第六章 图与网络分析

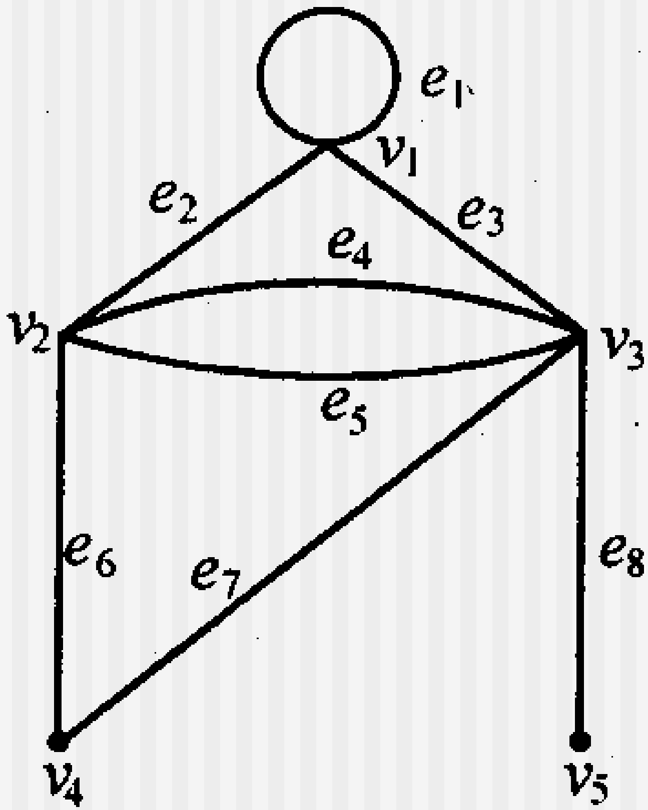
1．图的基本概念与模型
2．树图和图的最小部分树（第一节课）
3．最短路问题
4．中国邮路问题（第二节课）
5．网络的最大流（第三节课）

§1．图的基本概念与模型

图（graph）　图由若干个点（称作顶点或节点，vertex）和若干条连接两两顶点的线段（称作边，edge）组成，可以表示为顶点和边的集合，记作

$$G=\{V, E\}$$

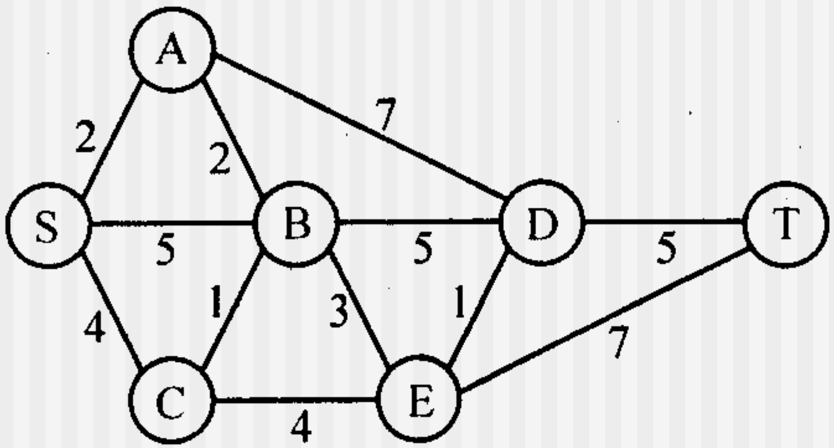
其中V表示点的集合，E表示边的集合。



一般图中的点数记作p（G），边数记作q（G），简记为p、q。

§1. 图的基本概念与模型

网络图 给图中的点和边赋以具体的含义和权数，如距离、费用、容量等，这样的图称为网络图（赋权图），记作N 。

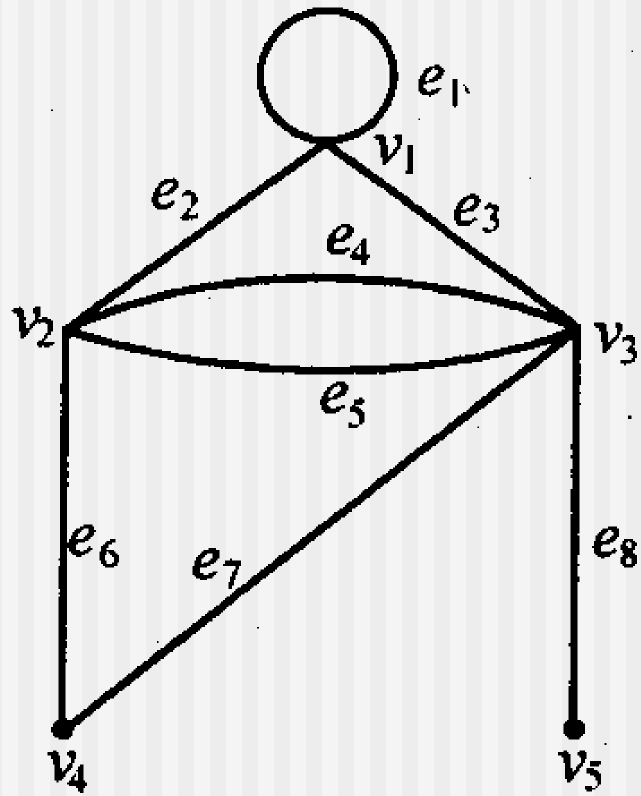


顶点 v 边 e

边可用它所连接的点表示 $e_3 = [v_1, v_3]$

§1．图的基本概念与模型

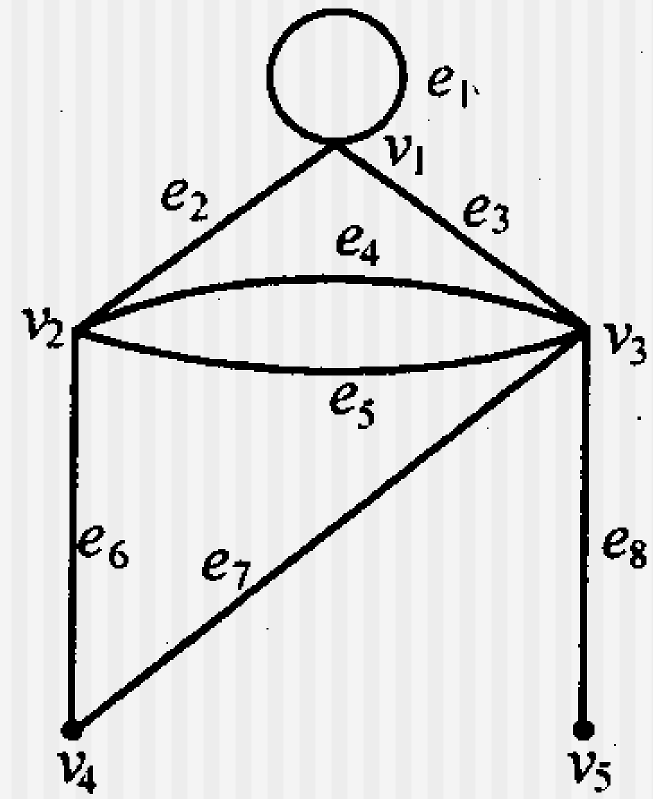
端点，关联边，相邻　若边e可表示为 $e = [v_i，v_j]$，则称 v_i 和 v_j 是边e 的端点，e 为 v_i 和 v_j 的关联边。若点 v_i、v_j 与同一条边关联，则称点 v_i 和 v_j 相邻；若边 e_i 和 e_j 具有公共的端点，则称边 e_i 和 e_j 相邻。



§1. 图的基本概念与模型

环，多重边，简单图

如果边e的两个端点相重，称该边为环，例如 e_1。如果两个点之间的边多于一条，称为多重边，例如e_4和e_5。无环、无多重边的图称为简单图。



§1．图的基本概念与模型

次，奇点，偶点，孤立点 与某一个点v_i相关联的边的数目称为点的次（也称作度），记作$d(v_i)$。次为奇数的点称作奇点，次为偶数的点称作偶点，次为0 的点称作孤立点，次为1的点称作悬挂点，悬挂点的关联边称为悬挂边。

定理1 图$G=\{V，E\}$中，所有点的次之和是边数的两倍，即

$$\sum_{v\in V} d(v)=2q$$

定理2 任一图中，奇点的个数为偶数。

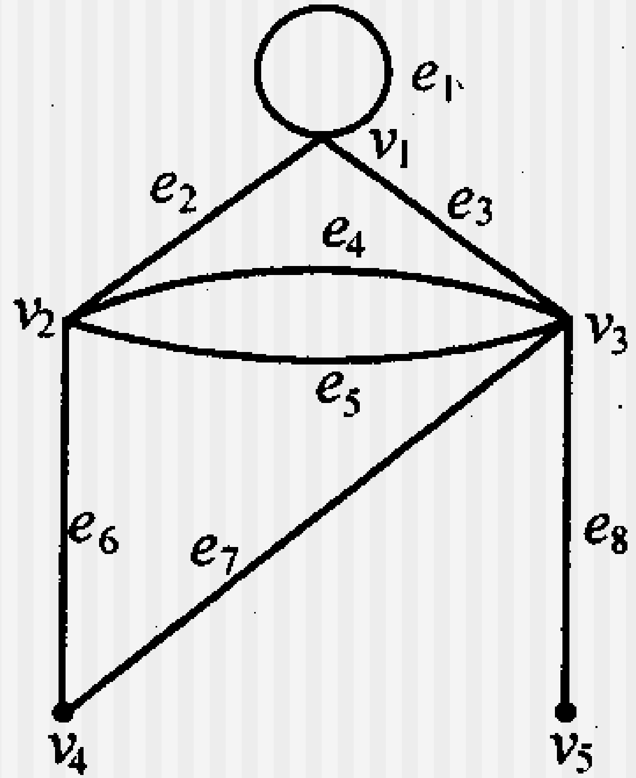
$$\sum_{v\in V_1} d(v)+\sum_{v\in V_2} d(v)=\sum_{v\in V} d(v)=2q$$

§1. 图的基本概念与模型

链，圈，连通图 如果点和边的交替序列

$$\mu=\{v_{j_1},e_1,v_{j_2},e_2,\cdots,e_{k-1},v_{j_k}\}$$

中，e_1、e_2、…、e_{k-1}互不相同，且任意v_{j_i}和v_{j_i}（$1 \le i \le k-1$）均相邻，则称μ为链。如果链中所有顶点都不相同，则称为路。



