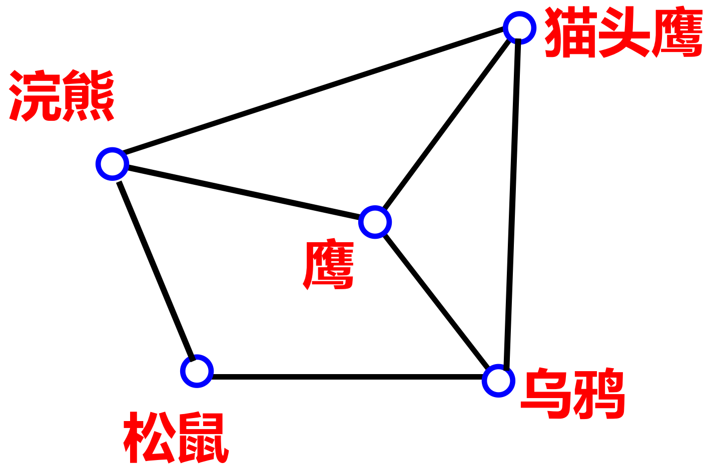
图

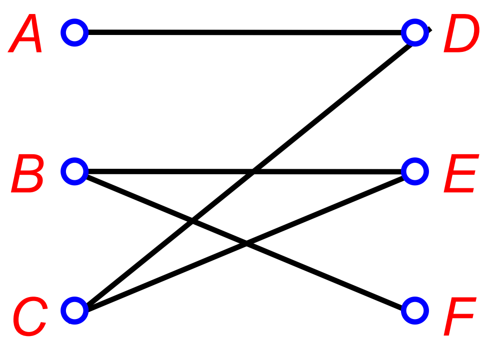
《离散数学》 Discrete Mathematics

现实世界中许多现象都是由某类事物及事物间的联系构成的，能用某种图形表示，这些图形**由一些点**和**两点间的连线**所组成。

（1）考虑一张物种栖息地重叠图，图中用点表示每个物种，当两个物种竞争（即它们共享某些食物来源）时，就用一条线将相应的点连接起来。这种图的一部分如左下图所示，从图中看出，松鼠与浣熊竞争，乌鸦不与浣熊竞争。

（2）在电子商务中，用户与商品之间的购买关系如下：有3个用户*A*、*B*和*C*，3件商品*D*、*E*和*F*，假设*A*购买了*D*，*B*购买了*E*和*F*，*C*购买了*D*和*E*，则这种购买情形可用右下图表示。

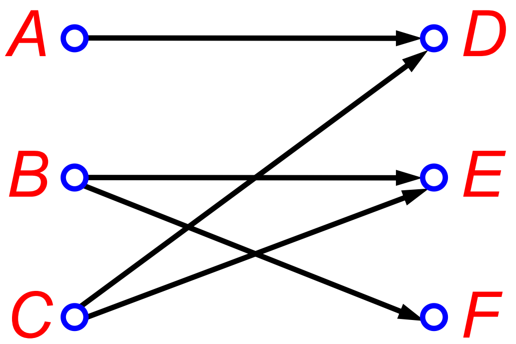




一个**图**(Graph)是一个**序偶**$<V, E>$，记为$G = <V, E>$，其中：

(1) $V = \{v_1, v_2, \ldots, v_n\}$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)。



- 注意：定义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)，也称结点u与边e(结点v与边e)是**彼此相关联的**。
- 若边e与**有序结点对**$<u, v>$相对应，则称e为**有向边**(Directed Point)(或弧)，记为$e = <u, v>$，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**。

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

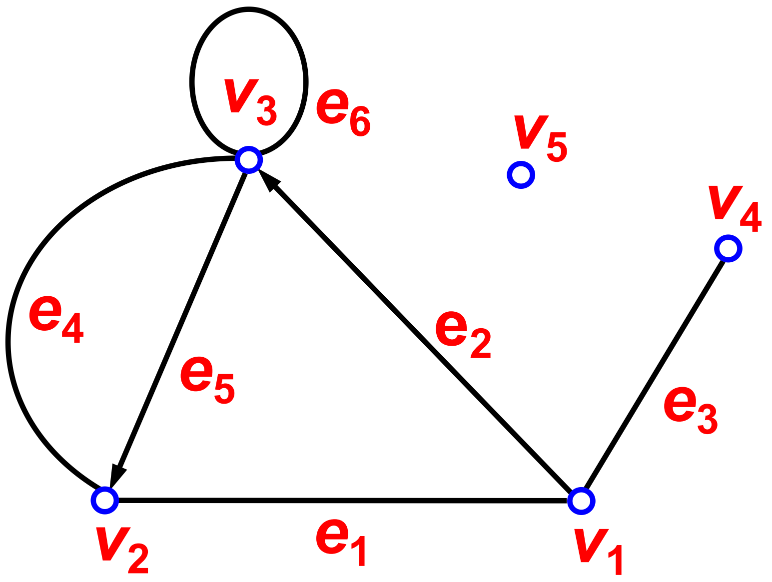
而为了描述简便起见，在一般情况下，往往只画出它的图形：用小圆圈表示*V*中的结点，用由*u*指向*v*的有向线段或曲线表示有向边<*u*, *v*>，无向线段或曲线表示无向边(*u*, *v*)，这称为**图的图形表示**。

解题小贴士——图*G* = <*V*,*E*>的集合表示与图形表示相互转换的方法

（1）**集合表示转换为图形表示**。用小圆圈表示*V*中的每一个结点，结点位置可随意放，元素<*u*,*v*>用由*u*指向*v*的有向边表示，元素(*u*,*v*)用*u*与*v*相连的无向边表示。

（2）**图形表示转换为集合表示**。图中的所有结点构成结点集，图中的无向边用无序偶对表示，有向边用序偶表示，注意箭头指向的结点是序偶的第二元素。

设图$G = <V, E>$，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$e_1 = (v_1, v_2)$，$e_2 = <v_1, v_3>$，$e_3 = (v_1, v_4)$，$e_4 = (v_2, v_3)$，$e_5 = <v_3, v_2>$，$e_6 = (v_3, v_3)$。试画出图$G$的图形，并指出哪些是有向边，哪些是无向边？

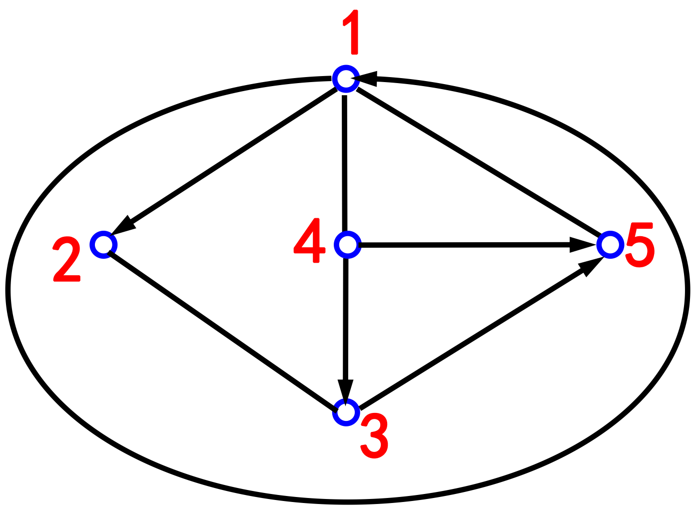


无向边：e_1、e_3、e_4、e_6

有向边：e_2、e_5

▸▸▸ 例6.3

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 图G的集合表示为$G = \langle V, E\rangle = \langle\{1, 2, 3, 4, 5\}$，$\{\langle 1, 1\rangle$，$\langle 1, 2\rangle$，$(1, 4)$，$(1, 5)$，$(2, 3)$，$\langle 3, 5\rangle$，$\langle 4, 3\rangle$，$\langle 4, 5\rangle\}\rangle$。

- 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。
- 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。

设图G = <V, E>，其中V = {v_1, v_2, …, 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

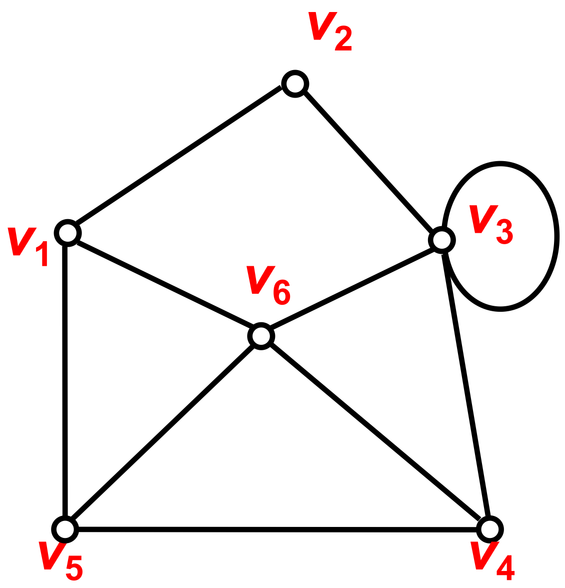
$$a_{ij} = \begin{cases} 1, & \text{若}(v_i, v_j) \in E\text{或} < v_i, v_j > \in E \\ 0, & \text{否则} \end{cases} \quad i, j = 1, 2, 3, \ldots, n$$

解题小贴士——图的邻接矩阵表示

（1）将图中的结点排序。

（2）图中第i个结点到第j个结点有边，则邻接矩阵的第i行第j列元素为1。

试写出右图所示图*G*的邻接矩阵。



解 若结点排序为$v_1v_2v_3v_4v_5v_6$，则其邻接矩阵

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

$$\begin{array}{c c}
 & \begin{array}{cccccc}v_1&v_2&v_3&v_4&v_5&v_6\end{array}\\
\begin{array}{c}v_1\\v_2\\v_3\\v_4\\v_5\\v_6\end{array} & \begin{pmatrix}0&1&0&0&1&1\\1&0&1&0&0&0\\0&1&1&1&0&1\\0&0&1&0&1&1\\1&0&0&1&0&1\\1&0&1&1&1&0\end{pmatrix}
\end{array}$$

二元关系图--矩阵

由定义6.2可看出，图$G = <V, E>$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。

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。

定义6.3 设图$G = <V, E>$。

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$G\backslash e$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**。

