD-WARSHALL算法进行改造：用逻辑或操作 (\vee) 和逻辑与操作

(\wedge) 替换算术操作min和+。得以下计算公式：

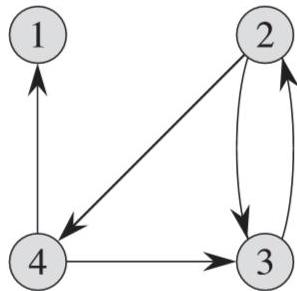
$$k=0 \text{时}, \quad t_{ij}^{(0)} = \begin{cases} 0 & \text{if } i \neq j \text{ and } (i, j) \notin E, \\ 1 & \text{if } i = j \text{ or } (i, j) \in E, \end{cases}$$

$$k \geq 1 \text{ 时}, \quad t_{ij}^{(k)} = t_{ij}^{(k-1)} \vee (t_{ik}^{(k-1)} \wedge t_{kj}^{(k-1)}) .$$

TRANSITIVE-CLOSURE(G)

```
1  n = |G.V|
2  let  $T^{(0)} = (t_{ij}^{(0)})$  be a new  $n \times n$  matrix
3  for  $i = 1$  to  $n$ 
4      for  $j = 1$  to  $n$ 
5          if  $i == j$  or  $(i, j) \in G.E$ 
6               $t_{ij}^{(0)} = 1$ 
7          else  $t_{ij}^{(0)} = 0$ 
8  for  $k = 1$  to  $n$ 
9      let  $T^{(k)} = (t_{ij}^{(k)})$  be a new  $n \times n$  matrix
10     for  $i = 1$  to  $n$ 
11         for  $j = 1$  to  $n$ 
12              $t_{ij}^{(k)} = t_{ij}^{(k-1)} \vee (t_{ik}^{(k-1)} \wedge t_{kj}^{(k-1)})$ 
13 return  $T^{(n)}$ 
```

例5.3.11(求下图G的传递闭包矩阵T)



TRANSITIVE-CLOSURE(G)

```
1   $n = |G.V|$ 
2  let  $T^{(0)} = (t_{ij}^{(0)})$  be a new  $n \times n$  matrix
3  for  $i = 1$  to  $n$ 
4    for  $j = 1$  to  $n$ 
5      if  $i == j$  or  $(i, j) \in G.E$ 
6         $t_{ij}^{(0)} = 1$ 
7      else  $t_{ij}^{(0)} = 0$ 
8  for  $k = 1$  to  $n$ 
9    let  $T^{(k)} = (t_{ij}^{(k)})$  be a new  $n \times n$  matrix
10   for  $i = 1$  to  $n$ 
11     for  $j = 1$  to  $n$ 
12        $t_{ij}^{(k)} = t_{ij}^{(k-1)} \vee (t_{ik}^{(k-1)} \wedge t_{kj}^{(k-1)})$ 
13  return  $T^{(n)}$ 
```

$$T^{(0)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \quad T^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \quad T^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

$$T^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad T^{(4)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

5.4 本章总结



主要知识点汇集



解题分析和方法

1、主要知识点汇集

- ① 图的概念：图的定义、图的表示、图的操作、邻接点与邻接边、图的分类等。
- ② 图的基本性质：结点的度数、图的基本定理（握手定理）、图的同构、完全图、补图、子图、真子图、生成子图、导出子图等。
- ③ 通路与回路：通路与回路、简单（基本）通路与简单（基本）回路、通路与回路长度、结点间的短程线和距离、可达与可达性矩阵。
- ④ 图的连通性：无向连通图与连通分支、强（单向、弱）连通图与强（单向、弱）分图、利用邻接矩阵和可达性矩阵判断图的连通性。
- ⑤ 图的应用：通讯网络、渡河问题、均分问题、最短通路算法。

2. 解题分析和方法

- ① 图的集合、图形、矩阵3种表示方式在同构的意义下是惟一的；
- ② 图的邻接矩阵既描述了图中结点间的邻接关系，还可以利用它来计算结点间的通路数目、判断图的连通性；
- ③ 判断两个图同构，只能依据同构的定义，构造两个结点集之间的双射函数，没有简单的方法；
- ④ 在计算和证明与结点的度数有关的问题时，经常使用握手定理；
- ⑤ 反证法非常有用，特别是在证明唯一性和不存在的时候。