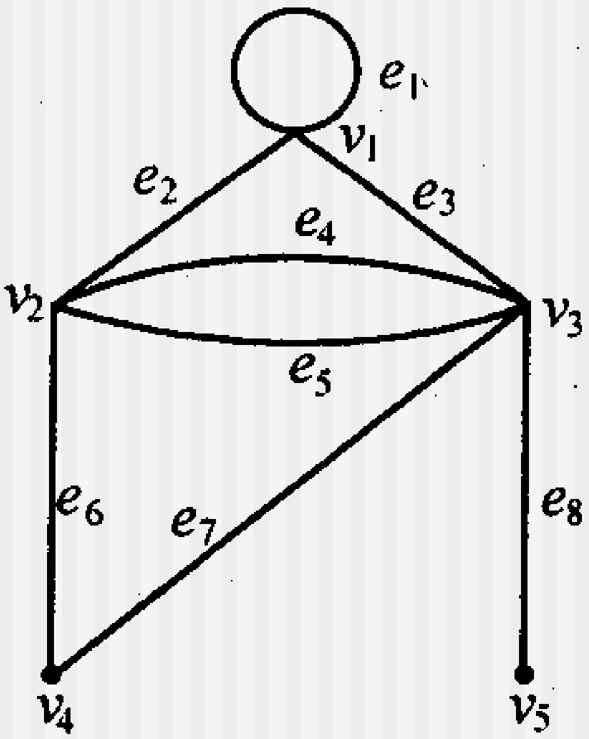
路：$\mu_2=\{v_5,e_8,v_3,e_7,v_4\}$

链：$\mu_1=\{v_5,e_8,v_3,e_3,v_1,e_2,v_2,e_4,v_3,e_7,v_4\}$

§1. 图的基本概念与模型

起点和终点相重合的链称作**圈**；起点与终点相重合的路称作**回路**。

若在一个图中，如果每一对顶点之间至少存在一条链，称这样的图为**连通图**，否则称该图是不连通的。



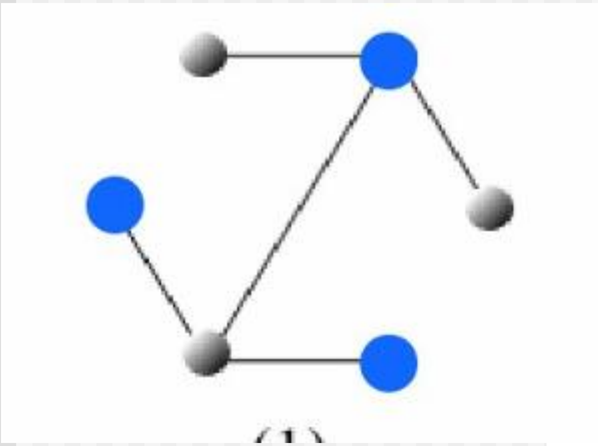
§1．图的基本概念与模型

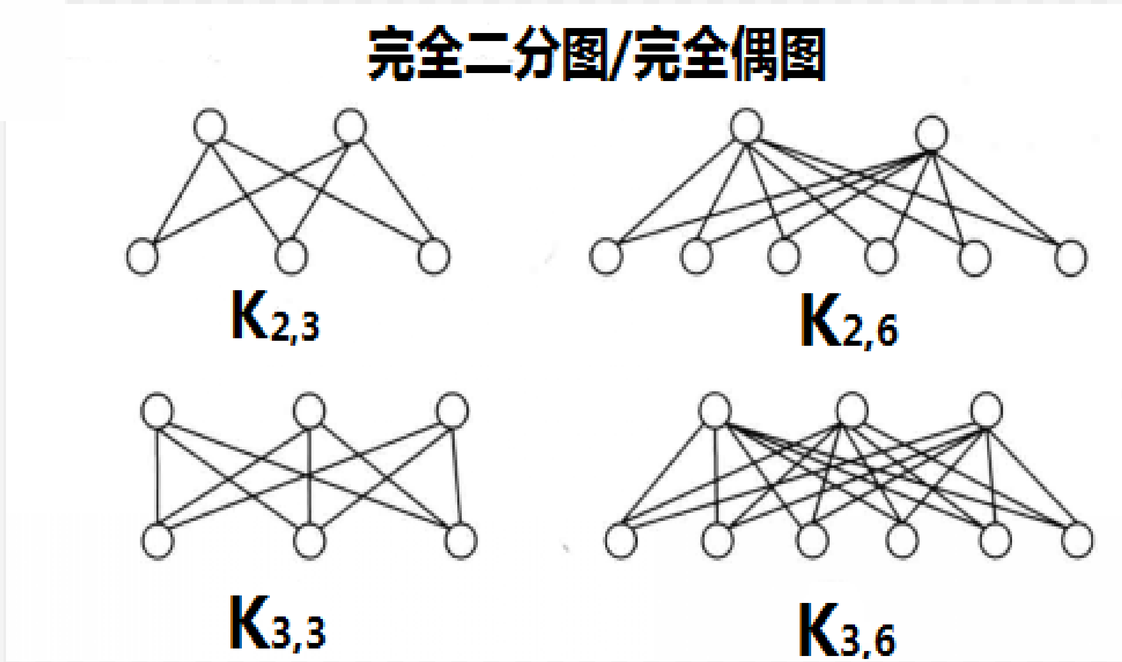
完全图，偶图 一个**简单图**中若任意两点之间均有边相连，称这样的图为完全图。

含有n个顶点的完全图，其边数有 $C_n^2=\frac{1}{2}n(n-1)$条。

如果图的顶点能分成两个互不相交的非空集合V_1和V_2，使在同一集合中任意两个顶点均不相邻，则称这样的图为**偶图（也称二分图）**。如果偶图的顶点集合V_1、V_2之间的每一对不同顶点都有一条边相连，称这样的图为**完全偶图**。完全偶图中V_1含m个顶点，V_2含n个顶点，则其边数共$m\cdot n$条。

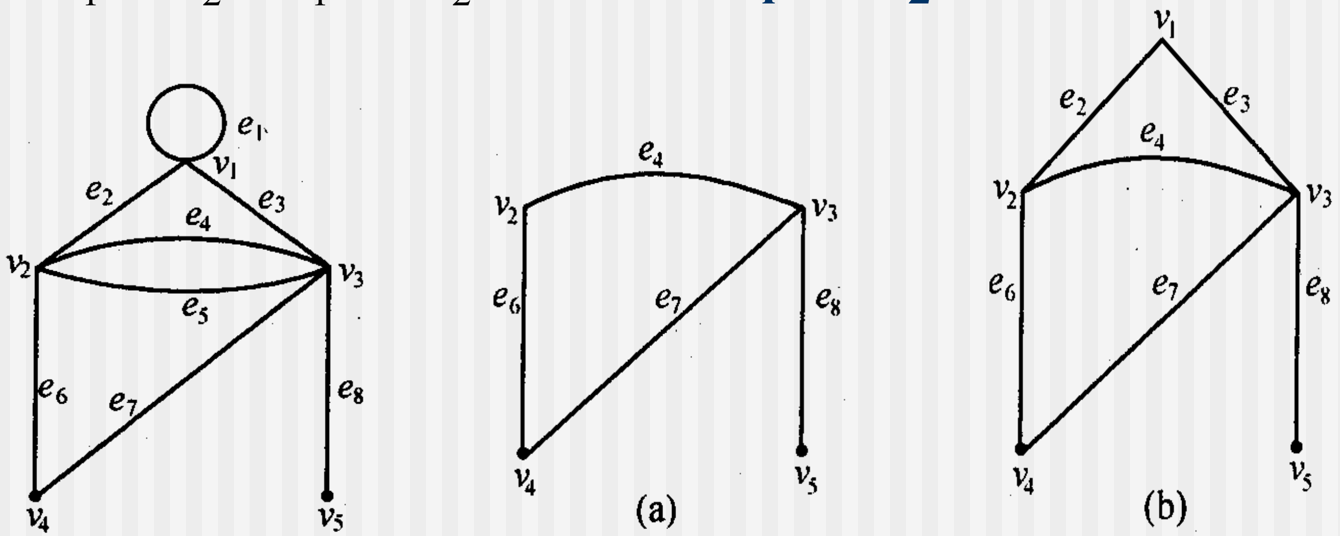
§1．图的基本概念与模型





§1．图的基本概念与模型

子图，部分图　图$G_1=\{V_1，E_1\}$和图$G_2=\{V_2，E_2\}$，如果有$V_1 \subseteq V_2$和$E_1 \subseteq E_2$，称G_1是G_2的一个子图。若有 $V_1 = V_2, E_1 \subset E_2$，则称$G_1$是$G_2$的一个部分图。



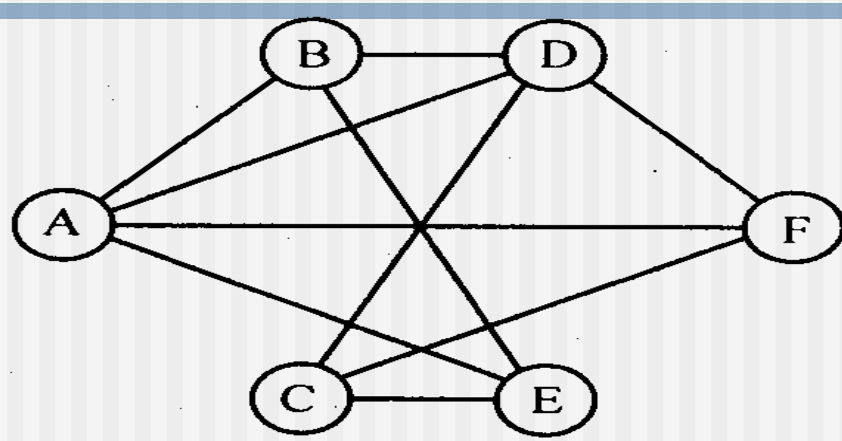
部分图也是子图，但子图不一定是部分图。

§1．图的基本概念与模型

【例1】 有甲、乙、丙、丁、戊、己六名运动员报名参加A、B、C、D、E、F六个项目的比赛。表6-1中打√的是各运动员报名参加的比赛项目。问六个项目的比赛顺序应如何安排，做到每名运动员都不连续地参加两项比赛。

	A	B	C	D	E	F
甲	√			√		√
乙	√	√		√		
丙			√		√	
丁	√				√	
戊	√	√			√	
己			√	√		√

§1．图的基本概念与模型



【解】把比赛项目作为研究对象，用点表示。如果两个项目有**同一名运动员参加，在代表这两个项目的点之间连一条线**。在图中找出一个点的序列，使依次排列的两个点不相邻，即能做到每名运动员不连续地参加两项比赛。

例如A、C、B、F、E、D。

§2．树图和图的最小部分树

树图（简称树，记作T（V，E））： **无圈的连通图。**

2-1 树的性质

性质1 任何树中必存在次为1的点。

称次为1的点为**悬挂点**，与悬挂点关联的边称为**悬挂边**。

如果从树图中拿掉悬挂点及与其关联的悬挂边，余下的点和边构成的图形仍连通且无圈，因此还是一个树图。

§2．树图和图的最小部分树

2-1 树的性质

性质2　具有n个顶点的树的边数恰好为（n - 1）条。

性质3　任何具有n个点、（n - 1）条边的连通图是树图。

§2．树图和图的最小部分树

以上性质说明：

1．树是边数最多的无圈的连通图，在树图上只要任意再加上一条边，必定会出现圈。

2．由于树图是无圈的连通图，即树图的任意两个点之间有一条且仅有一条唯一通路。因此树图也是最脆弱的连通图。只要从树图中取走任一条边，图就不连通。因此一些重要的网络不能按树的结构设计。

§2．树图和图的最小部分树

2-2图的最小部分树

如果G 1是G 2的**部分图**，又是**树图**，则称G 1是G2 的**部分树(或支撑树)**。

设有一个连通图G={V，E}，每一边$e=[v_i, v_j]$有一个非负权 w（e）$=w_{ij}$ （$w_{ij}\geq 0$），如果T＝{V，E＇}是G的一个部分树，称E＇中所有边的权之和为部分树T的权，记为w（T）：

$$w(T)=\sum_{[v_i,v_j]\in T} w_{ij}$$

§2．树图和图的最小部分树

如果部分树T*的权w（T*）是G的所有部分树中权数最小的，则称T*是G的**最小部分树（也称最小支撑树）**，即 $w(T^*) = \min_{T} w(T)$