（1）删除边e，就是直接从图中去掉边e。

（2）删除结点v，不仅要去掉结点v，还要去掉结点v关联的所有边。

（3）收缩边e，就是将边e的长度缩短到零，并用一个新结点w代替e的两个端点，原来与e的两个端点关联的所有边改为与w关联。

（4）加新边就是增加一条边。

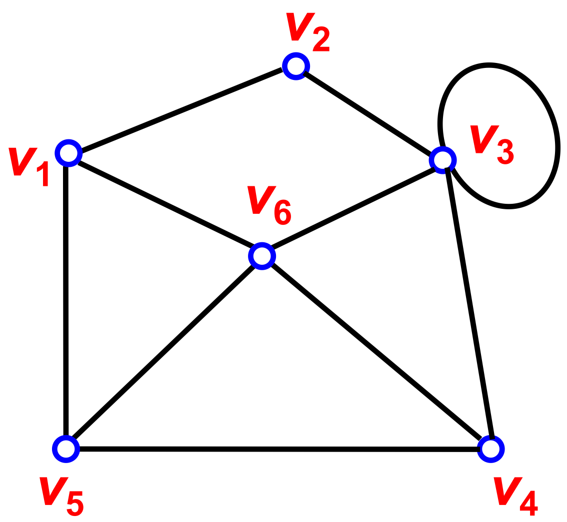
对右图所示的图G完成下列操作。

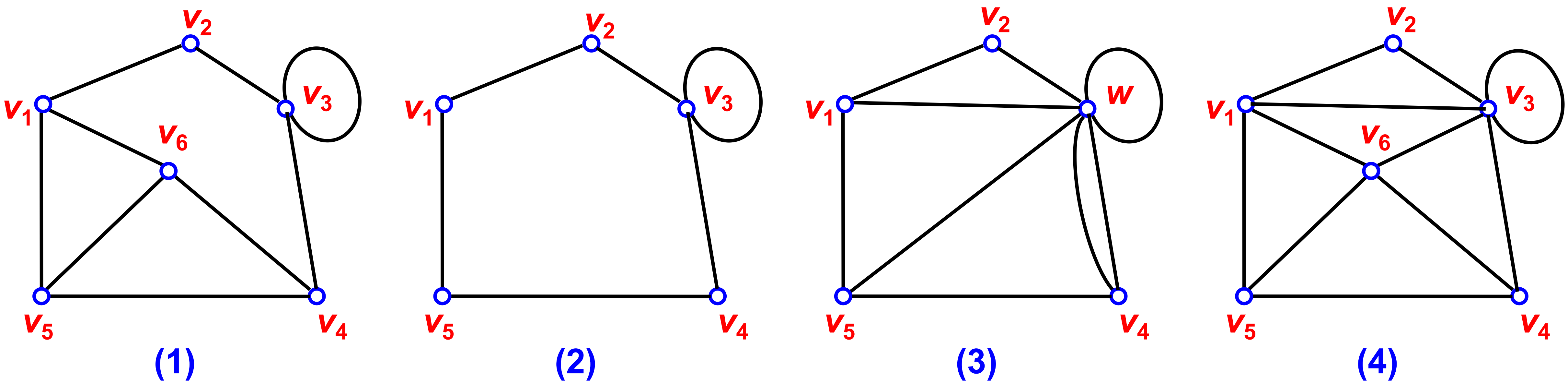
（1）删除边(v_3,v_6)。

（2）删除结点v_6。

（3）收缩边(v_3,v_6)。

（4）加新边(v_1,v_3)。





定义6.4 在图$G = <V, E>$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**

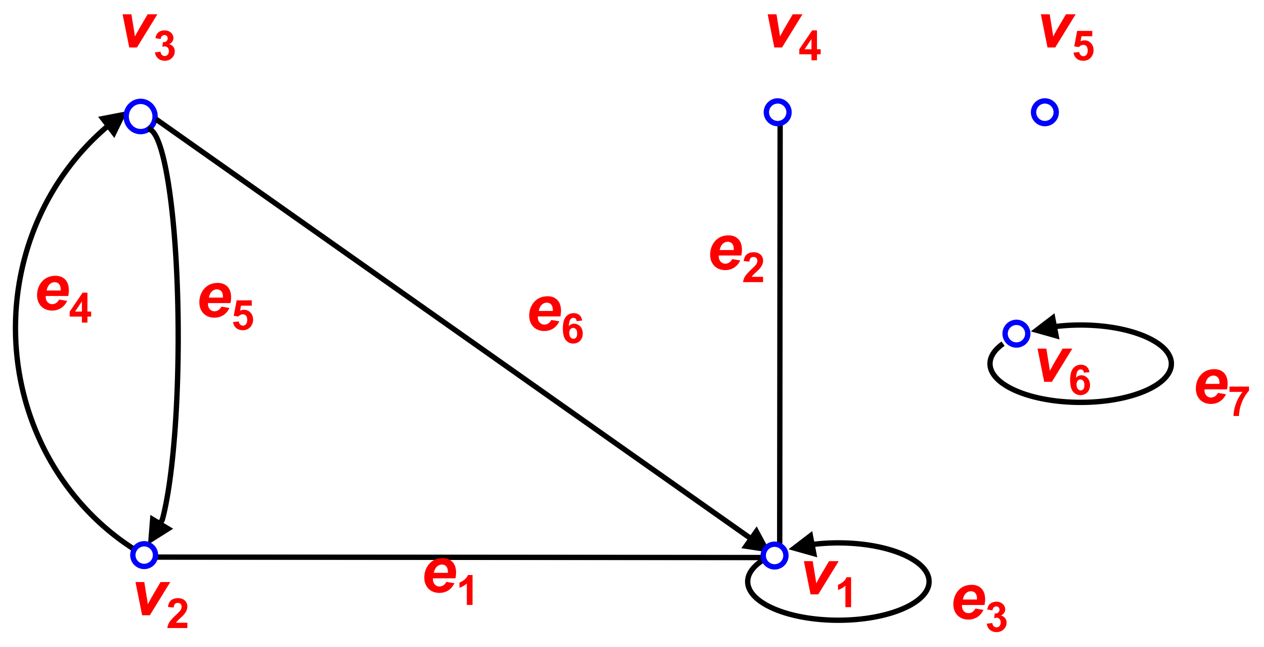
▸▸▸ 解题小贴士——邻接点与邻接边的计算

（1）一个点的邻接点就是所有以这个点为端点的边的另一个端点。

（2）一条边的邻接边就是所有以这条边的两个端点为公共结点的边。

注意：只有当一个结点处有环时，它才是自己的邻接点，而所有边都是自己的邻接边。

写出下面图G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。

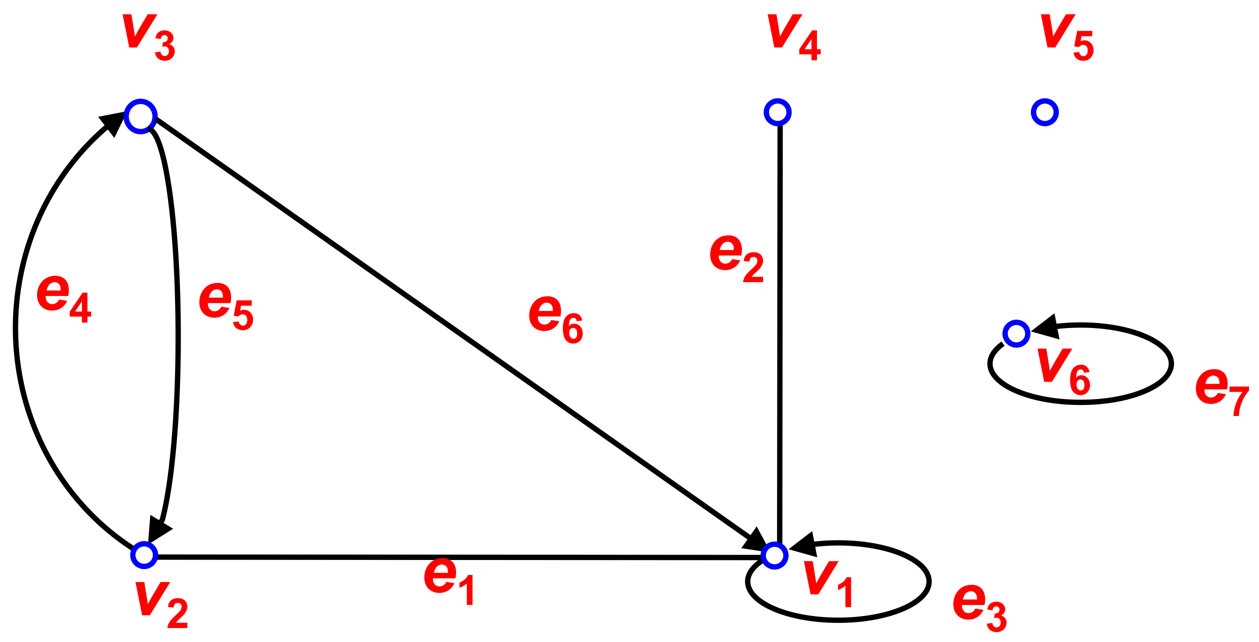


图G既不是平凡图，也不是零图，而是一个(6,7)图。

结点	邻接点	孤立结点
v_1	v_1, v_2, v_3, v_4	否
v_2	v_1, v_3	否
v_3	v_1, v_2	否
v_4	v_1	否
v_5		是
v_6	v_6	否

例9.2.5

写出下面图G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。



图G既不是平凡图，也不是零图，而是一个(6,7)图。

边	邻接边	环
e_1	$e_1, e_2, e_3, e_4, e_5, e_6$	否
e_2	e_1, e_2, e_3, e_6	否
e_3	e_1, e_2, e_3, e_6	是
e_4	e_1, e_4, e_5, e_6	否
e_5	e_1, e_4, e_5, e_6	否
e_6	$e_1, e_2, e_3, e_4, e_5, e_6$	否
e_7	e_7	是

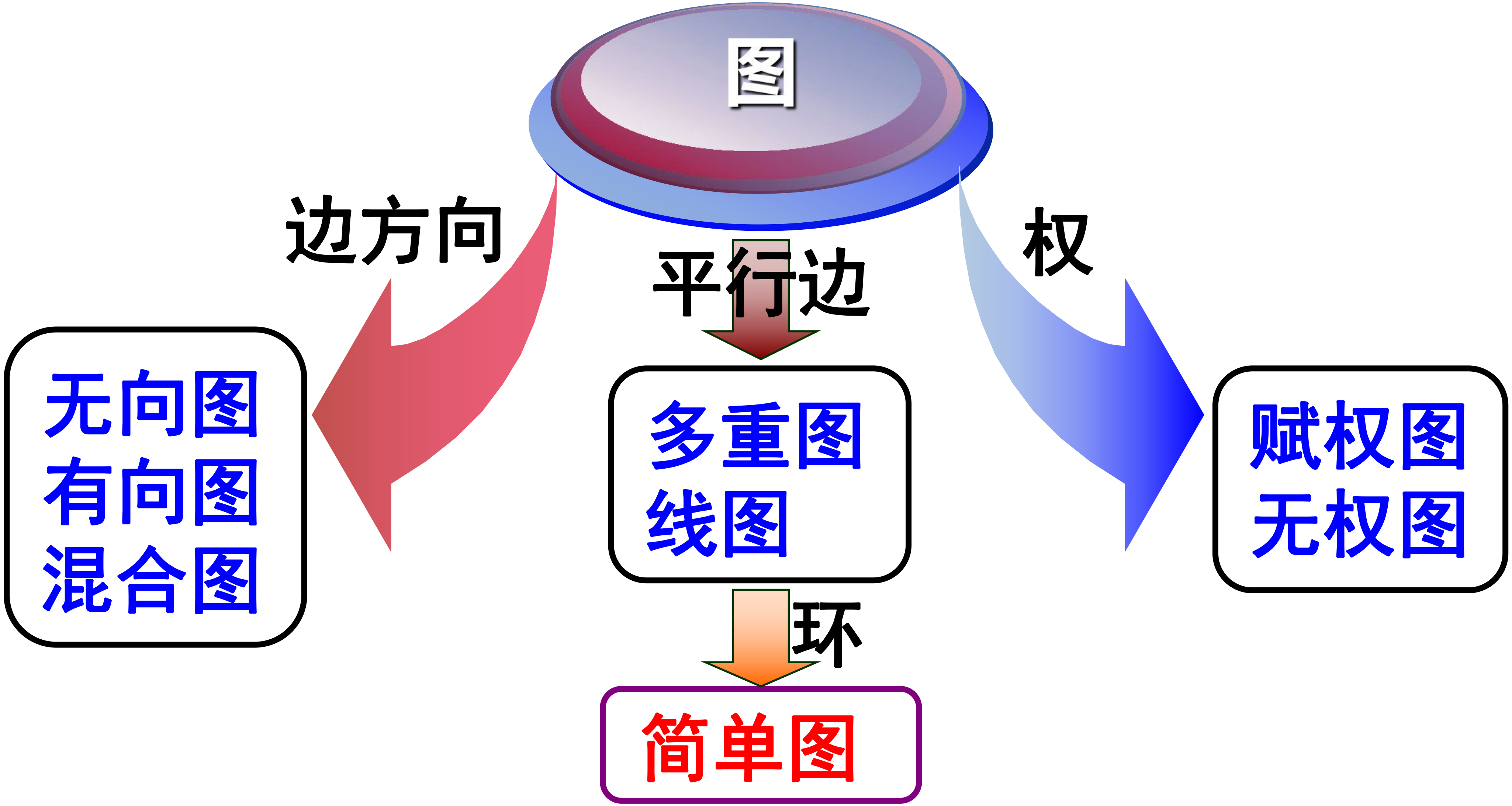


图
边方向
平行边
权
无向图
有向图
混合图
多重图
线图
赋权图
无权图
环
简单图

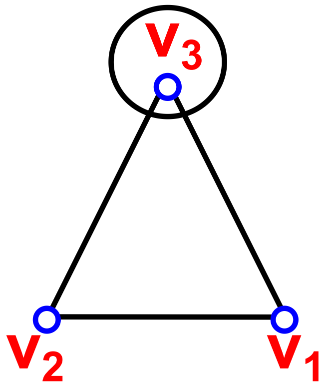
定义6.5 每条边都是**无向边**的图称为**无向图**(Undirected Graph)；

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)。

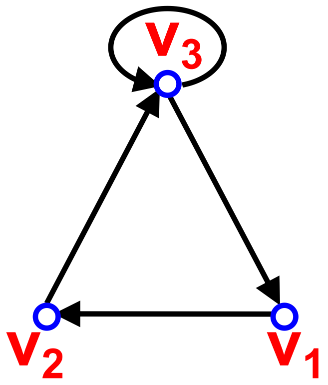
- 二元关系图都是有向图，这时邻接矩阵就是关系矩阵。

例6.7 试判断下面三个图是无向图、有向图，还是混合图 。



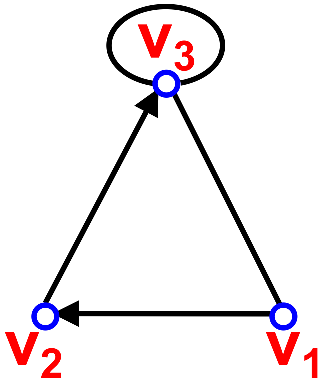
G_1

无向图



G_2

有向图

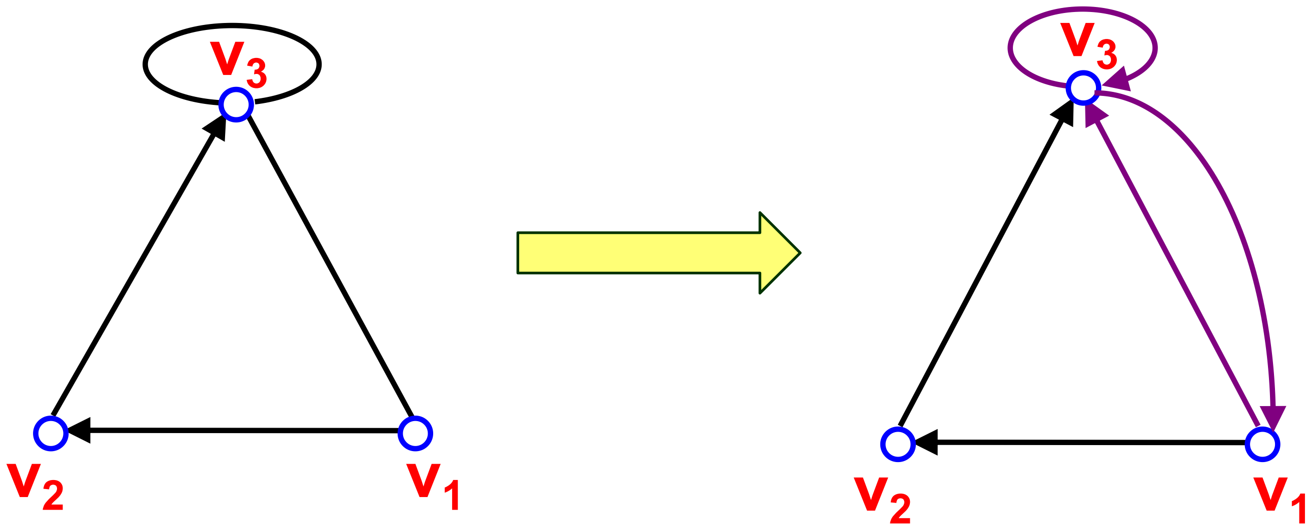


G_3

混合图

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。

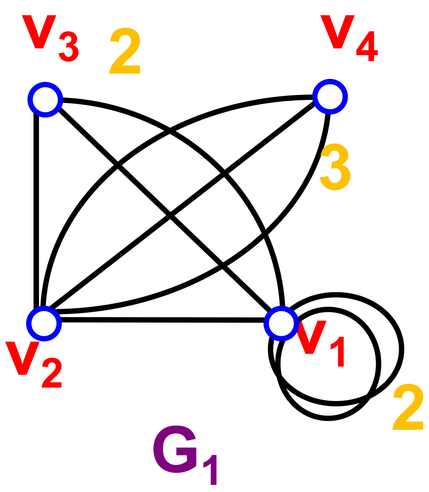
例如可将混合图G_3转化为有向图。



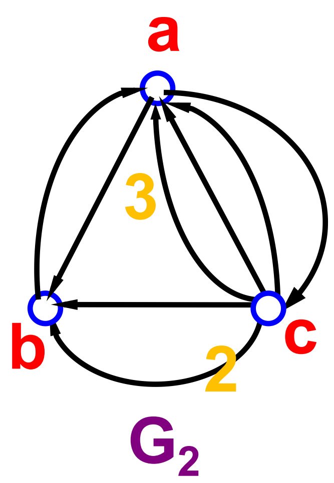
定义6.6

- 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)。

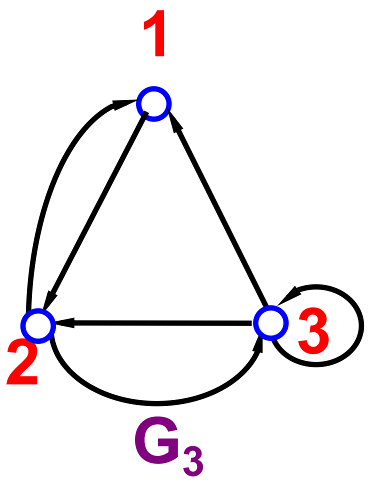
试判断下图所示的4个图是多重图、线图还是简单图，并指出多重图中所有平行边的重数。