树图的各条边称为**树枝**，一般图包含有多个部分树，最小部分树是其中**树枝总长最小的部分树**。

§2．树图和图的最小部分树

定理 3　图中任一个点i，若j是与i相邻点中距离最近的，则边[i，j] 一定包含在该图的最小部分树内。

推论　把图的所有点分成V和V1两个集合，则两集合之间连线的最短边一定包含在最小部分树内。

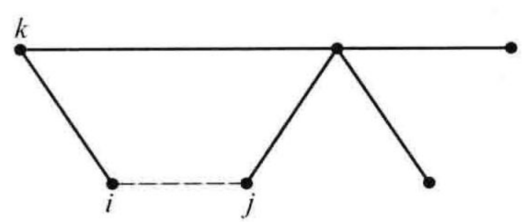


图 6 －4

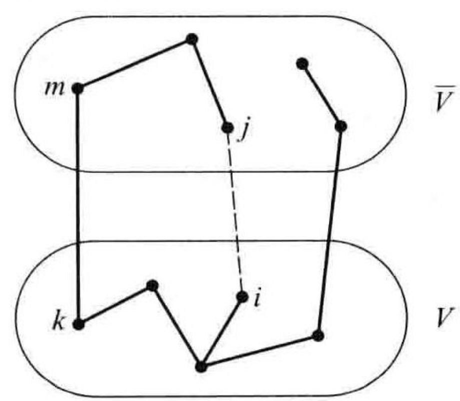
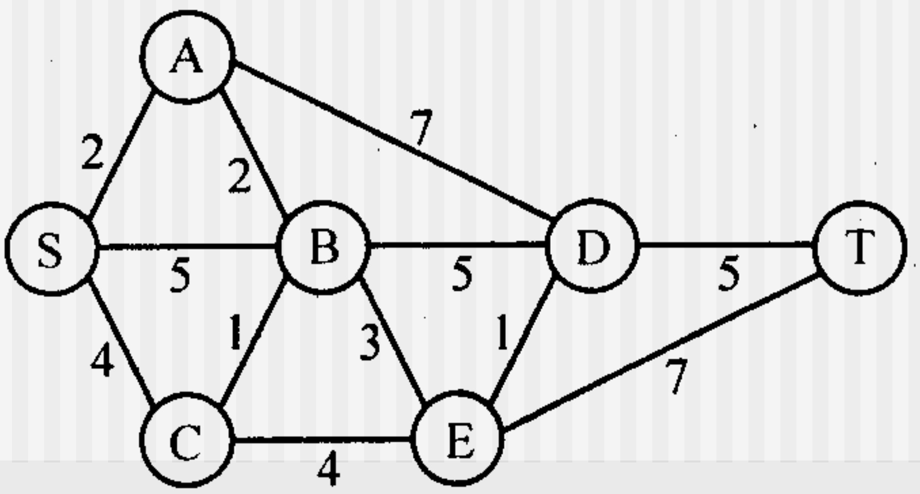


图 6 －5

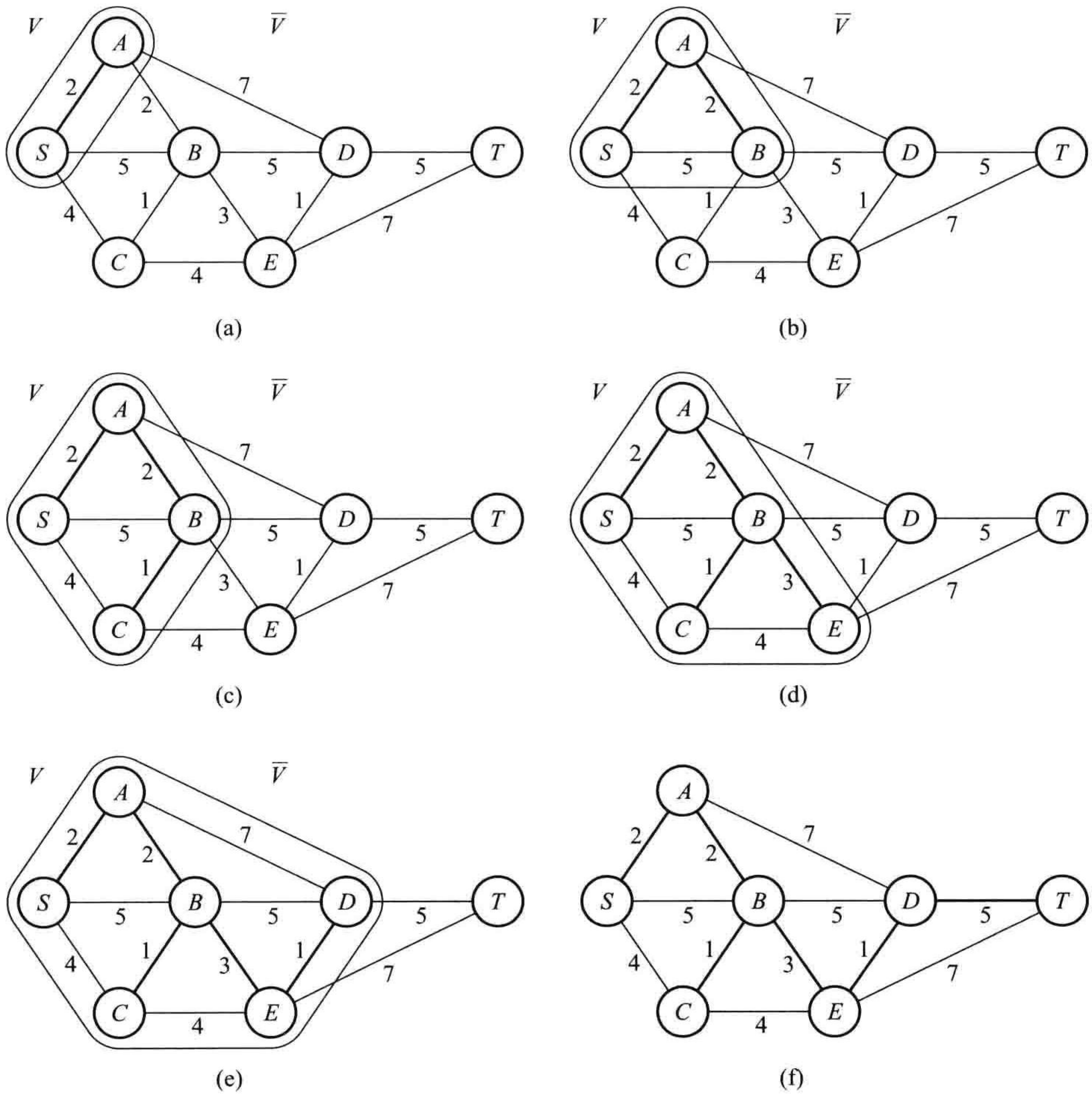
§2．树图和图的最小部分树

2-3 避圈法和破圈法

【例2】如图，S、A、B、C、D、E、T代表村镇，它们间连线表明各村镇间现有道路交通情况，连线旁数字代表道路的长度。现要求沿图中道路架设电线，使上述村镇全部通上电，应如何架设使总的线路长度最短?



§



(a)
(b)
(c)
(d)
(e)
(f)

§2. 树图和图的最小部分树

求最小部分树的两种方法 —— 避圈法和破圈法

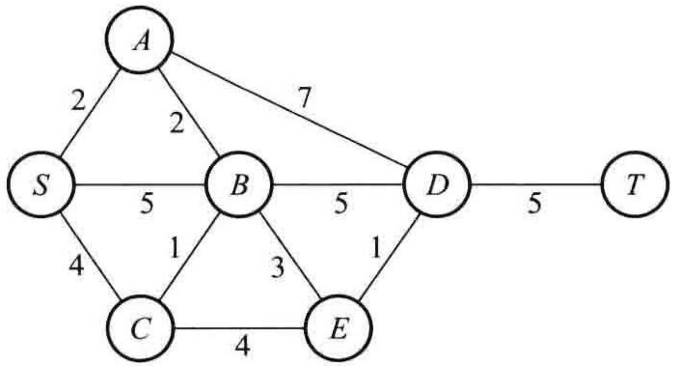
避圈法的步骤：

(1) 从图中任选一点v_i，让$v_i \in V$，图中其余点均包含在 $\overline{V}$ 中；

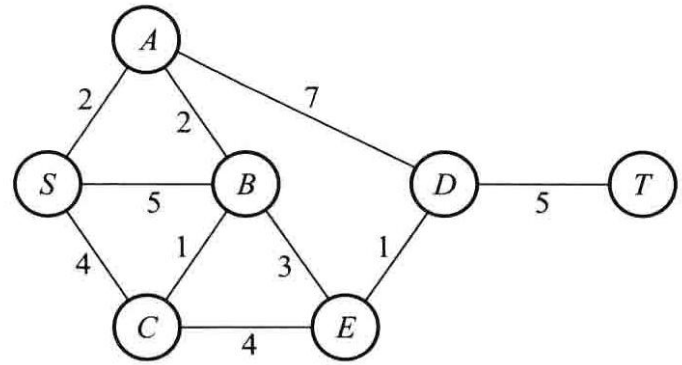
(2) 从V与 $\overline{V}$ 的连线中找出最小边，这条边一定包含在最小部分树内，不妨设最小边为 $[v_i, v_j]$，将$[v_i, v_j]$加粗，以标记是最小部分树内的边；

(3) 令 $V \cup v_j \Rightarrow V, \overline{V} \setminus v_j \Rightarrow \overline{V}$

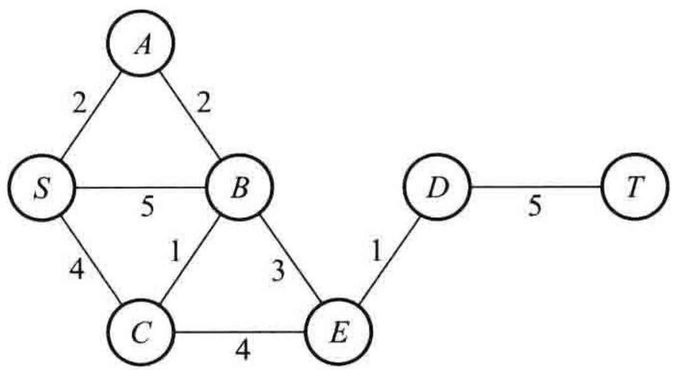
(4) 重复2、3两步，**一直到图中所有点均包含在V中为止。**



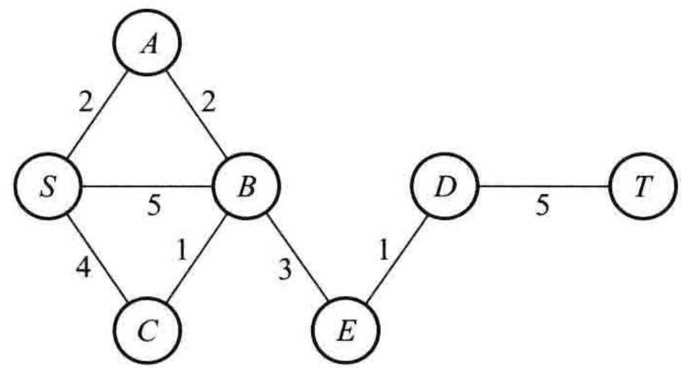
(a) N_1



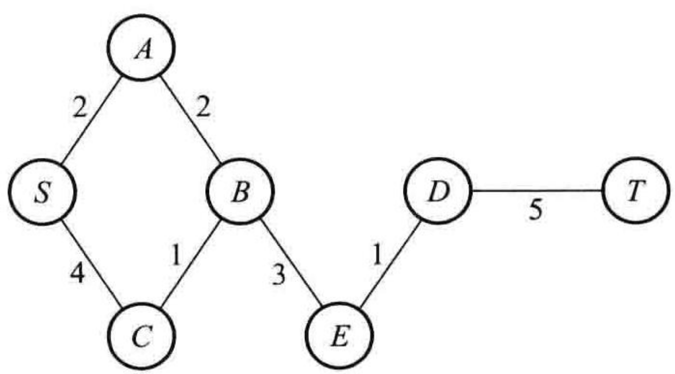
(b) N_2



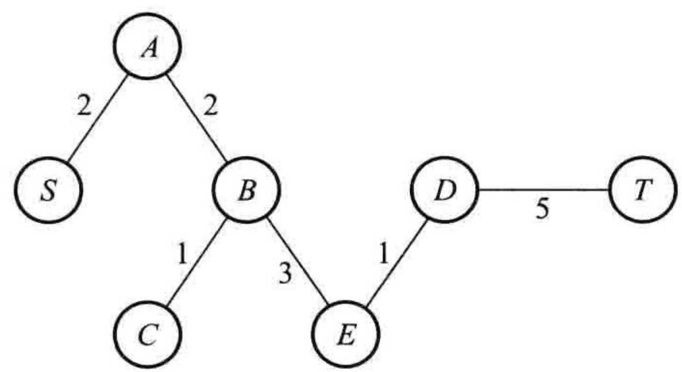
(c) N_3



(d) N_4



(e) N_5



(f) N_6

§2．树图和图的最小部分树

破圈法步骤：从网络图N中任取一回路，去掉这个回路中权数最大的一条边，得一子网络图N_1。在N_1中再任取一回路，再去掉回路中权数最大的一条边，得N_2。如此继续下去，**一直到剩下的子图中不再含回路止**。该子图就是N的最小部分树。

§3．最短路问题

最短路问题：从给定的网络图中找出任意两点之间距离最短的一条路。

距离：权数的代称，在实际的网络中，权数可以是时间、费用等等。

求最短路有两种算法：

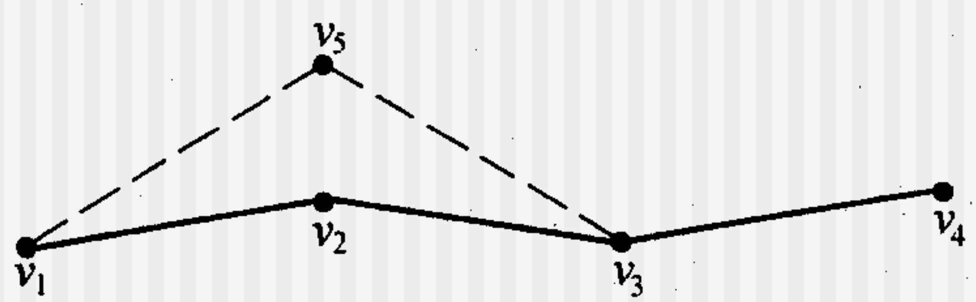
1．求从某一点至其它各点之间最短距离的狄克斯屈拉（Dijkstra）算法

2．求网络图上任意两点之间最短距离的矩阵算法

§3．最短路问题

3-1 狄克斯屈拉（Dijkstra）算法

算法的基本思路：假定v1 → v2→v3→v4是v1→v4的最短路，则v1→v2→v3一定是v1→v3的最短路，v2→v3→v4是v2→v4的最短路。否则，设v1→v3之间的最短路为v1→v5→v3，就有v1→v5→v3→v4小于v1→v2→v3→v4，这与原假设矛盾。



§3．最短路问题

用d_{ij}表示图中两相邻点i 与j 的距离，若i与j不相邻，令$d_{ij}=\infty$，显然$d_{ii}=0$

用 L_{si}表示从s 点到 i 点的最短距离，求从s 点到 t 点的最短路，狄克斯屈拉（Dijkstra）算法步骤：

（1）从点s出发，因$L_{ss}=0$，将此值标注在s旁的小方框内，表示s点已标号；

（2）从s点出发，找出与s相邻的点中距离最小的一个，设为r。将$L_{sr}=L_{ss}+d_{sr}$的值标注在r 旁的小方框内，表明点r 也已标号；

§3．最短路问题

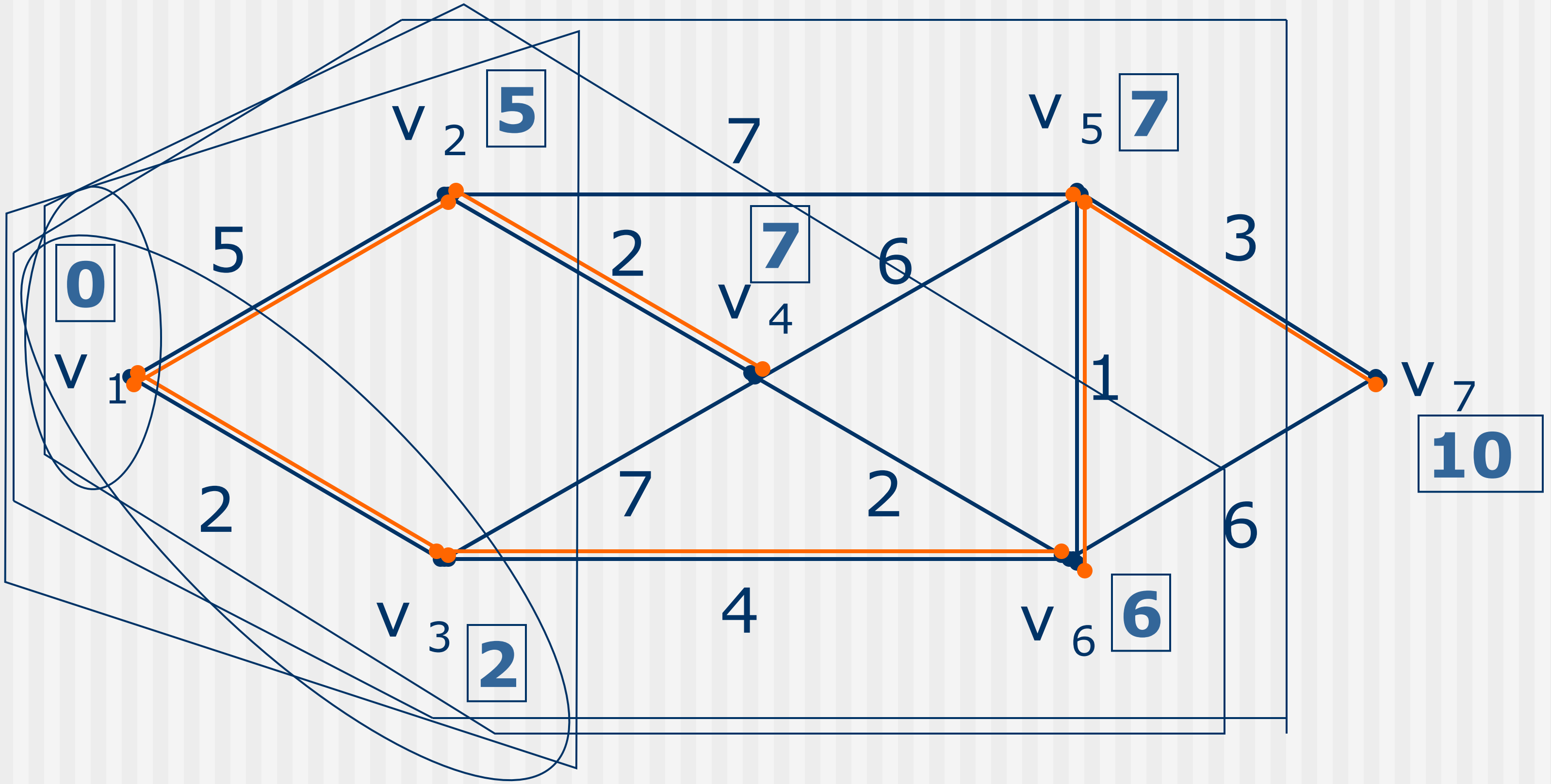
（3）从已标号的点出发，找出与这些点相邻的所有未标号点p。若有

$L_{sp} = \min\{ L_{ss} + d_{sp}, \ L_{sr} + d_{rp} \}$，则对p点标号，并将$L_{sp}$的值标注在p旁的小方框内；