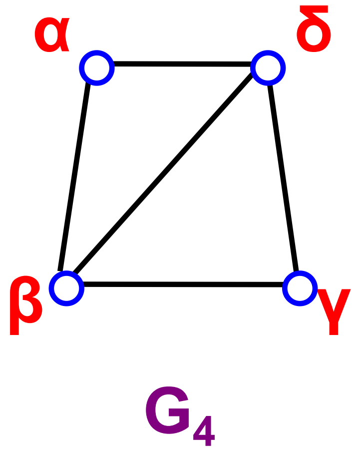
多重图



多重图



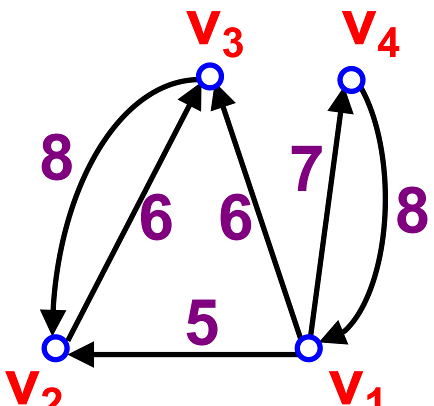
线图



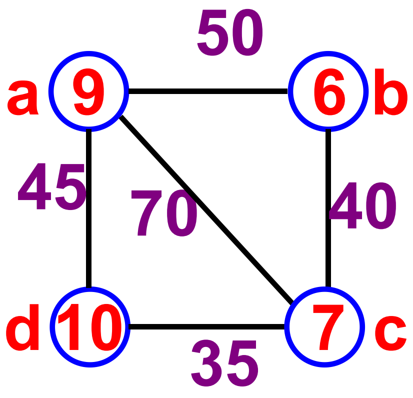
线图
简单图

定义9.2.7 **赋权图**(Weight Graph)G是一个**三重组**$<V, E, g>$或**四重组**$<V, E, f, g>$，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**。非赋权图称为**无权图**。

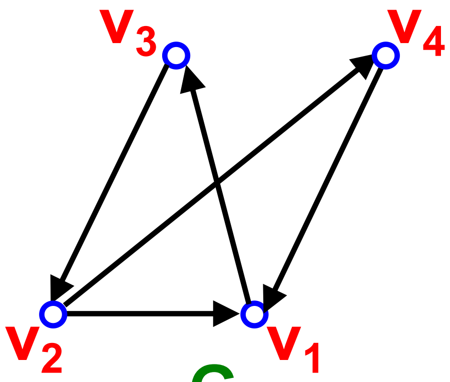
例6.9 下图所示的图哪个是赋权图，哪个是无权图？是赋权图的请写出相应的函数。



G_1



G_2



G_3

3. 按边或结点是否含权分类

记图$G_1 = <V_1, E_1, g_1>$，其中：

$g_1(<v_1, v_2>) = 5$　$g_1(<v_1, v_3>) = 6$

$g_1(<v_1, v_4>) = 7$　$g_1(<v_2, v_3>) = 6$

$g_1(<v_3, v_2>) = 8$　$g_1(<v_4, v_1>) = 8$

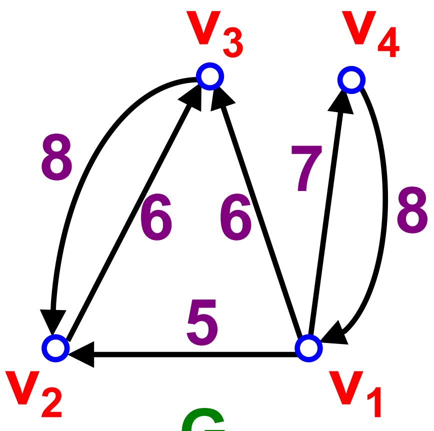
记图$G_2 = <V_2, E_2, f_2, g_2>$，其中：

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

$g_2((a, b)) = 50$，　$g_2((a, c)) = 70$，　$g_2((a, d)) = 45$，

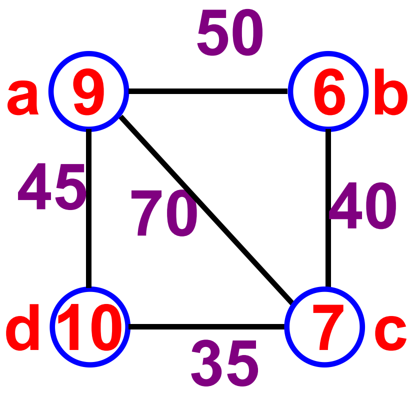
$g_2((b, c)) = 40$，　$g_2((c, d)) = 35$

例6.9 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。



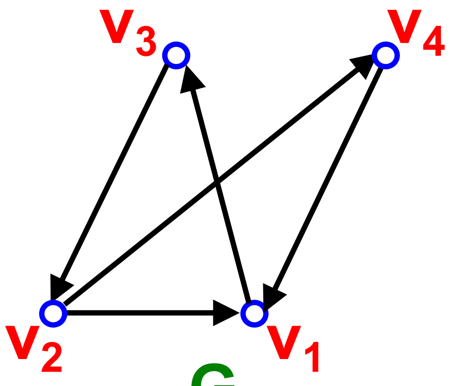
G_1

赋权图



G_2

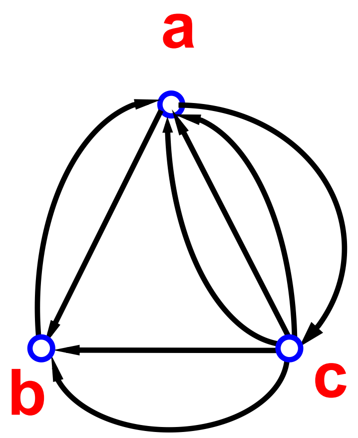
赋权图



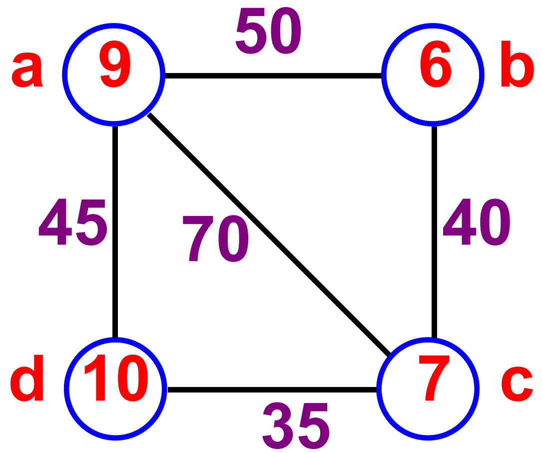
G_3

无权图

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

定义6.8 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$V_1 \subset V$或$E_1 \subset E$)，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$V_2 \neq \Phi$，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)

V_2导出的G的子图即为 $G - (V - V_2)$

解题小贴士——子图的判断

(1)子图的结点集和边集是G的结点集和边集的子集。

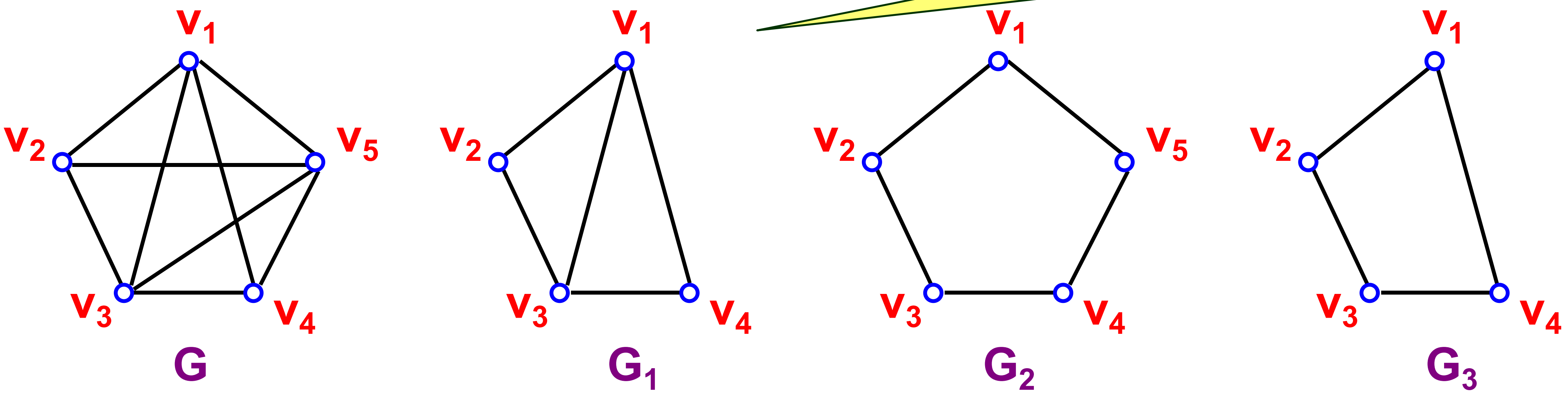
(2)真子图的结点集和边集是G的结点集或边集的真子集。

(3)生成子图与G的结点集相同而边集是子集。

(4)V_2的导出子图要求包含G中所有两个端点属于V_2的边。

判断下图中，图G_1、G_2和G_3是否是图G的子图、真子图、生成子图、导出子图？

注 每个图都是它自身的子图、生成子图和导出子图



G_1：子图、真子图、$\{v_1, v_2, v_3, v_4\}$的导出子图

G_2：子图、真子图、生成子图

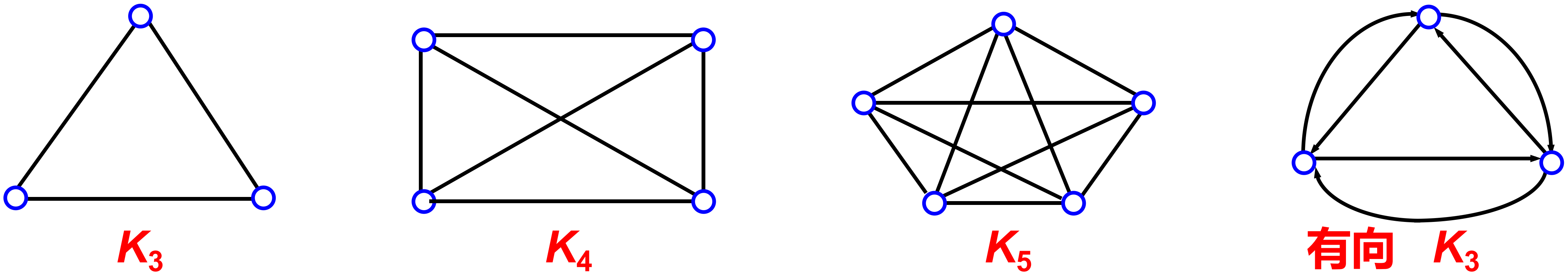
G_3：子图、真子图

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

定义9.2.9 设G = <V, E>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 = <V, E>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。

几个完全图



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 $P_n^2 = n(n-1)$

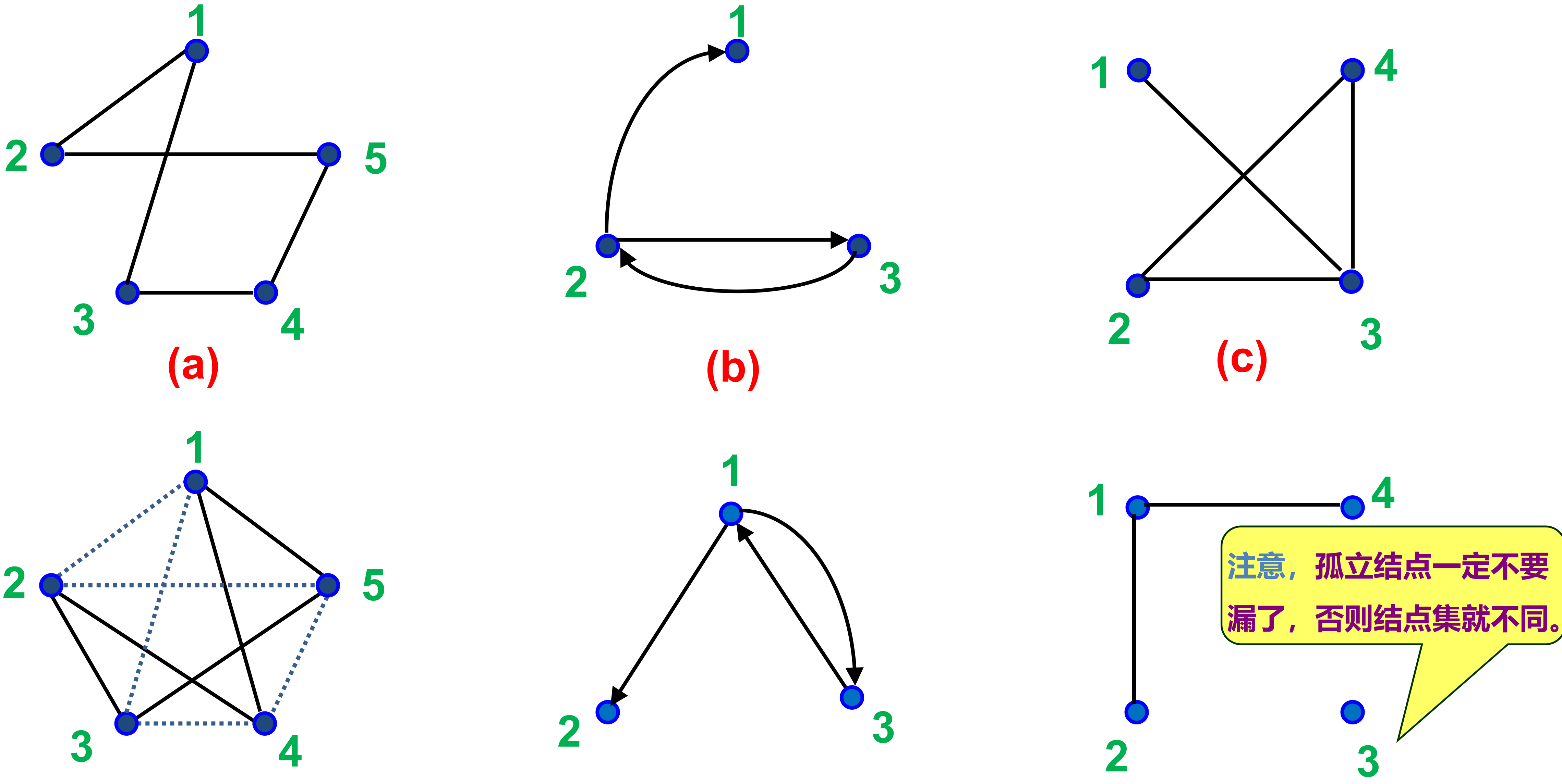
定义6.10 设$G = <V, E>$为简单图，$G' = <V, E_1>$为完全图，则称$G_1 = <V, E_1-E>$为G的**补图**(Complement of Graph)，记为G^C或$\overline{G}$。