（4）重复第3步，一直到t点得到标号为止。

§3. 最短路问题

【例3】



§3．最短路问题

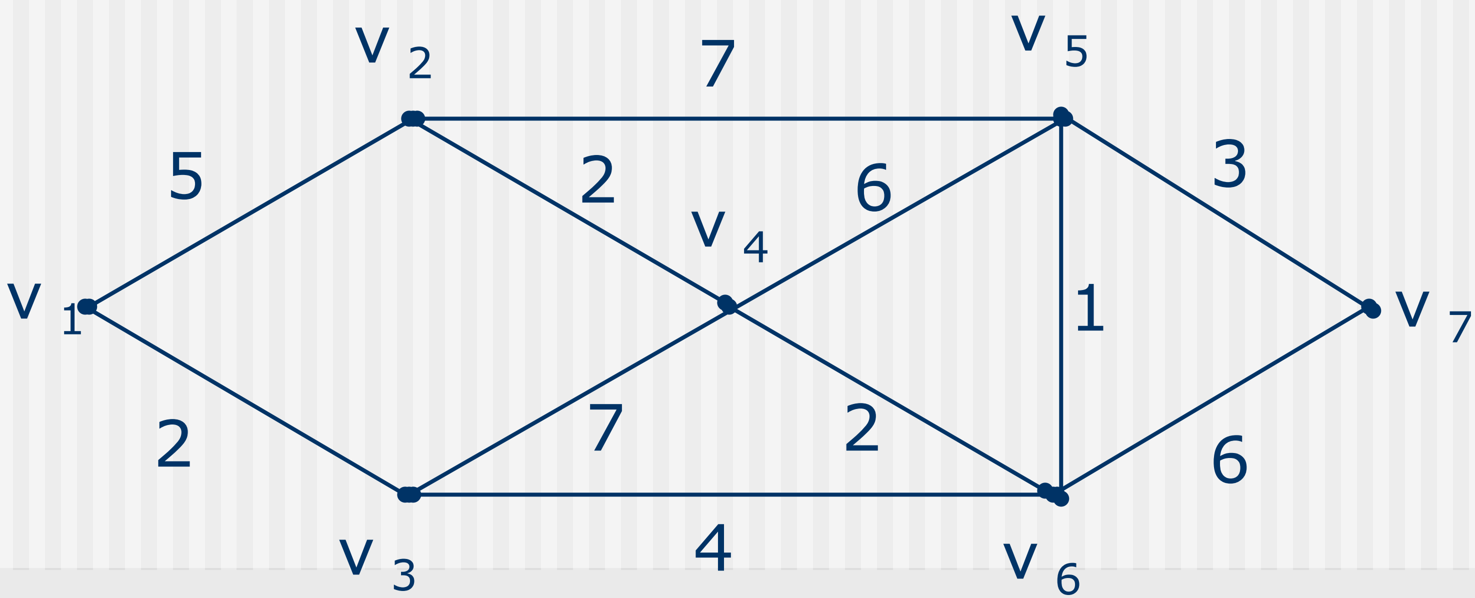
3-2 求任意两点间最短距离的矩阵算法

【例4】 用矩阵算法求各点之间最短距离

用d_{ij}表示图中两相邻点i与j的距离，若i与j不相邻，令$d_{ij}=\infty$，令$d_{ii}=0$

§3. 最短路问题

$$
\mathbf{D}^{(0)}=\begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} & d_{16} & d_{17} \\ d_{21} & d_{22} & d_{23} & d_{24} & d_{25} & d_{26} & d_{27} \\ d_{31} & d_{32} & d_{33} & d_{34} & d_{35} & d_{36} & d_{37} \\ d_{41} & d_{42} & d_{43} & d_{44} & d_{45} & d_{46} & d_{47} \\ d_{51} & d_{52} & d_{53} & d_{54} & d_{55} & d_{56} & d_{57} \\ d_{61} & d_{62} & d_{63} & d_{64} & d_{65} & d_{66} & d_{67} \\ d_{71} & d_{72} & d_{73} & d_{74} & d_{75} & d_{76} & d_{77} \end{bmatrix}=\begin{bmatrix} 0 & 5 & 2 & \infty & \infty & \infty & \infty \\ 5 & 0 & \infty & 2 & 7 & \infty & \infty \\ 2 & \infty & 0 & 7 & \infty & 4 & \infty \\ \infty & 2 & 7 & 0 & 6 & 2 & \infty \\ \infty & 7 & \infty & 6 & 0 & 1 & 3 \\ \infty & \infty & 4 & 2 & 1 & 0 & 6 \\ \infty & \infty & \infty & \infty & 3 & 6 & 0 \end{bmatrix}
$$



§3．最短路问题

D$^{(0)}$表示从i点到j点的直接最短距离，但从i点到j点的最短距离并不一定是直接到达，有可能经过中间点，如果只经过一个中间点，从i点到j点的最短距离为

$$\min\{d_{11}+d_{12}, d_{12}+d_{22}, d_{13}+d_{32}, d_{14}+d_{42}, d_{15}+d_{52}, d_{16}+d_{62}, d_{17}+d_{72}\}$$

即

$$\min\{d_{1r}+d_{r2}\}$$

构造 D$^{(1)}$ $=\{d_{ij}^{(1)}\}$，其中 $d_{ij}^{(1)}=\min\{d_{ir}+d_{rj}\}$，从i点到j点最多只经过一个中间点的最短距离

§3. 最短路问题

构造 D（2）$=\{d_{ij}^{(2)}\}$，其中 $d_{ij}^{(2)}=\min\{d_{ir}^{(1)}+d_{rj}^{(1)}\}$，从i点到j点最多只经过3个中间点的最短距离

… …

构造 D（k）$=\{d_{ij}^{(k)}\}$，其中$d_{ij}^{(k)}=\min\{d_{ir}^{(k-1)}+d_{rj}^{(k-1)}\}$

从i点到j点最多只经过（2 k - 1）个中间点的最短距离

设网络图有p个点，则一般计算不超过D（k）：

$2^{k-1}-1<p-2\le 2^{k}-1$ 即 $k-1<\dfrac{\lg(p-1)}{\lg 2}\le k$

如果出现D（m+1）＝D（m）时，计算也可结束。

§3. 最短路问题

例中，p = 7， $\frac{\lg(p-1)}{\lg 2} = \frac{\lg 6}{\lg 2} \le 2.6$，**k = 3**

$$D^{(1)} = \begin{bmatrix} 0 & 5 & 2 & 7 & 12 & 6 & \infty \\ 5 & 0 & 7 & 2 & 7 & 4 & 10 \\ 2 & 7 & 0 & 6 & 5 & 4 & 10 \\ 7 & 2 & 6 & 0 & 3 & 2 & 8 \\ 12 & 7 & 5 & 3 & 0 & 1 & 3 \\ 6 & 4 & 4 & 2 & 1 & 0 & 4 \\ \infty & 10 & 10 & 8 & 3 & 4 & 0 \end{bmatrix}$$

§3．最短路问题

$$
D^{(2)} = \begin{bmatrix} 0 & 5 & 2 & 7 & 7 & 6 & 10 \\ 5 & 0 & 7 & 2 & 5 & 4 & 8 \\ 2 & 7 & 0 & 6 & 5 & 4 & 8 \\ 7 & 2 & 6 & 0 & 3 & 2 & 6 \\ 7 & 5 & 5 & 3 & 0 & 1 & 3 \\ 6 & 4 & 4 & 2 & 1 & 0 & 4 \\ 10 & 8 & 8 & 0 & 3 & 4 & 0 \end{bmatrix}
$$

$$D^{(3)} = D^{(2)}$$

P127 【例5】

§4. 中国邮路问题

欧拉回路：连通图G中，若存在一条回路，经过每边一次且仅一次，称这条回路为欧拉回路。具有欧拉回路的图称为**欧拉图**。

可以证明：连通图G是欧拉回路的充要条件是图中的点全为偶点。

§4．中国邮路问题

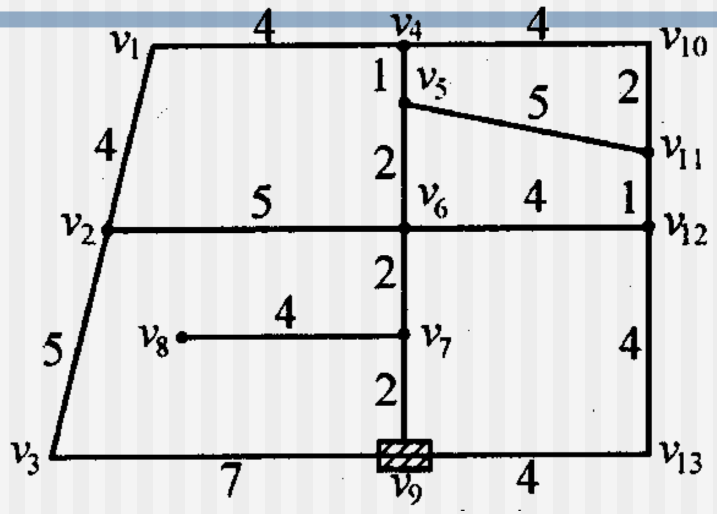
如果G是欧拉图，则欧拉回路就是最短路线。如果G 不是欧拉图，可将G转化为欧拉图：将图中奇点两两相连，变成偶点，则包括连线在内的图构成欧拉图，而连线的长度就是邮递员要重复走的路。

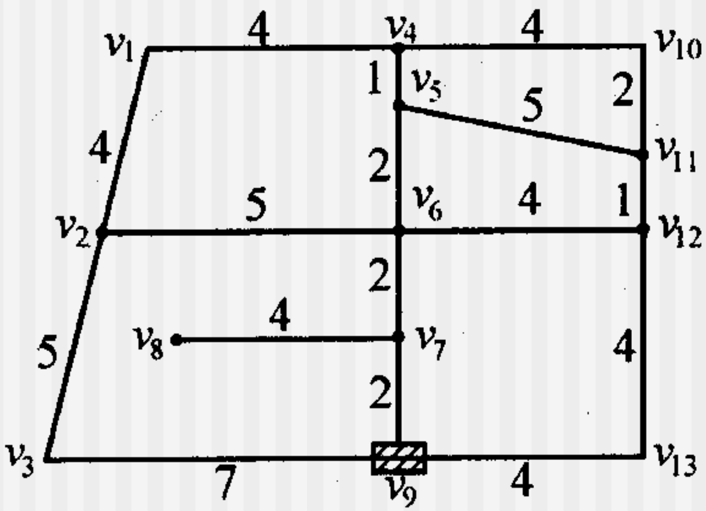
为使重复走的路最短，应使奇点之间的连线最短：

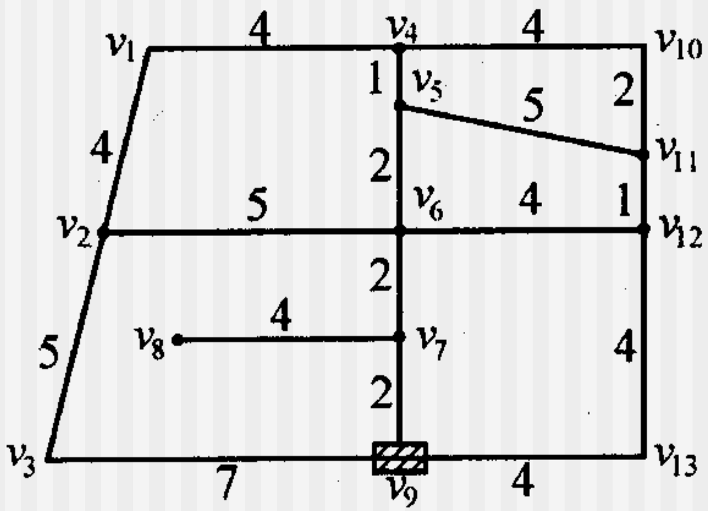
（1）每条边最多重复一次

（2）在图G的每个回路上，重复的边的长度不超过回路总长的一半

§4．中国邮路问题


v_1
4
v_4
4
v_{10}
1
v_5
5
2
4
v_{11}
2
5
v_6
4
1
v_2
v_{12}
2
4
v_8
v_7
5
4
2
v_3
v_{13}
7
v_9
4

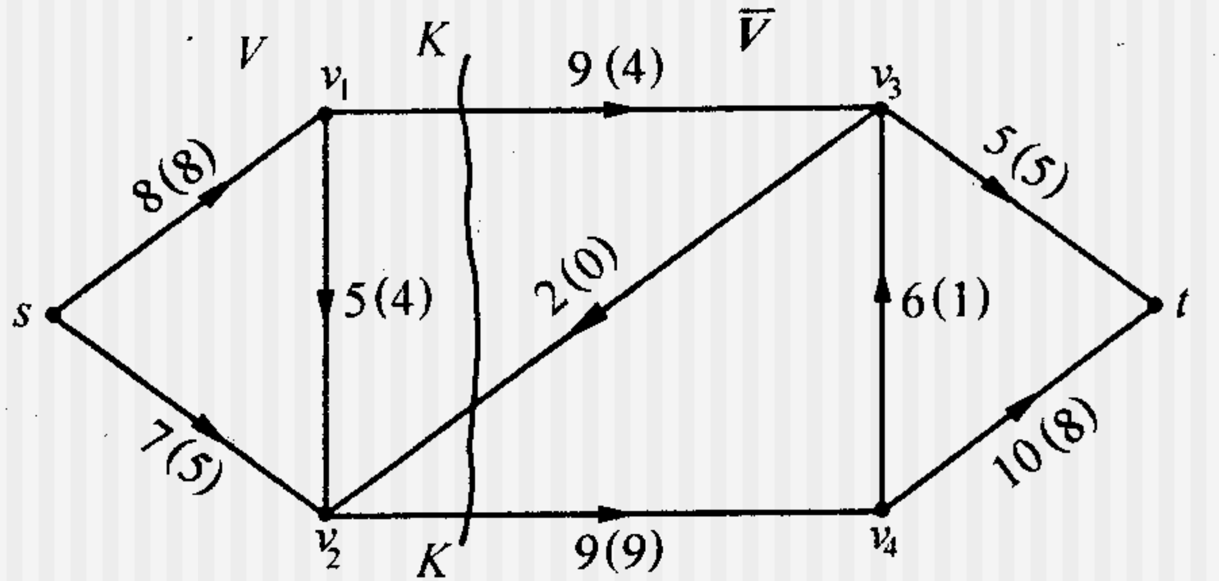

v_1
4
v_4
4
v_{10}
1
v_5
5
2
4
v_{11}
2
5
v_6
4
1
v_2
v_{12}
2
4
v_8
v_7
5
4
2
v_3
v_{13}
7
v_9
4


v_1
4
v_4
4
v_{10}
1
v_5
5
2
4
v_{11}
2
5
v_6
4
1
v_2
v_{12}
2
4
v_8
v_7
5
4
2
v_3
v_{13}
7
v_9
4