注 在定义6.10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G^C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。
- 若$G_1 = G^C$，则$G = G_1^C$，即它们互为补图。
- K_n的补图为n个结点的零图$(n, 0)$。

解题小贴士——补图的计算

结点集相同，边集是相对于完全图边集的补集。


(a)
(b)
(c)
注意，孤立结点一定不要漏了，否则结点集就不同。

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

$$A^C = (a_{ij}^C)_{n\times n}$$

$$a_{ij}^C = \begin{cases} 1 - a_{ij}, & i \neq j \\ 0, & i = j \end{cases} \qquad i, j = 1, 2, 3, \cdots, n$$

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）。

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^C为G的补图，这样问题就转化为证明G或中至少有一个子图是完全图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^C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。

定义6.11 （1）图G = <V, E>中以结点v∈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 = <V, E>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 = <V, E>，**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)。

设图$G=<V,E>$，$V=\{v_1,v_2,\ldots,v_n\}$的邻接矩阵为 $A=\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix}$

- 若G是**无向图**，则A中第i行元素是由结点v_i为端点的边所决定，其中为1的元素数目等于v_i的度数，即 $deg(v_i)=\sum_{k=1}^{n}a_{ik}+a_{ii}=\sum_{k=1}^{n}a_{ki}+a_{ii}$
- 若G是有向图，则A中第i行元素是由结点vi为始点的边所决定，其中为1的元素数目等于vi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 $deg^{-}(v_i)=\sum_{k=1}^{n}a_{ki}$

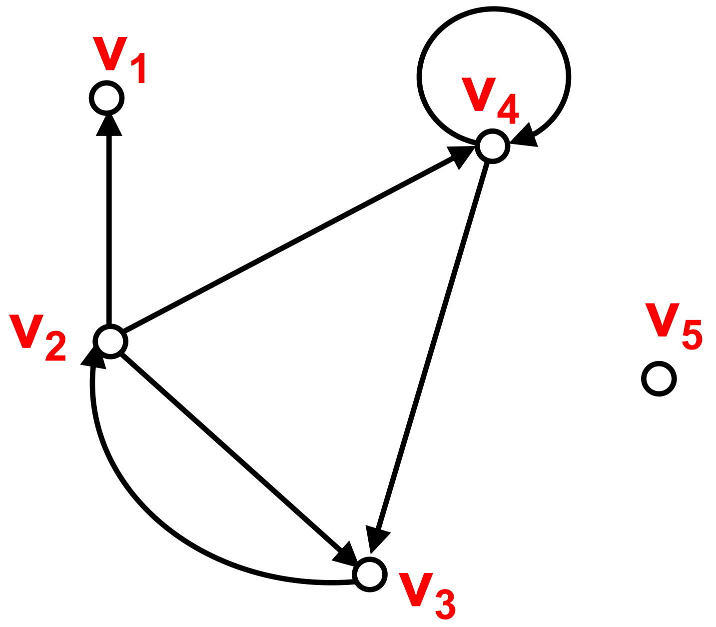
（1）结点*v*的度数就是以*v*为端点的边数（有环时计算两次）。

（2）结点*v*的出度就是以*v*为始点的边数。

（3）结点*v*的入度就是以*v*为终点的边数。

例6.13

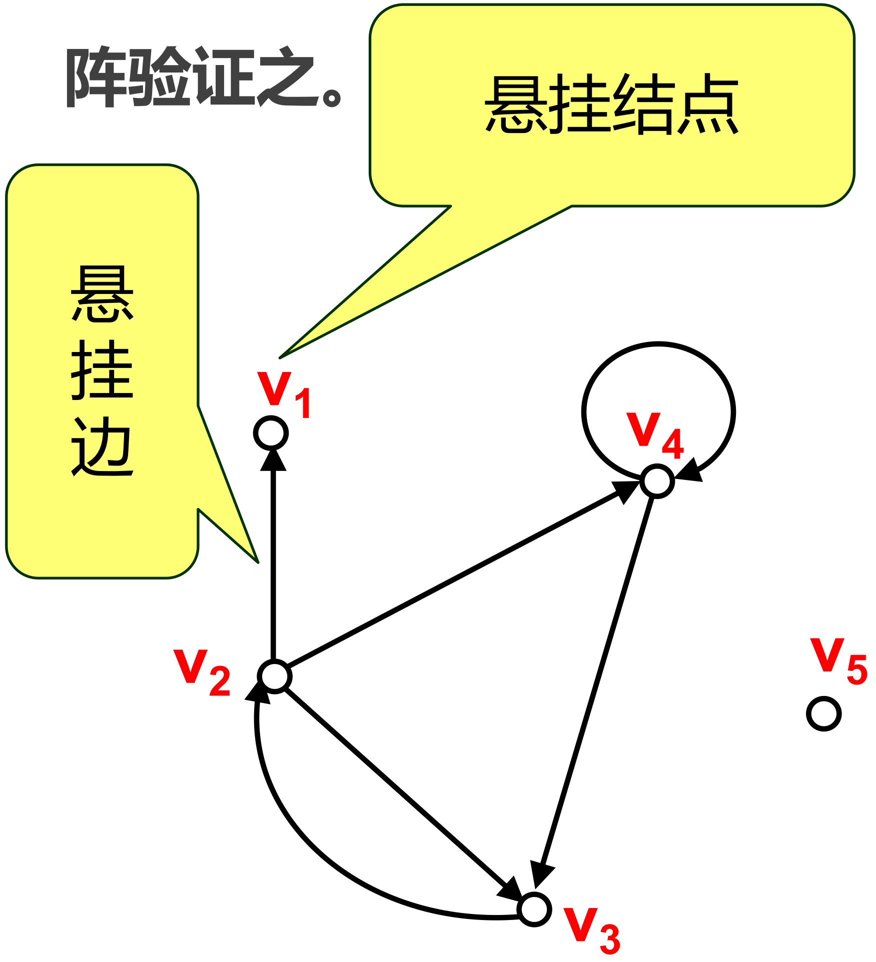
求下图中所有结点的度数、出度和入度，指出悬挂结点和为悬挂边，并用邻接矩阵验证之。



v	deg(v)	$deg^+(v)$	$deg^-(v)$
v_1	1	0	1
v_2	4	3	1
v_3	3	1	2
v_4	4	2	2
v_5	0	0	0

例6.13

求下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。



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

图中结点度数的总和等于边数的二倍，即设图G = <V, E>，则有

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

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。

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。

▸▸▸ 推论6.1 图中度数为奇数的结点个数为偶数。

证明 设图$G = <V, E>$，$V_1 = \{v \mid v \in V\text{且}deg(v)\text{为奇数}\}$，

$V_2 = \{v \mid v \in V\text{且}deg(v)\text{为偶数}\}$。

- **度数为奇数的结点称为奇度数结点(Odd Degree Point)**
- **度数为偶数的结点称为偶度数结点(Even Degree Point)**

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。

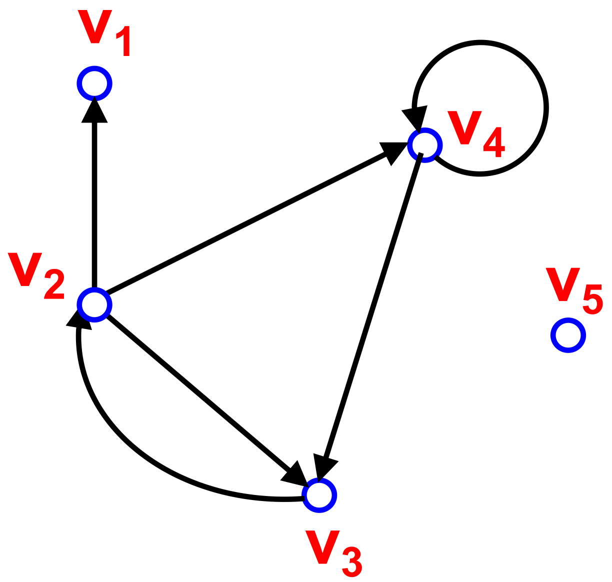
有向图中各结点的出度之和等于各结点的入度之和，等于边数，即设有向图 G = <V, E>，则有

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

证明 因为每条有向边具有1个始点和1个终点（环的始点和终点是同1个结点），因此，每条有向边对应1个出度和1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$|E|$。

以上两个定理及其推论都是非常重要的，应牢记、理解并灵活运用。

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



解题小贴士——握手定理的应用

（1）所有结点度数的总和等于边数的2倍。

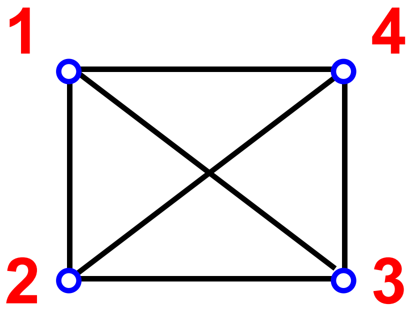
（2）奇度数结点的个数一定是偶数。

（3）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。

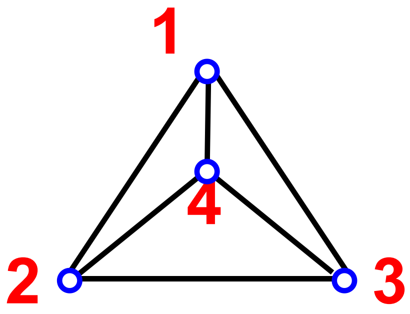
上图的度数序列为(1, 4, 3, 4, 0)。

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。

例如下面的两个图G_1和G_2实际上是同一个图，都是K_4。



G_1



G_2

设两个图G = <V, E>和G’ = <V’, E’>，如果存在**双射**函数g：V→V’，使得对于任意的e = (v_i, v_j)(或者$<v_i, v_j>$)∈E当且仅当e’ = $(g(v_i), g(v_j))$(或者$<g(v_i), g(v_j)>$)∈E’，并且e与e’的**重数相同**，则称G与G’**同构**(Isomor- phism)，记为G≅ 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