§5. 网络的最大流

1. 网络最大流的有关概念
2. 割和流量
3. 最大流和最小割定理
4. 求网络最大流的标号算法
5. 应用举例

§5. 网络的最大流



§5. 网络的最大流

5-1. 网络最大流的有关概念

1. 有向图与容量网络

有向图：图中的连线有规定指向，称作弧，

(v_i, v_j)

有向图是点与弧的集合，记作D（V，A）

容量网络：对网络上每条弧（v_i，v_j）都给出一个最大的通过能力，称为弧的容量，记为

$c(v_i, v_j)$ 或 c_{ij}

§5. 网络的最大流

发点　收点　中间点

在网络图中通常规定一个发点（源点，记为s）和一个收点（汇点，记为t），网络中既非发点又非收点的其他点称为中间点。

网络的最大流：网络中从发点到收点之间允许通过的最大流量。

§5．网络的最大流

2．流与可行流

流　加载在网络中各条弧上的一组负载量。加载在（v_i，v_j）上的负载量记作f（v_i，v_j），简写为f_{ij}

零流　网络上所有$f_{ij}=0$

可行流　在容量网络上满足下面条件的一组流称为可行流：

（1）容量限制条件　对所有弧满足

$$0\le f(v_i, v_j)\le c(v_i, v_j)$$

（2）中间点平衡条件

$$\sum_{(v_i,v_j)\in A} f(v_i,v_j) - \sum_{(v_j,v_i)\in A} f(v_j,v_i) = 0 \quad (i\neq s, t)$$

§5. 网络的最大流

若以v（f）表示网络中从s → t的流量，则应有

$$v(f)=\sum_j f(v_s,v_j)=\sum_j f(v_j,v_t)$$

网络上一定存在可行流（例如零流是可行流）

求网络的最大流：在满足容量限制条件和中间点平衡的条件下，使v（f）值达到最大。

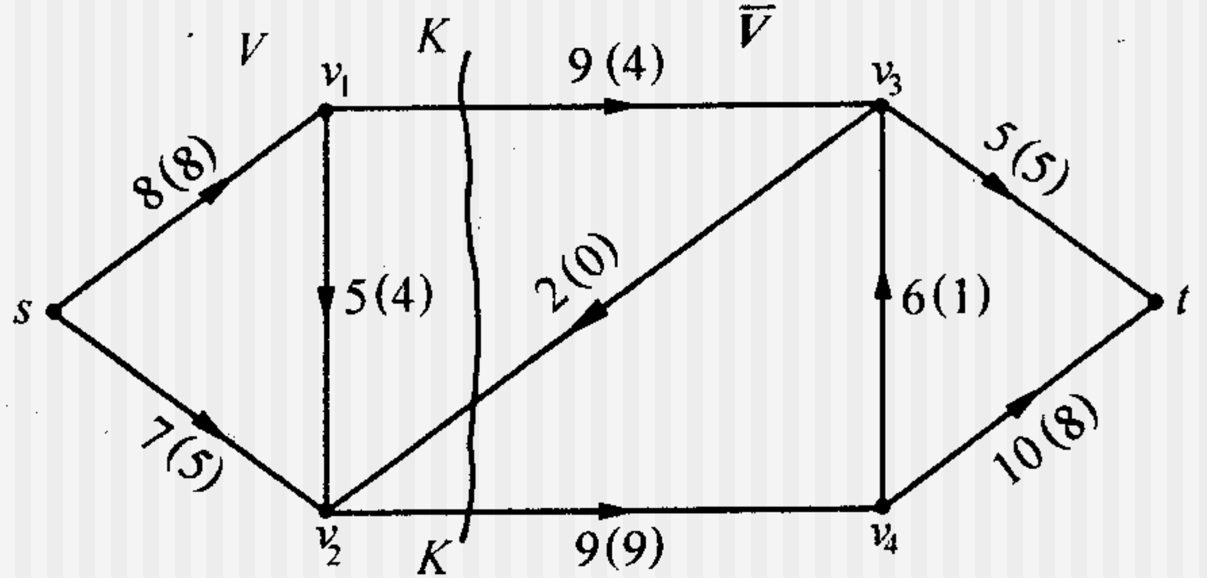
§5．网络的最大流

5-2．割和流量

给定网络D =（V，A），若点集V被分分为两个非空集合 V_1 和 $\overline{V}_1$，使 $v_s \in V_1$，$v_t \in \overline{V}_1$，则把弧集（ V_1， $\overline{V}_1$ ）称为是（分离 v_s 和 v_t 的）割（或称截集）。

从直观上说，**割是从 v_s 到 v_t 的必经之路**，若把割中的弧从网络中去掉，则从 v_s 到 v_t 就不存在路了。

§5. 网络的最大流



例如：$(V_1, \bar{V}_1) = \{(v_1, v_3), (v_2, v_4)\}$ 是一个割

注意：其中不包括 (v_3, v_2)

§5．网络的最大流

割的容量：组成割的集合中各弧的容量之和，用c（V_1，$\overline{V}_1$）表示

$$c(V_1,\overline{V}_1)=\sum_{(i,j)\in(V_1,\overline{V}_1)} c(v_i,v_j)$$

§5. 网络的最大流

例中全部不同的割： P130 表6-3

V	$\bar{V}$	割	割的容量
s	v_1,v_2,v_3,v_4,t	$(s,1)(s,2)$	15
s,v_1	v_2,v_3,v_4,t	$(s,2)(1,2)(1,3)$	21
s,v_2	v_1,v_3,v_4,t	$(s,1)(2,4)$	17
s,v_1,v_2	v_3,v_4,t	$(1,3)(2,4)$	18
s,v_1,v_3	v_2,v_4,t	$(s,2)(1,2)(3,2)(3,t)$	19
s,v_2,v_4	v_1,v_3,t	$(s,1)(4,3)(4,t)$	24
s,v_1,v_2,v_3	v_4,t	$(2,4)(3,t)$	14
s,v_1,v_2,v_4	v_3,t	$(1,3)(4,3)(4,t)$	25
s,v_1,v_2,v_3,v_4	t	$(3,t)(4,t)$	15

§5．网络的最大流

5-3．最大流和最小割

用f （V_1, $\overline{V}_1$）表示割（V_1, $\overline{V}_1$）中 V_1→ $\overline{V}_1$方向弧的流量的总和,

f （$\overline{V}_1$, V_1）表示割中$\overline{V}_1$→V_1方向弧流量的总和:

$$f(V_1, \overline{V}_1) = \sum_{(i,j)\in(V_1,\overline{V}_1)} f(v_i, v_j)$$

$$f(\overline{V}_1, V_1) = \sum_{(j,i)\in(\overline{V}_1,V_1)} f(v_j, v_i)$$

§5. 网络的最大流

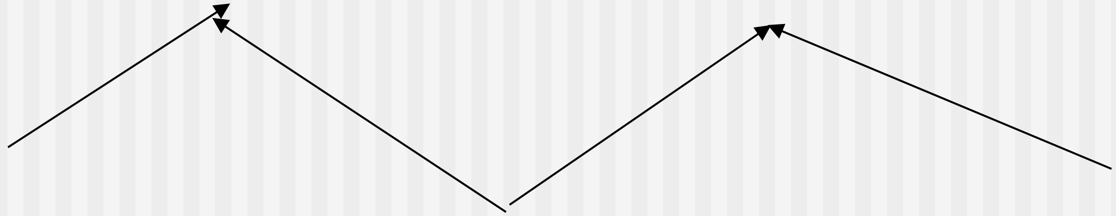
则s→t 的流量v（f）等于通过割的从 $V_1 \to \overline{V_1}$ 的流量减去 $\overline{V_1} \to V_1$ 的流量：

$$v(f) = f(V_1, \overline{V_1}) - f(\overline{V_1}, V_1)$$

§5．网络的最大流

5-3．最大流和最小割定理

前向弧与后向弧 设μ是网络中从v_s到v_t的一条链，定义链的方向是从v_s到v_t，则链上的弧可分为两类：一类是弧的方向与链的方向一致，称为前向弧，所有前向弧记为μ^+；另一类是弧与链的方向相反，称为后向弧，所有后向弧记为μ^-。



§5．网络的最大流

增广链　设f是一个可行流，μ是从v_s到v_t的一条链，若μ满足下列条件，则称之为（关于可行流f的）一条增广链：

（1）在弧（v_i，v_j）$\in \mu^+$上，有$0 \le f_{ij} < c_{ij}$

（2）在弧（v_i，v_j）$\in \mu^-$上，有$0 < f_{ij} \le c_{ij}$

当存在增广链时，可对可行流进行调整，令

$$\theta = \min \begin{cases} (c_{ij} - f_{ij}) & ,\quad 对\mu^+ \\ f_{ij} & ,\quad 对\mu^- \end{cases}$$