解题小贴士——图同构的判断

找到结点集之间的双射，满足两结点间有边当且仅当它们的函数值间有边并且方向和重数一致。

解题小贴士—— 图不同构的判断

至少满足下列情况之一的两个图是不同构的。

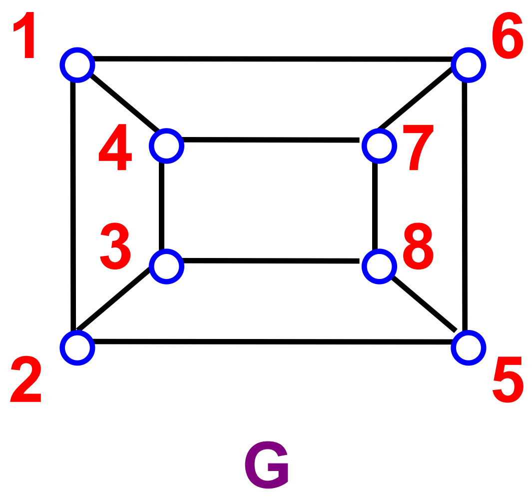
（1）结点数目不同。

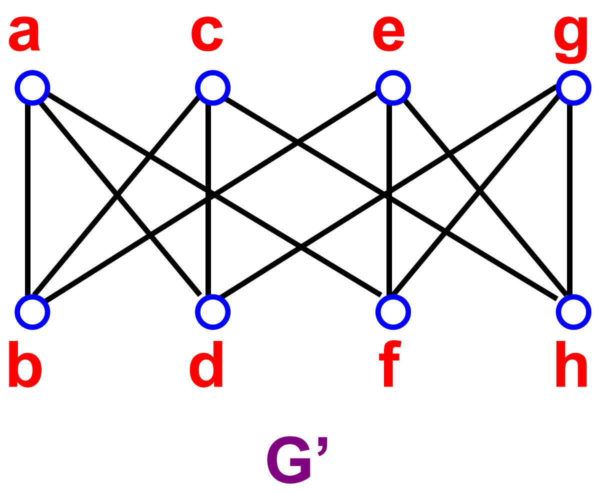
（2）边数不同。

（3）度数相同的结点数不同。

（4）有两个度数相同的结点的邻接点的度数不完全相同。

▸▸▸ 例9.2.14 证明下图中，G≅ G’。





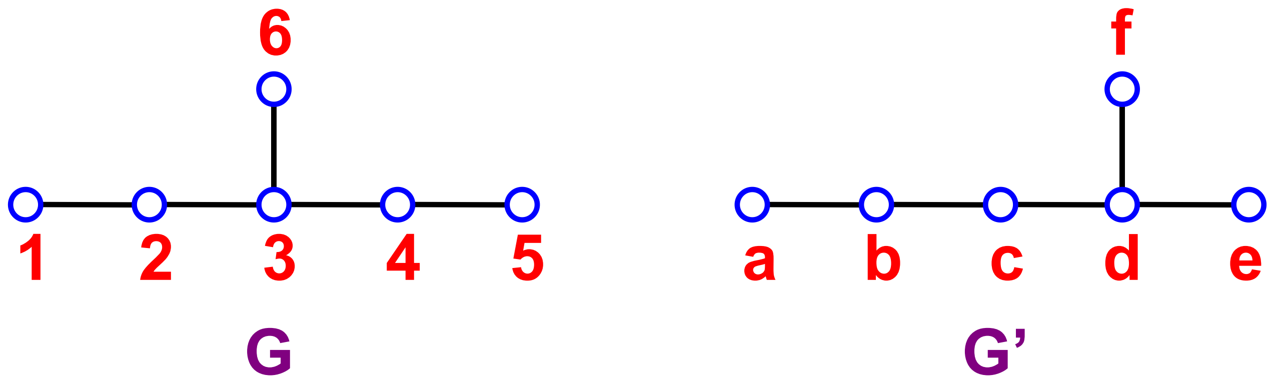
证明 构造结点之间的双射函数 g 如下：

g(1) = a， g(2) = b， g(3) = c， g(4) = d

g(5) = e， g(6) = f， g(7) = g， g(8) = h

容易验证，g 满足同构的定义，所以G≅ G’。

例9.2.15 证明下图中G与G’不同构。

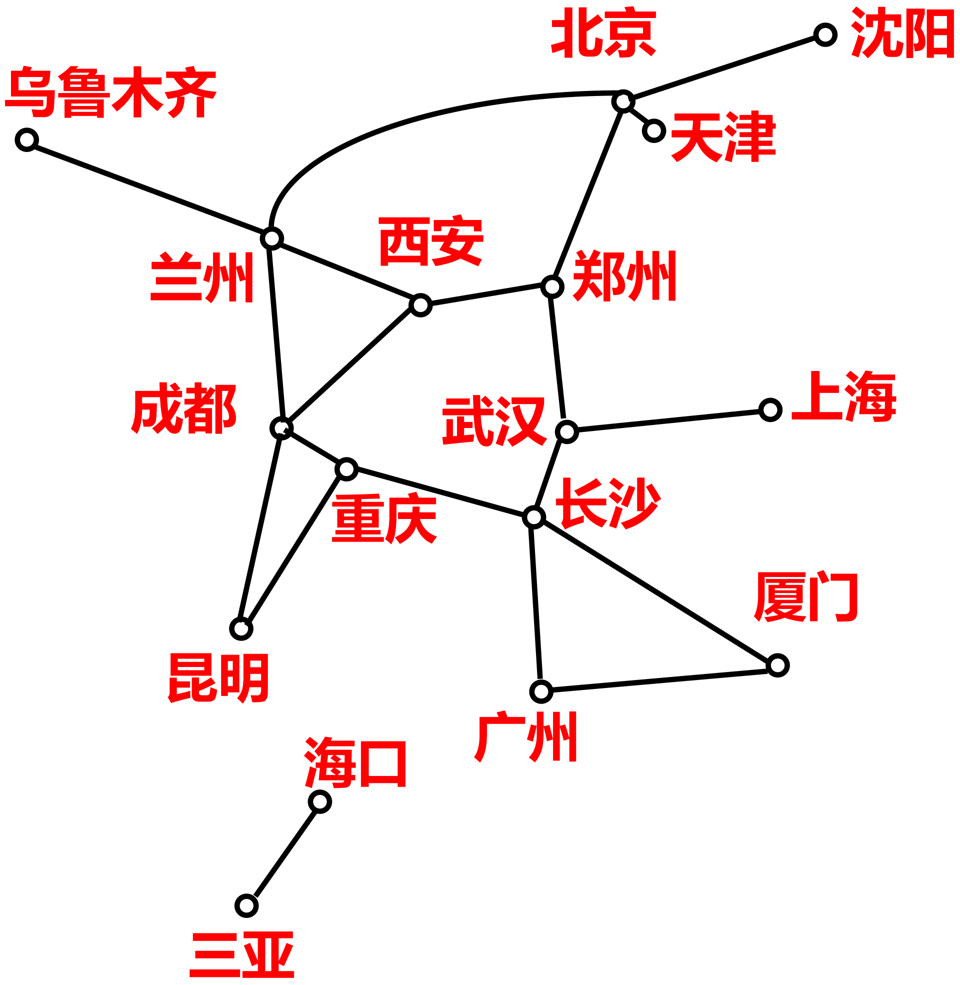


注意

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。

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他车站；而旅客不可能从成都乘火车到达三亚。这就引出了图的通路与连通的概念。



通路与回路是图论中两个重要的基本概念。本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6.14 给定图G = <V, E>中结点和边相继交错出现的序列Γ = $v_0e_1v_1e_2v_2\cdots e_kv_k$。

（1）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），$i = 1, 2, \cdots, k$，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)。

（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**。

（1）回路是通路的特殊情况。因而，当我们说某条通路，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$v_0e_1v_1e_2v_2\ldots e_nv_n$也可以用**边的序列**$e_1e_2\ldots e_n$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$v_0e_1v_1e_2v_2\ldots e_nv_n$也可以用**结点的序列**$v_0v_1v_2\ldots v_n$来表示。

解题小贴士——简单（基本）通（回）路的判断

（1）**简单通（回）路没有相同的边。**

（2）**基本通（回）路没有相同的结点，当然也没有相同的边。**

判断图G_1中的下列回路是否是简单回路、基本回路？并求其长度。

$v_3e_5v_4e_7v_1e_4v_3e_3v_2e_1v_1e_4v_3$

长度为6

不是简单回路

$v_3e_3v_2e_2v_2e_1v_1e_4v_3$

长度为4

是简单回路，不是基本回路

$v_3e_3v_2e_1v_1e_4v_3$

长度为3

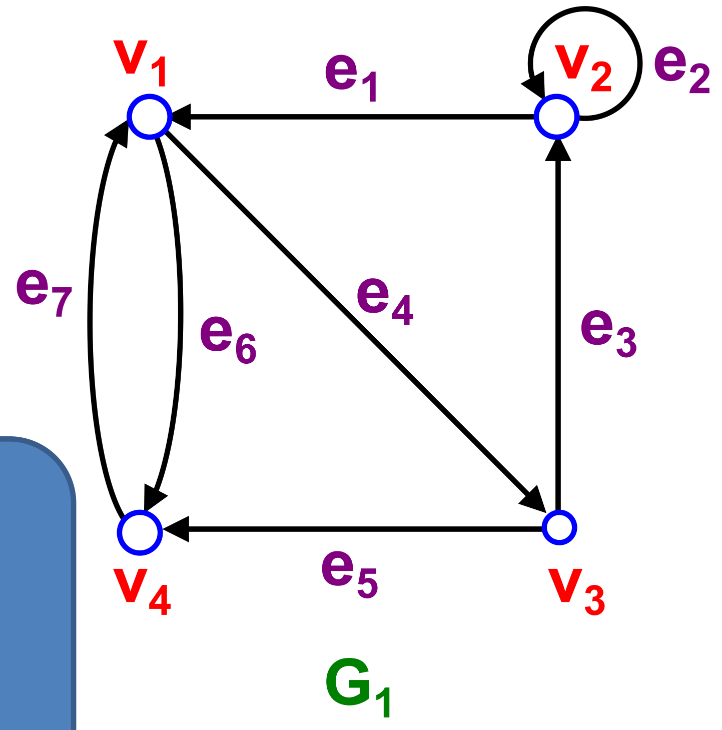
是基本回路

- 用边的序列表示

  $e_3e_2e_1e_4$

- 用结点的序列表示

  $v_3v_2v_2v_1v_3$



判断图G_2中的下列通路是否是简单通路、基本通路？并求其长度。

$v_1e_1v_2e_6v_5e_7v_3e_2v_2e_6v_5e_8v_4$ 长度为6

不是简单通路

$v_1e_5v_5e_7v_3e_2v_2e_6v_5e_8v_4$ 长度为5

是简单通路，不是基本通路

- 用边的序列表示

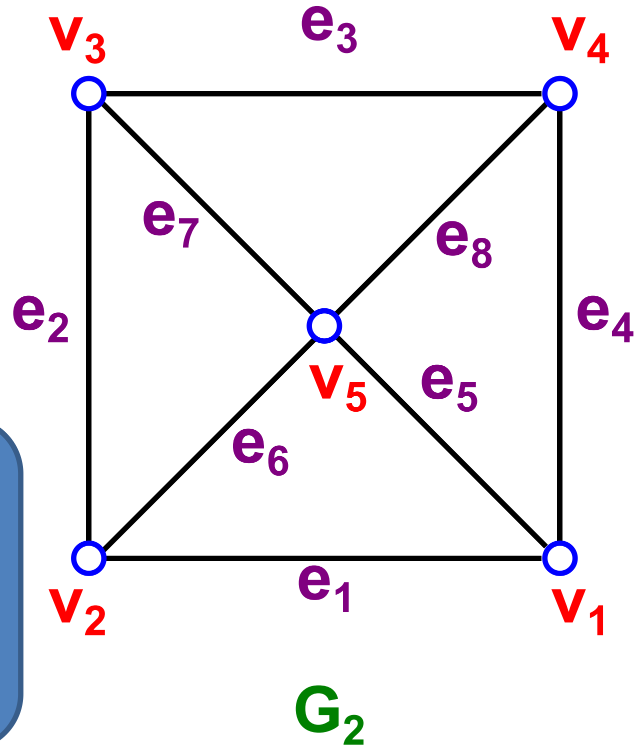
  $e_5e_7e_2e_6e_8$

- 用结点的序列表示

  $v_1v_5v_3v_2v_5v_4$

$v_1e_1v_2e_6v_5e_7v_3e_3v_4$ 长度为4

是基本通路



定理6.3 设$G = \langle V, E\rangle$为线图，$V = \{v_1, v_2, \cdots, v_n\}$，$A = (a_{ij})_{n\times n}$为$G$的邻接矩阵，$A^m = (a_{ij}^{(m)})_{n\times n}$。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m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。

- 观察图G的邻接矩阵$A = (a_{ij})_{n\times n}$
- a_{ij}表示从结点v_i到结点v_j长度为1的通路数目
- a_{ii}表示结点v_i到自身的长度为1的回路（即为环）数目
- 而A中所有元素之和 $\sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}$ 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。

- 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_p
- 对于任意的$p(1 \le p \le n)$，若存在通路$v_i v_p v_j$，必有$a_{ip} = 1$且$a_{pj} = 1$，即$a_{ip} \times a_{pj} = 1$。
- 反之，若不存在通路$v_i v_p v_j$，则必有$a_{ip} = 0$或$a_{pj} = 0$，即$a_{ip} \times a_{pj} = 0$。
- 于是从结点v_i到v_j长度为2的通路总数为：

$$a_{i1}a_{1j} + a_{i2}a_{2j} + \cdots + a_{in}a_{nj} = \sum_{p=1}^{n} a_{ip}a_{pj}$$

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

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

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。

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2的通路，从v_k到v_j存在长度为1的通路。

因而，$a_{ik}^{(2)}\geq 1$且$a_{kj}=1$，即$a_{ik}^{(2)}\times a_{kj}\geq 1$。

若 $a_{ik}^{(2)}$或 $a_{kj}=0$，从而$a_{ik}^{(2)}\times a_{kj}=0$，则从$v_i$经过$v_p$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：

$$a_{i1}^{(2)}a_{1j}+a_{i2}^{(2)}a_{2j}+\cdots+a_{in}^{(2)}a_{nj}=\sum_{k=1}^{n}a_{ik}^{(2)}a_{kj}$$

这正是$A^2 \times A = A^3$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。

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

而$a_{ip}^{(k)}$是结点v_i到v_p长度为k的通路数目，a_{pj}是结点v_p到v_j长度为1的通路数目，

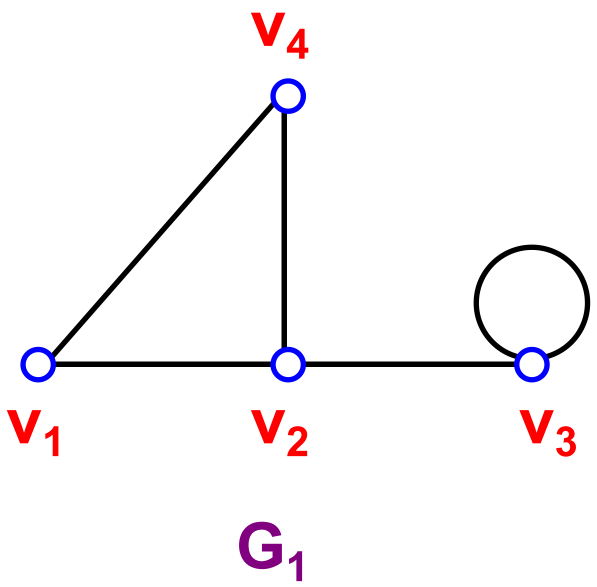
故 $a_{ip}^{(k)}\cdot a_{pj}$ 是从结点v_i经过v_p到结点v_j的长度为$k+1$的通路数目，

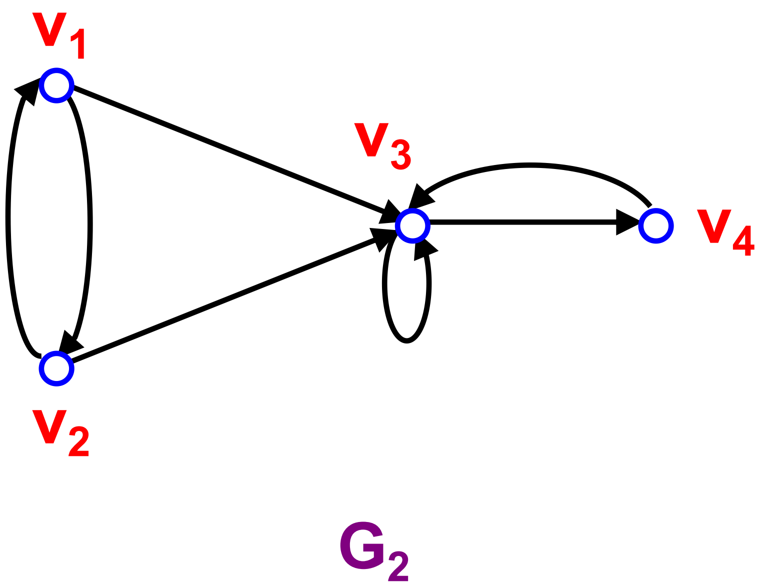
那么$\sum_{p=1}^{n}a_{ip}^{(k)}\cdot a_{pj}$ 是从结点v_i到结点v_j的长度为$k+1$的通路数目。