则$\theta > 0$

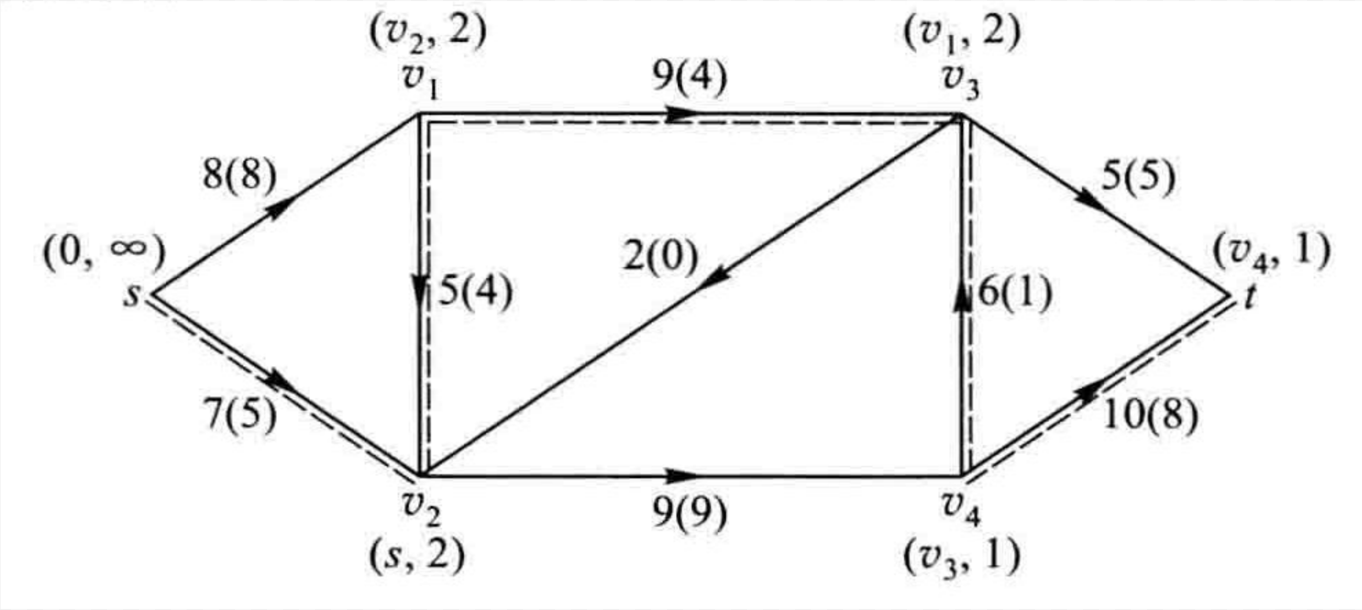
§5. 网络的最大流

再令 $$f_{ij}^{'}=\begin{cases}f_{ij}+\theta & \text{对所有}\mu^{+}\\ f_{ij}-\theta & \text{对所有}\mu^{-}\\ f_{ij} & \text{对非增广链上的弧}\end{cases}$$

则调整之后仍为可行流，但比原来的可行流流量增大了θ（θ > 0）

只有网络中找不到增广链时，s→t 的流才不可能进一步增大。

§5. 网络的最大流



§5．网络的最大流

定理4 在网络中s→t 的最大流量等于它的最小割集的容量，即

$$v^*(f) = c^*(V_1, \overline{V}_1)$$

§5．网络的最大流

5-4．求网络最大流的标号算法

该算法由Ford和Fulkerson于1956年提出，故又称Ford-Fulkerson标号算法。其实质是判断有否增广链存在，并设法把增广链找出来。算法的步骤如下：

第1步：首先给发点s标号（0,ε（s））。括弧中第二个数字是使这个点得到标号的前一个点的代号，因s是发点，故记为0。括弧中第二个数字ε（s）表示**从上一标号点到这个标号点的流量的最大允许调整值**。s为发点，不限允许调整量，故ε（s）＝∞。

§5．网络的最大流

第2步：列出与已标号点相邻的所有未标号点：

（1）考虑从标号点i出发的弧（i，j），如果有$f_{ij}=c_{ij}$，不给点j标号；若有$f_{ij}<c_{ij}$，则对点j标号，记为（i，ε（j））。括弧中的i表示点j的标号是从点i延伸过来的，ε（j）=min{ε（i），（$c_{ij}-f_{ij}$）}；

（2）考虑所有指向标号点i的弧（h，i），如果有$f_{hi}=0$，对h点不标号；若有$f_{hi}>0$，则对点h标号，记为（i，ε（h）），其中ε（h）= min{ε（i），f_{hi})}；

§5．网络的最大流

（3）如果某未标号点k**有两个以上相邻的标号点**，为减少迭代次数，可按（1）、（2）中所述规则分别计算出ε（k）的值，并取其中最大的一个标记。

第3步：重复第2步，可能出现两种结果：

（1）标号过程中断，t得不到标号，说明该网络中不存在增广链，给定的流量即为最大流。记**已标号点的集合为V，未标号点集合为** $\bar{V}$，$(V, \bar{V})$ 为网络的最小割；

（2）t得到标号，这时可用**反向追踪法**在网络中找出一条从s→t的由标号点及相应的弧连接而成的增广链。

§5．网络的最大流

第4步：修改流量。设图中原有可行流为f，令

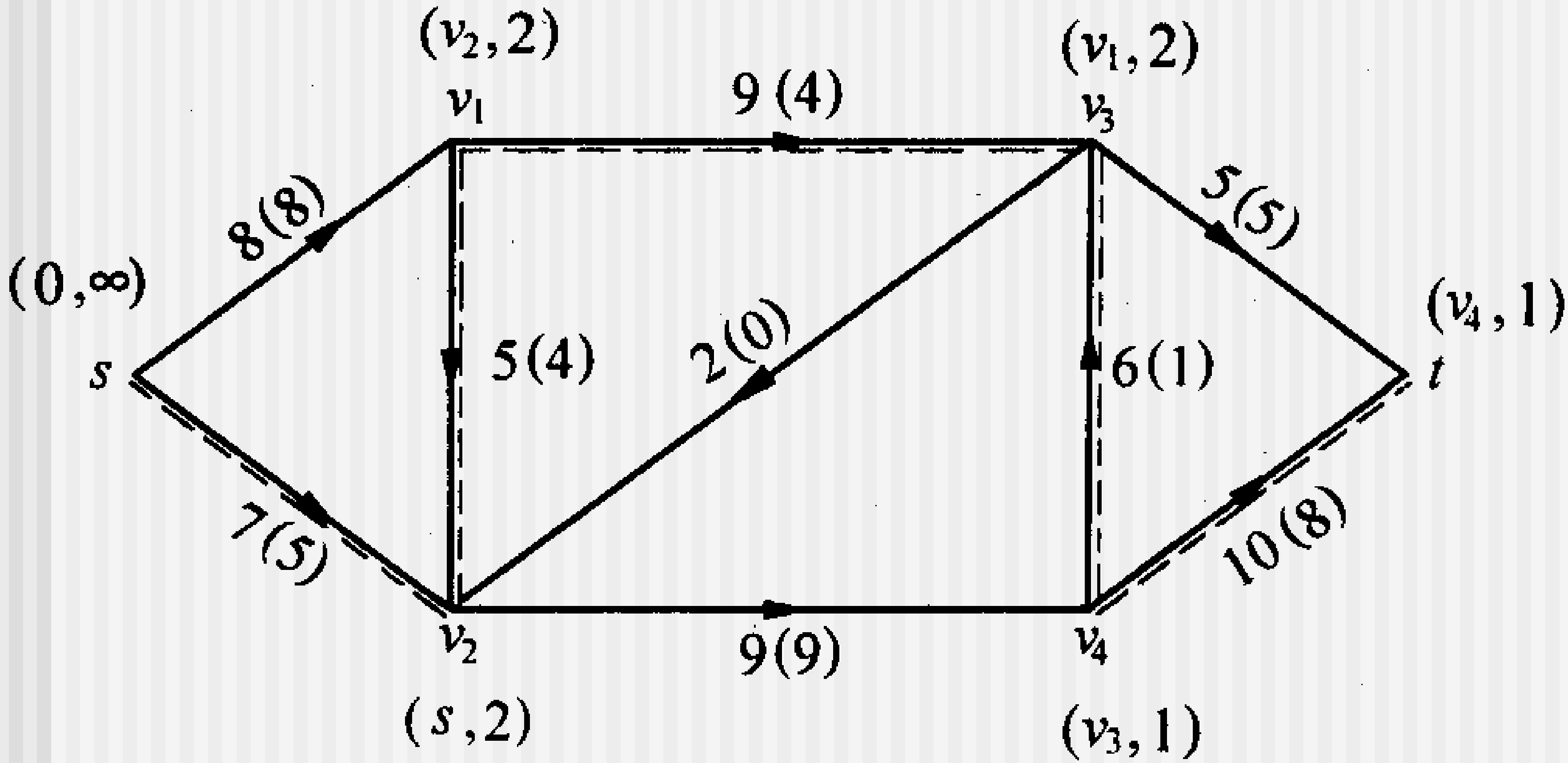
$$f_{ij}^{'} = \begin{cases} f_{ij} + \varepsilon(t) & \text{对增广链上所有 } \mu^{+} \\ f_{ij} - \varepsilon(t) & \text{对增广链上所有 } \mu^{-} \\ f_{ij} & \text{对非增广链上的弧} \end{cases}$$

这样又得到网络上的一个新的可行流f ＇。

第5步：抹掉图上所有标号，重复第1到4步，直至图中找不到任何增广链，即出现第3步的结果（1）为止，这时网络图中的流量即为最大流。

§5．网络的最大流

【例7】



§5. 网络的最大流

$$f'_{s2} = f_{s2} + \varepsilon(t) = 5 + 1 = 6$$

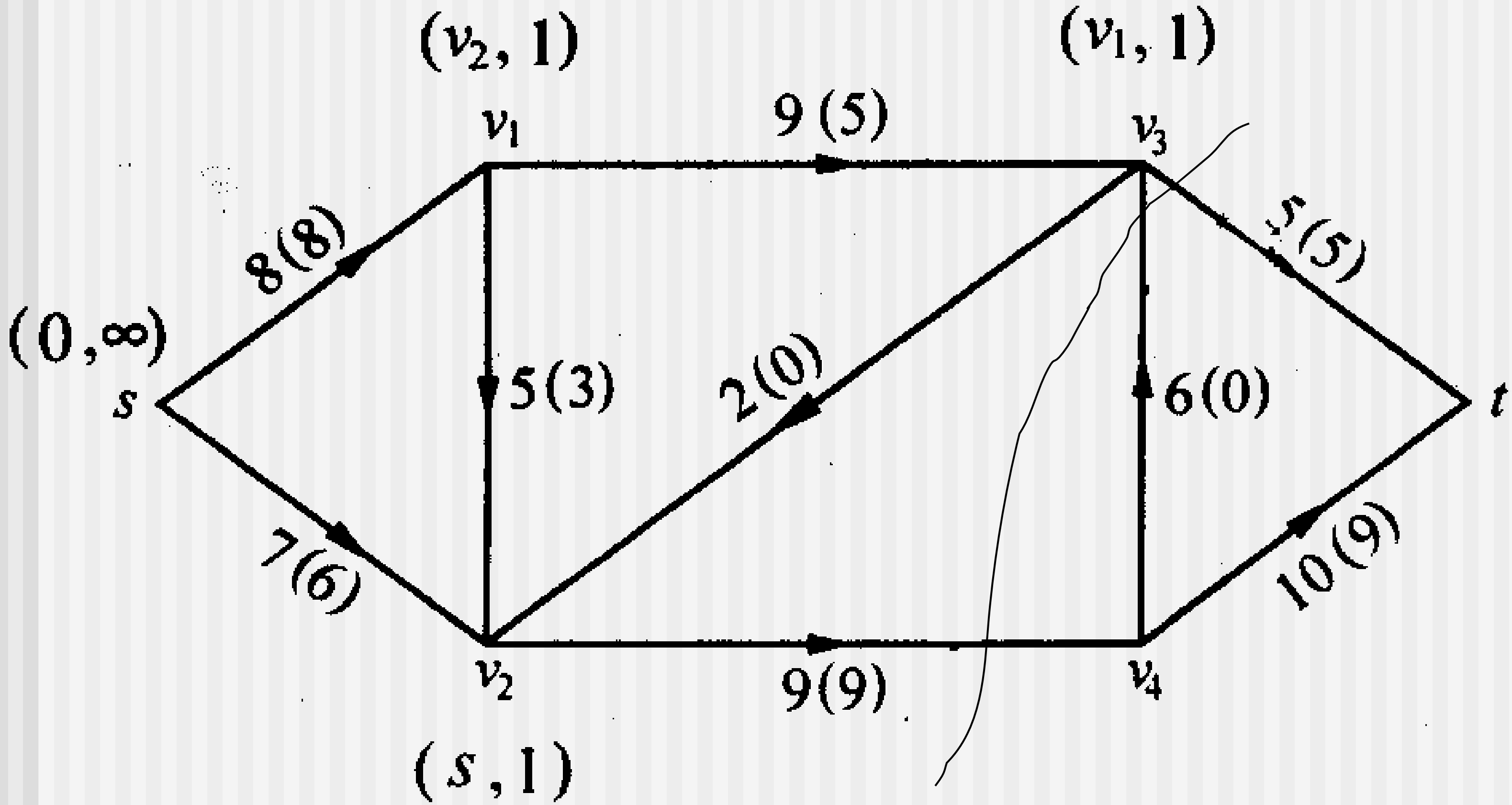
$$f'_{12} = f_{12} - \varepsilon(t) = 4 - 1 = 3$$