▸▸▸ 解题小贴士——结点vi到vj长度m的通路数目的计算

1. 写出邻接矩阵A
2. 计算A的m次幂A^m
3. A^m中第 i 行第 j 列元素即为所求

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。





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：

$$A(G_1)=\begin{pmatrix}0&1&0&1\\1&0&1&1\\0&1&1&0\\1&1&0&0\end{pmatrix}$$

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

下面计算邻接矩阵的幂，

$$(A(G_1))^2=\begin{pmatrix}0&1&0&1\\1&0&1&1\\0&1&1&0\\1&1&0&0\end{pmatrix}\times\begin{pmatrix}0&1&0&1\\1&0&1&1\\0&1&1&0\\1&1&0&0\end{pmatrix}=\begin{pmatrix}2&1&1&1\\1&3&1&1\\1&1&2&1\\1&1&1&2\end{pmatrix}$$

$$a_{13}^{(2)}=1 \qquad \sum_{i=1}^{4}\sum_{j=1}^{4}a_{ij}^{(2)}=21 \qquad \sum_{i=1}^{4}a_{ii}^{(2)}=9$$

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。

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

$$a_{13}^{(2)}=2 \qquad \sum_{i=1}^{4}\sum_{j=1}^{4}a_{ij}^{(2)}=13 \qquad \sum_{i=1}^{4}a_{ii}^{(2)}=5$$

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。

$$\left(A\left(G_1\right)\right)^3=\left(A\left(G_1\right)\right)^2\cdot A\left(G_1\right)=\begin{pmatrix}2&1&1&1\\1&3&1&1\\1&1&2&1\\1&1&1&2\end{pmatrix}\times\begin{pmatrix}0&1&0&1\\1&0&1&1\\0&1&1&0\\1&1&0&0\end{pmatrix}=\begin{pmatrix}2&4&2&3\\4&3&4&4\\2&4&3&2\\3&4&2&2\end{pmatrix}$$

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

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。

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

$$a_{13}^{(3)}=4 \qquad \sum_{i=1}^{4}\sum_{j=1}^{4}a_{ij}^{(3)}=22 \qquad \sum_{i=1}^{4}a_{ii}^{(3)}=4$$

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6.15 在图G = <V, E>中，$v_i, v_j \in V$。

（1）如果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v_i到v_j的**短程线**(Geodesic)；v_i到v_j的短程线的长度称为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$d(v_i, v_j) = \infty$。

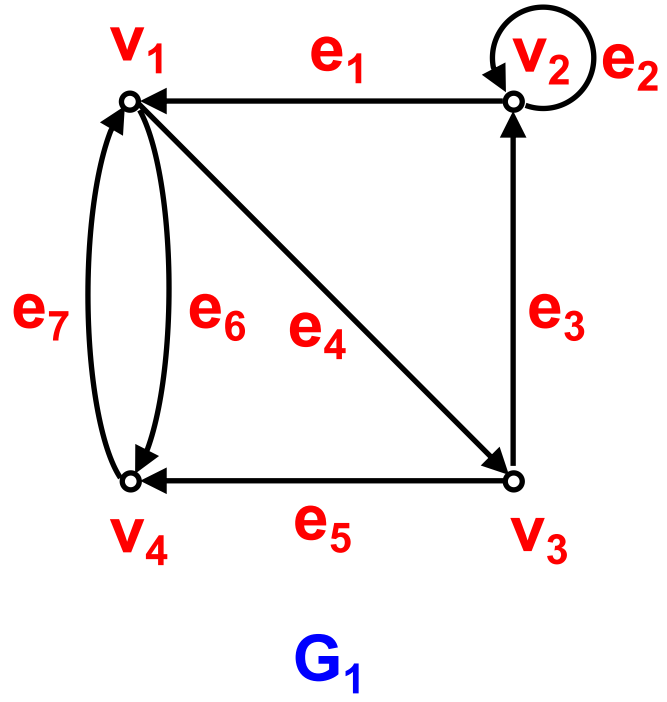
$d(v_i, v_j)$满足下列性质：

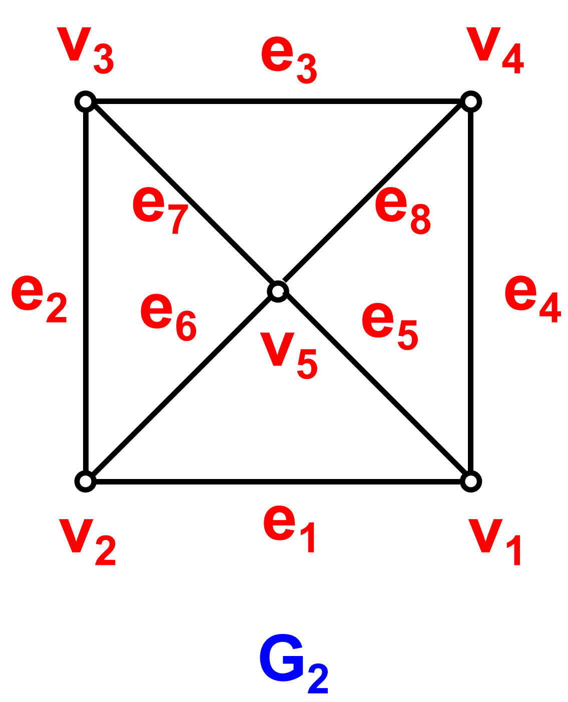
$d(v_i, v_j) \geq 0$；　　　$d(v_i, v_i) = 0$；

$d(v_i, v_k) + d(v_k, v_j) \geq d(v_i, v_j)$。

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$d(v_i, v_j) = d(v_j, v_i)$。





在图G_1中，$d(v_1, v_2) = 2$，$d(v_2, v_1) = 1$，$d(v_4, v_1) = d(v_1, v_4) = 1$，

$d(v_2, v_4) = 2$，$d(v_4, v_2) = 3$

在图G_2中，$d(v_1, v_3) = 2$，$d(v_3, v_4) = 1$，$d(v_2, v_4) = 2$

▸▸▸ 定理6.4

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。

推论6.2 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的基本通路。

定理6.5 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6.3 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的基本回路。

利用定理6.4和定理6.5，我们可以通过计算图的邻接矩阵及其幂的方法来判断v_i到v_j是否可达，以及v_i到v_j的距离。

设矩阵

$$B_m = I + A + A^2 + A^3 + \ldots + A^m$$ （I为n阶单位阵）

则B_m中的元素 $b_{ij}^{(m)} = a_{ij}^{(0)} + a_{ij}^{(1)} + a_{ij}^{(2)} + \cdots + a_{ij}^{(m)} = \sum_{k=0}^{m} a_{ij}^{(m)}$

$$\left(a_{ij}^{(0)} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad a_{ij}^{(1)} = a_{ij}, i, j = 1, 2, \ldots, m \right)$$

表示图G中结点v_i到结点v_j的长度小于等于m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m的回路总数。

设$G = \langle V, E\rangle$为线图，$V = \{v_1, v_2, \cdots, v_n\}$，$A = (a_{ij})_{n\times n}$为G的邻接矩阵，

$A^m = (a_{ij}^{(m)})_{n\times n}$，$m = 1, 2, \cdots, n$-1；$B_{n-1} = (b_{ij}^{(n-1)})_{n\times n} = I + A + A^2 + A^3 + \cdots + A^{n-1}$。

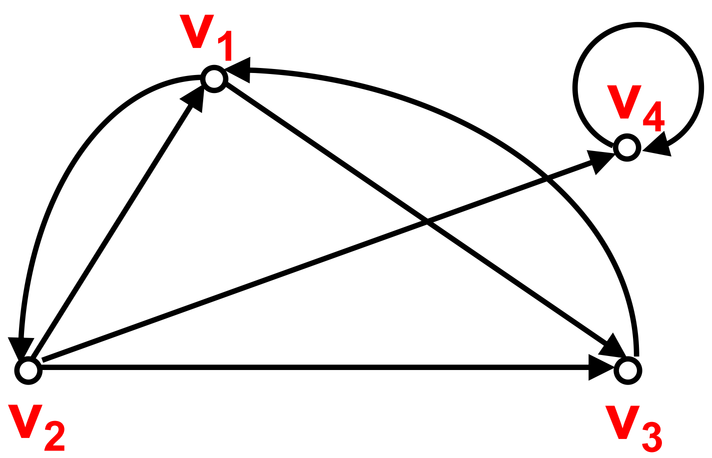
则有：如果$b_{ij}^{(n-1)} > 0$，那么v_i到v_j可达，否则不可达；并且

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

解题小贴士——结点间可达的判断与距离的计算

使用定理6.6，利用邻接矩阵及其幂与和的计算即可。

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。



计算图G的邻接矩阵及其2、3次幂分别为：

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

$d(v_1, v_2) = d(v_1, v_3) = d(v_2, v_1) = d(v_2, v_3) = d(v_2, v_4) = d(v_3, v_1) = 1$

$d(v_1, v_4) = d(v_3, v_2) = 2$

$d(v_3, v_4) = 3$

$d(v_4, v_1) = d(v_4, v_2) = d(v_4, v_3) = \infty$

计算 $$B_3 = I + A + A^2 + A^3 = \begin{pmatrix} 4 & 3 & 4 & 2 \\ 4 & 3 & 4 & 1 \\ 3 & 1 & 3 & 1 \\ 0 & 0 & 0 & 4 \end{pmatrix}$$

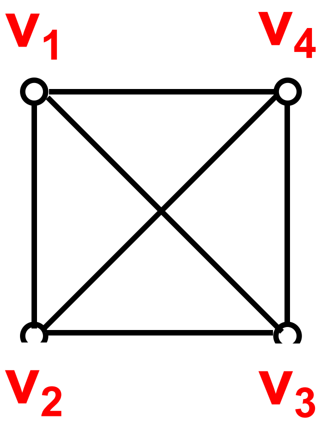
- **v_1到v_1, v_2, v_3, v_4都是可达的;**
- **v_2到v_1, v_2, v_3, v_4都是可达的;**
- **v_3到v_1, v_2, v_3, v_4都是可达的;**
- **v_4到v_4都是可达的，v_4到v_1, v_2, v_3都是不可达的。**

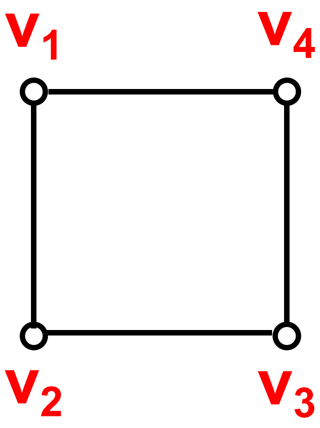
定义6.16

设$G = <V, E>$是一个**线图**，其中$V = \{v_1, v_2, \ldots, v_n\}$，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

$$p_{ij} = \begin{cases} 1, & \text{当}v_i\text{到}v_j\text{可达} \\ 0, & \text{否则} \end{cases} \qquad i, j = 1, 2, \ldots, n,$$

- 无向图的可达性矩阵是对称的，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。

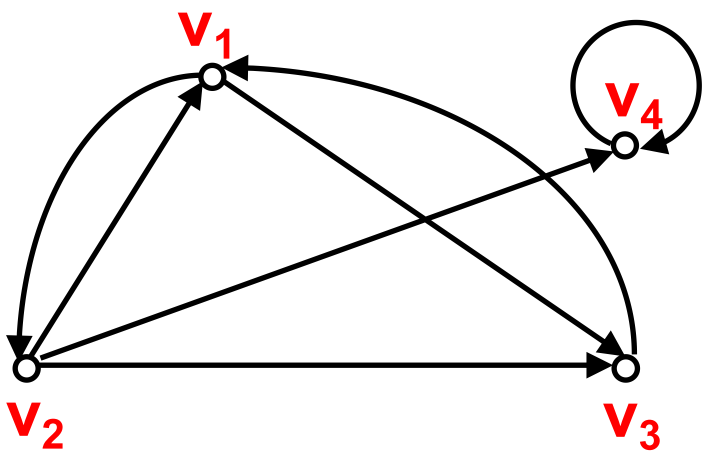




- 如果我们知道矩阵B_{n-1}，则只需将其中的非零元素写成1，就可得到可达性矩阵，即

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

- 例6.19图



$$B_3=\begin{pmatrix}4&3&4&2\\4&3&4&1\\3&1&3&1\\0&0&0&4\end{pmatrix}\Rightarrow P=\begin{pmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\0&0&0&1\end{pmatrix}$$

设G = <V, E>为线图，A、P分别是G的邻接矩阵和可达性矩阵，则有

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

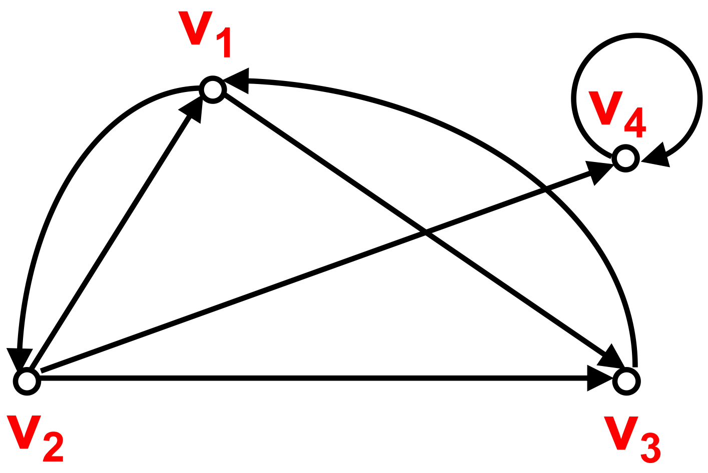
这里，$A^{(i)}$表示做矩阵布尔积的i次幂。

解题小贴士——可达性矩阵的计算

使用定理6.7，利用邻接矩阵及其布尔乘积与布尔并的计算即可。

求右图中图G中的可达性矩阵。

解 图G的邻接矩阵及其2、3次布尔乘法幂分别为：



与我们利用B_3求得的结果完全一致

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

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

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 迪杰斯特拉

如何求出简单无向赋权图$G=<V,E>$中从结点v_1到v_n的最短通路，目前公认最好的算法是由迪杰斯特拉（Dijkstra）在1959年提出的，称为**Dijkstra算法**，其基本思想如下。