$$f'_{13} = f_{13} + \varepsilon(t) = 4 + 1 = 5$$

$$f'_{43} = f_{43} - \varepsilon(t) = 1 - 1 = 0$$

$$f'_{4t} = f_{4t} + \varepsilon(t) = 8 + 1 = 9$$

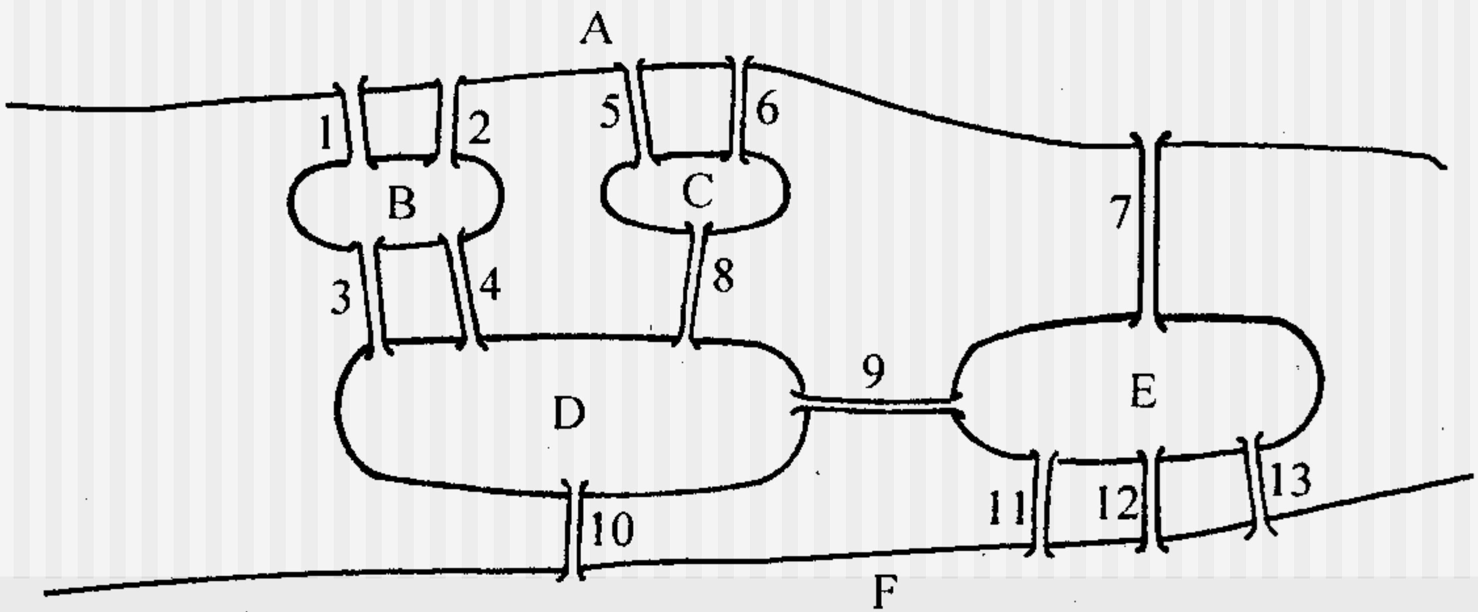
§5. 网络的最大流



§5．网络的最大流

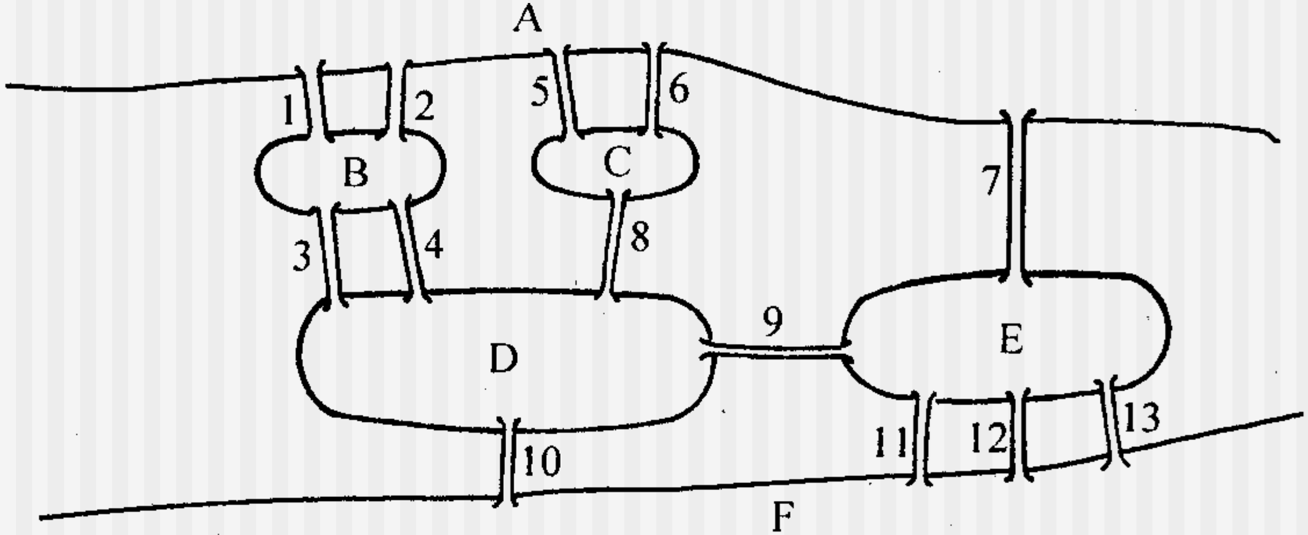
5-5．应用举例

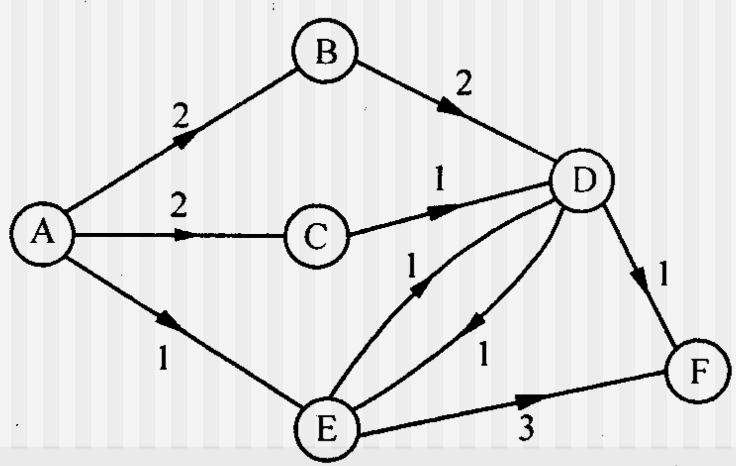
【例8】某河流中有几个岛屿，从两岸至各岛屿及各岛屿之间的桥梁编号如图所示。在一次敌对的军事行动中，问至少应炸断几座及哪几座桥梁，才能完全切断两岸的交通联系？



§5．网络的最大流

【解】将两岸及岛屿用点表示，相互间有桥梁联系的用线表示



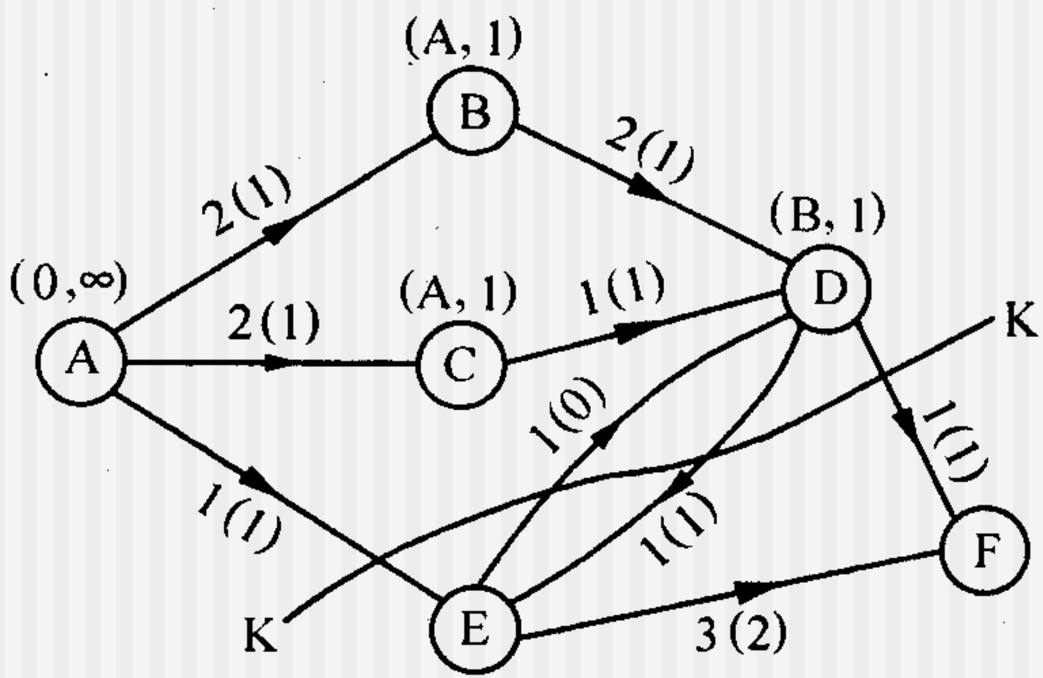


§5. 网络的最大流

图中连线方向根据从A出发通向F的方向来定。因如A→F方向不通的话，从F→A的方向也走不通。其中**D、E之间可能从D→E，也可能从E→D，故画相对方向的两条线**。各弧旁数字为两点间的桥梁数，相当于容量。要求切断A、F间交通联系的最少桥梁数，就相当于网络的最小割。

因此可以先在图中**任意给出一个可行流**，用标号算法求出网络的最大流。

§5. 网络的最大流



由图得到该网络的最小割为{（D，F）（D，E）（A，E）}，即至少应炸断编号为7、9、10的三座桥梁，才能完全切断两岸的交通联系。