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（最短通路长度的上界）。

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。

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

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

(2) **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$P=P\cup\{v_k\}$，$T=T-\{v_k\}$。

(3) **修改**：修改与v_k相邻的结点的T标号值。$\forall v_i \in V$，

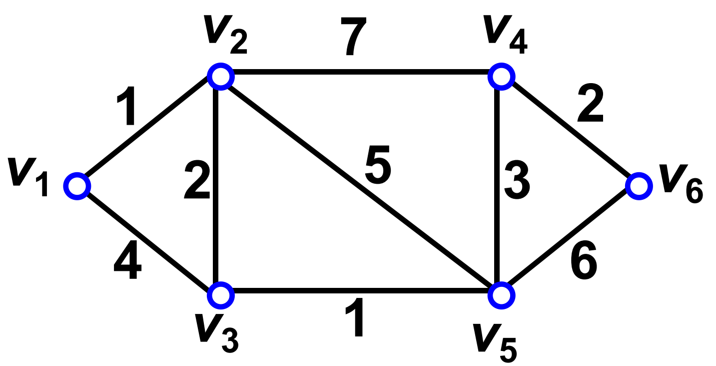
$$d(v_i)=\begin{cases} d(v_k)+w(v_k,v_i), & d(v_k)+w(v_k,v_i)<d(v_i) \\ d(v_i), & \end{cases}$$

(4) **重复**（2）和（3），直到v_n改为P标号为止。

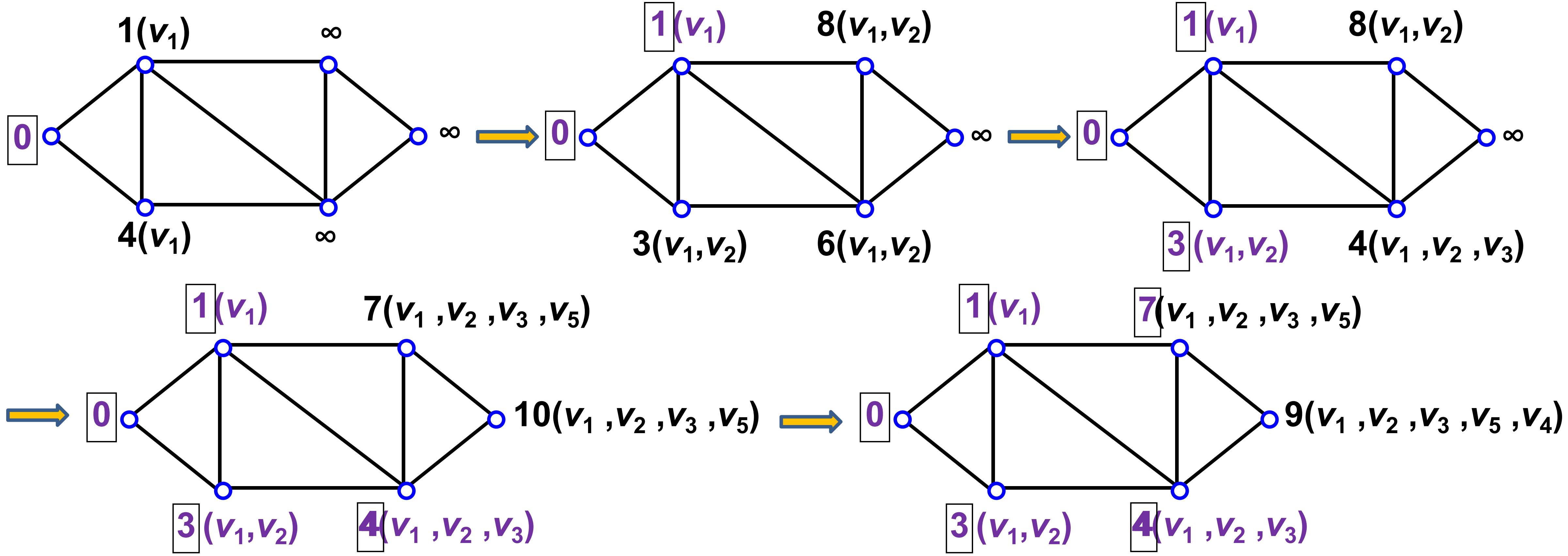
当v_n归入P而正好P = V时，不仅求出了从v_1到v_n的最短通路，而且实际上求出了从v_1到所有结点的最短通路。

上述算法的正确性是显然的。因为在每一步，设P中每一结点的标号是从v_1到该结点的最短通路的长度（开始时，P = {v_1}，d(v_1) = 0，这个假设是正确的），故只要证明上述d(v_i)是从v_1到v_i的最短通路的长度即可。事实上，任何一条从v_1到v_i通路，若通过T的第一个结点是v_p，而v_p≠v_i的话，由于所有边的长度非负，则这种通路的长度不会比d(v_i)小。

求简单无向赋权图中v_1到v_6的最短通路。

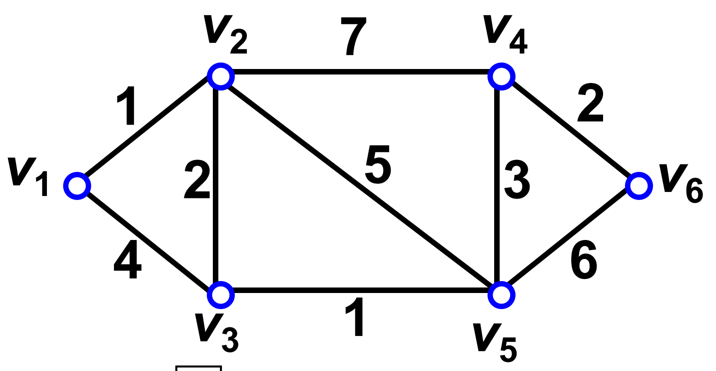


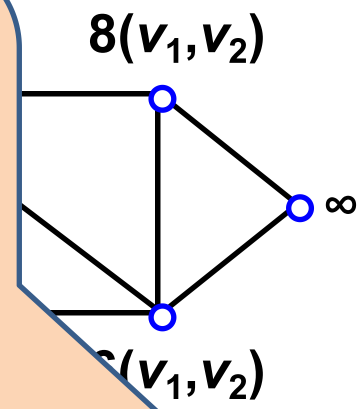
解 根据Dijkstra算法，求解过程如下。

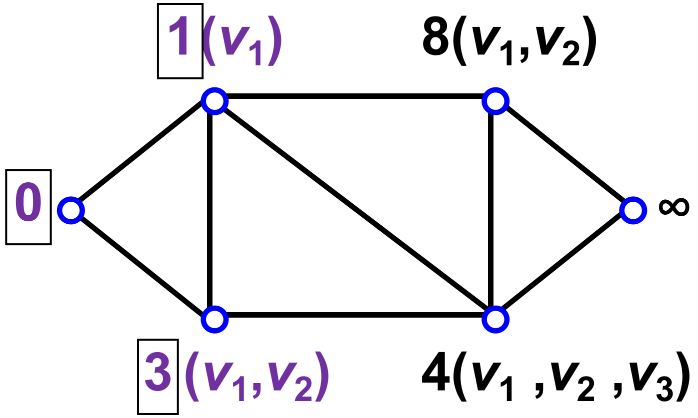


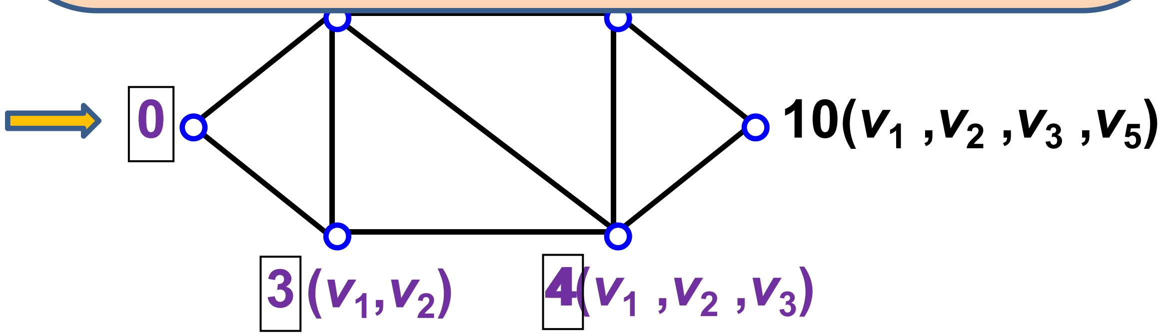
求简单无向赋权图中v_1到v_6的最短通路。

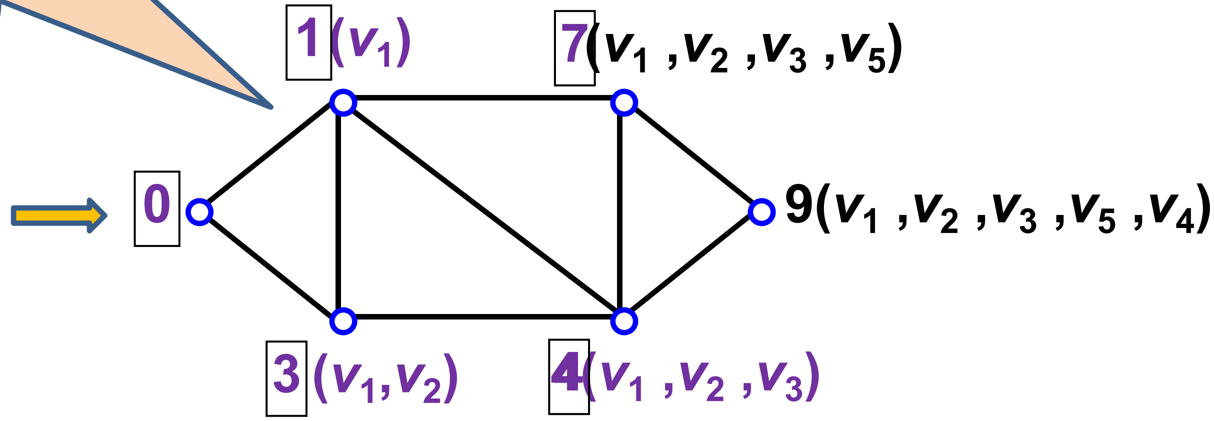
解 根据Dijkstra算法，求解过程如下。











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，例如，v_1到v_5的最短通路为$v_1v_2v_3v_5$，其长度为4，等等。

算法6.2 Floyd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$D^{(k)}=(d_{ij}^{(k)})_{n\times n}$的元素$d_{ij}^{(k)}$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$D^{(k-1)}=(d_{ij}^{(k-1)})_{n\times n}$，则$D^{(k)}=(d_{ij}^{(k)})_{n\times n}$的元素规定为

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

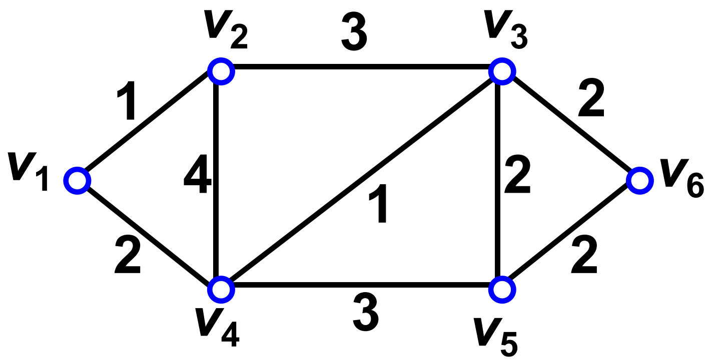
运算过程从$k=1$开始，让i和j分别取遍从1到n的所有值，然后k增加1，如此反复进行，直到$k=n$为止。这时$D^{(n)}=(d_{ij}^{(k)})_{n\times n}$的元素$d_{ij}^{(k)}$就是从$v_i$到$v_j$的最短通路长度。

算法6.2 Floyd算法

算法的正确性是显然的。

Floyd算法算法求出了任意两个结点间的最短通路的长度，从而很容易得出相应的最短通路。

求简单无向赋权图中的所有最短通路。



解 根据根据Floyd算法，有：

$$D^{(0)}=\begin{pmatrix}0&1&\infty&2&\infty&\infty\\1&0&3&4&\infty&\infty\\\infty&3&0&1&2&2\\2&4&1&0&3&\infty\\\infty&\infty&2&3&0&2\\\infty&\infty&2&\infty&2&0\end{pmatrix}\quad D^{(1)}=\begin{pmatrix}0&1&\infty&2&\infty&\infty\\1&0&3&3&\infty&\infty\\\infty&3&0&1&2&2\\2&3&1&0&3&\infty\\\infty&\infty&2&3&0&2\\\infty&\infty&2&\infty&2&0\end{pmatrix}$$

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

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

故v_2到v_6的最短通路长度为5，其最短通路为$v_2v_3v_6$，其余类似。

定义6.17 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≥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。

定义9.3.4 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≥1）都是连通图
- 多于一个结点的零图都是非连通图

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。

无向图G=<V, E>中结点之间的可达关系R定义如下：

R = {<u, v> | u, v∈V，u到v可达}，

则R是V上的等价关系。

证明（1）对任意v∈V，由于规定任何结点到自身总是可达的，因此<v, v>∈R，故R是自反的；

（2）对任意u, v∈V，若<u, v>∈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<v, u>∈R，故R是对称的；

（3）对任意u, v, w∈V，若<u, v>∈R，<v, w>∈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<u, w>∈R，故R是传递的。

由（1）、（2）、（3）知，R是V上的等价关系。

无向图G=<V, E>中结点之间的可达关系R定义如下：

$$R = \{<u, v> \mid u, v \in V，u\text{到}v\text{可达}\},$$

则R是V上的等价关系。

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6.18

无向图G=<V, E>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。

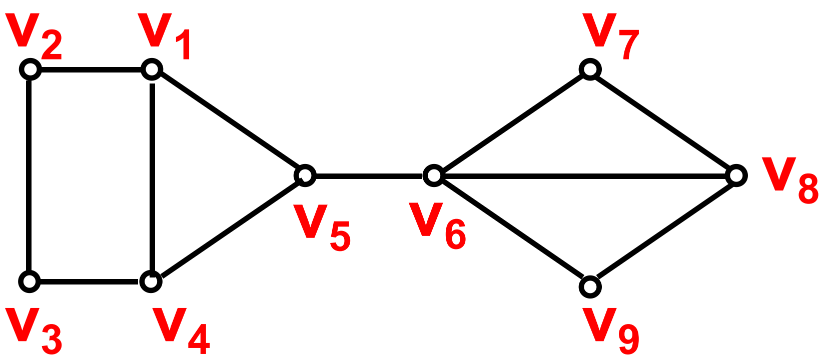
- 无向图G是连通图当且仅当p(G) = 1;
- 每个结点和每条边都在且仅在一个连通分支中。

解题小贴士——无向图连通性的判断及其连通分支个数计算

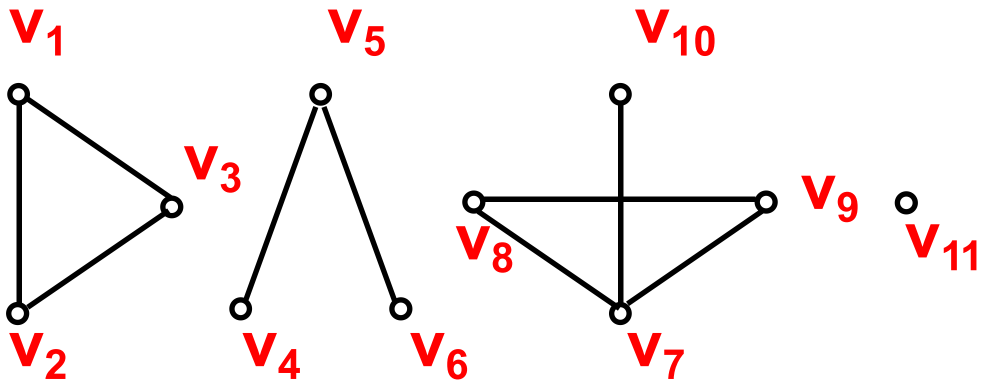
(1) 利用结点之间的可达关系是等价关系，计算出所有等价类，每个等价类导出的子图就是一个连通分支，不同等价类的数目就是连通分支个数，连通分支个数为1即为连通图。

(2) 对于给出图形的无向图，直接观察图形易得相关结果。

判断下图中图G_1和G_2的连通性，并求其连通分支个数。



G_1



G_2

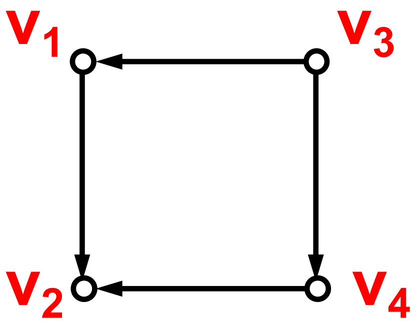
连通图

$p(G_1) = 1$

非连通图

$p(G_2) = 4$

由于有向图中边都有方向性，因此有向图结点之间的可达关系仅仅具有自反性和传递性，而不具有对称性。例如，下图中v_3到v_2可达，但v_2到v_3不可达。因此，可达关系不是等价关系。



设G = <V, E>是一个有向图，

1. **略去**G中所有有向边的**方向**得无向图G’，如果无向图G’是**连通图**，则称有向图G是**连通图**或称为**弱连通图**(Weakly Connected Graph)。否则称G**是非连通图**；

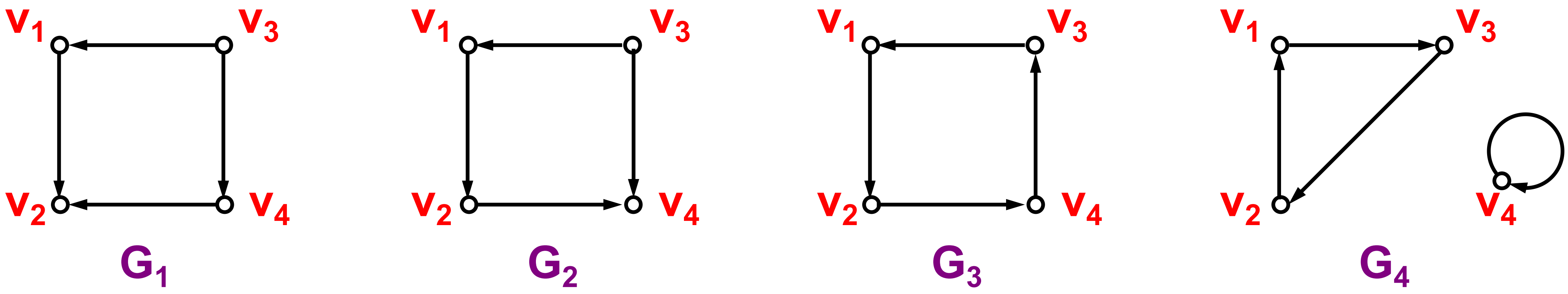
2. 若G中**任何一对结点**之间**至少有一个结点到另一个结点是可达的**，则称G是**单向连通图**(Unilaterally Connected Graph)；

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)。

设G = <V, E>是一个有向图，

- 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。
- 上述二命题的逆均不成立。

判断下图中4个图的连通性。



弱连通图

单向连通图

强连通图

非连通图

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。

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。

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$v_1,v_2,\ldots,v_n$，因为v_i到v_{i+1}是可达的，$i=1,2,\ldots,n-1$，且v_n到v_1是可达的，所以v_i到v_{i+1}存在通路，$i=1,2,\ldots,n-1$，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6.10 有向图G是单向连通图的充分必要条件是G中存在一条经过所有结点的通路。

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**

解题小贴士——有向图连通性的判断

（1）能够找到一条经过所有结点的回路，则是强连通图。

（2）能够找到一条经过所有结点的通路，则是单向连通图。

（3）将有向边看作无向边的无向图是连通图，则是弱连通图；否则是非连通图。

（4）利用邻接矩阵*A*和可达性矩阵*P*来判断有向图的连通性，适用于计算机处理。

▸▸▸ 6.6.1 网络的结构

自从克希荷夫运用图论从事电路网络的拓扑分析以来，尤其是近几十年来，网络理论的研究和应用十分引人注目，电路网络、运输网络、信息网络等与工程和应用紧密相关的课题受到了高度的重视，其中多数问题都与优化有关，涉及到问题的费用、容量、可靠性和其它性能指标，有重要的应用价值。网络应用的一个重要方面就是通讯网络。如电话网络、计算机网络、管理信息系统、医疗数据网络、银行数据网络、开关网络等等。这些网络的基本要求是网络中各个用户能够快速安全地传递信息，不产生差错和故障，同时使建造和维护网络所需费用低。因此通讯网络涉及的因素很多，我们就不详细介绍，仅说明一些基本知识。

例6.26 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。

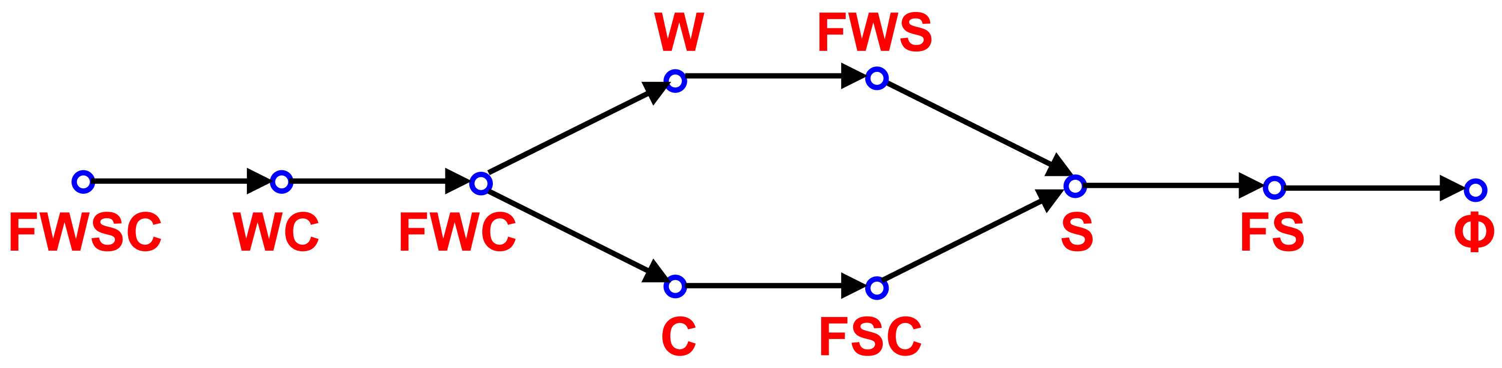
若用FWSC表示人和其它三样东西在河的原岸的状态，这样原岸全部可能出现的状态为以下16种：

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。

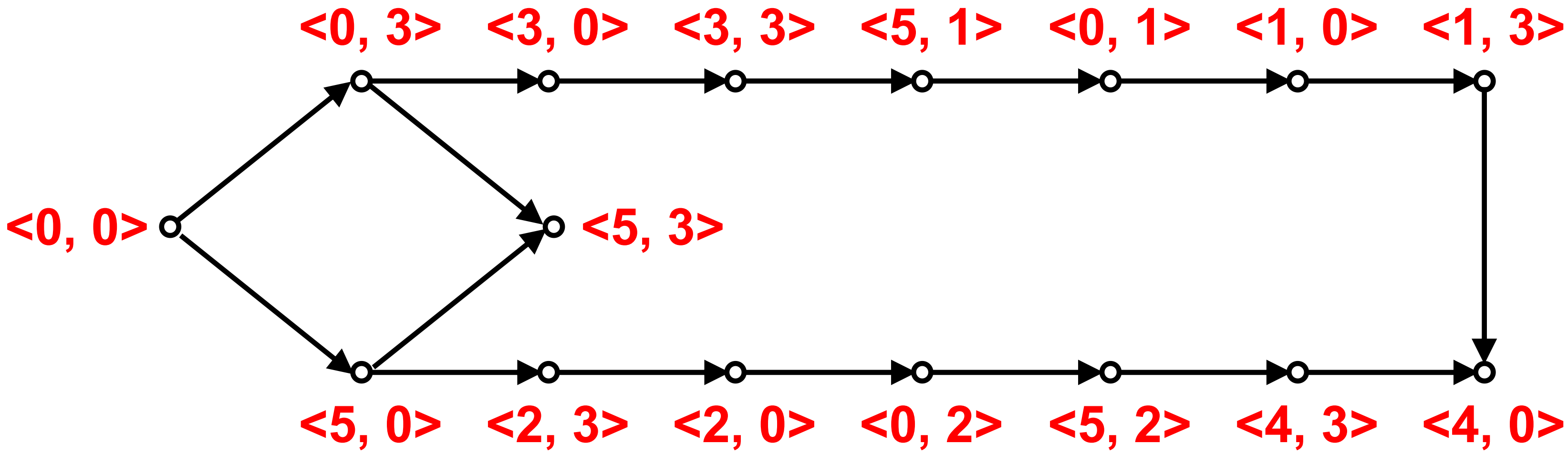
以这10种状态为结点，以摆渡前原岸的一种状态与摆渡一次后仍在原岸的状态所对应的结点之间的联线为边做有向图G，如图



图中给出了两种方案，方案为图中的从FWSC到Φ的不同的基本通路，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。

例6.27 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<B,C>表示桶b和桶c装酒的情况，可得下图。



例6.27 … a装

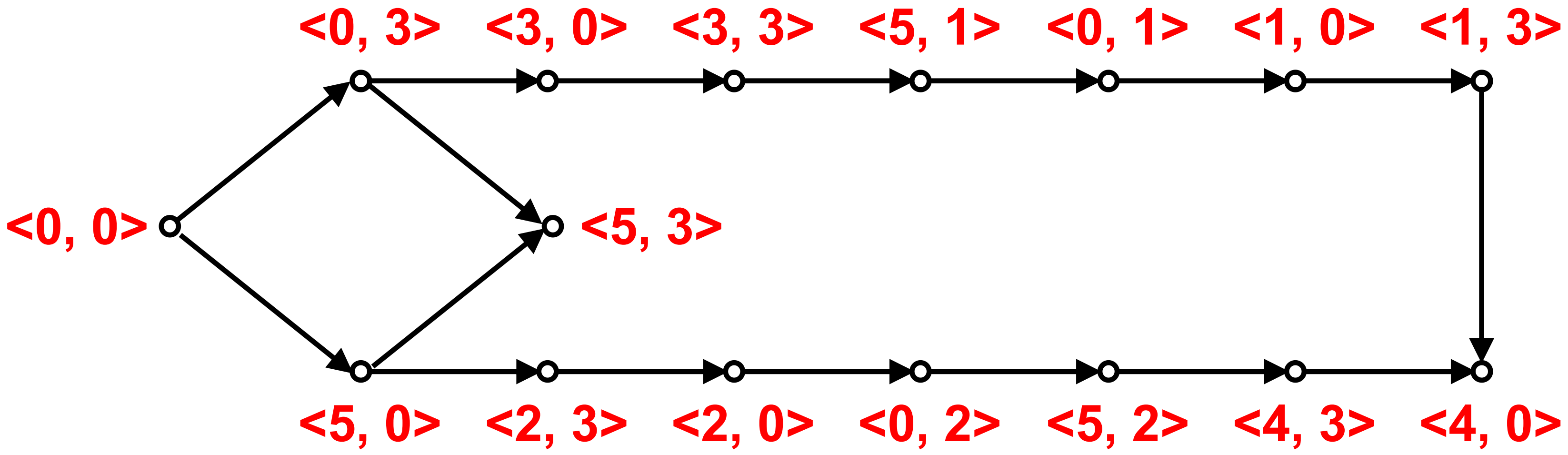
满了酒， … 问怎样

均分?

解 用<E …

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。



THANKS