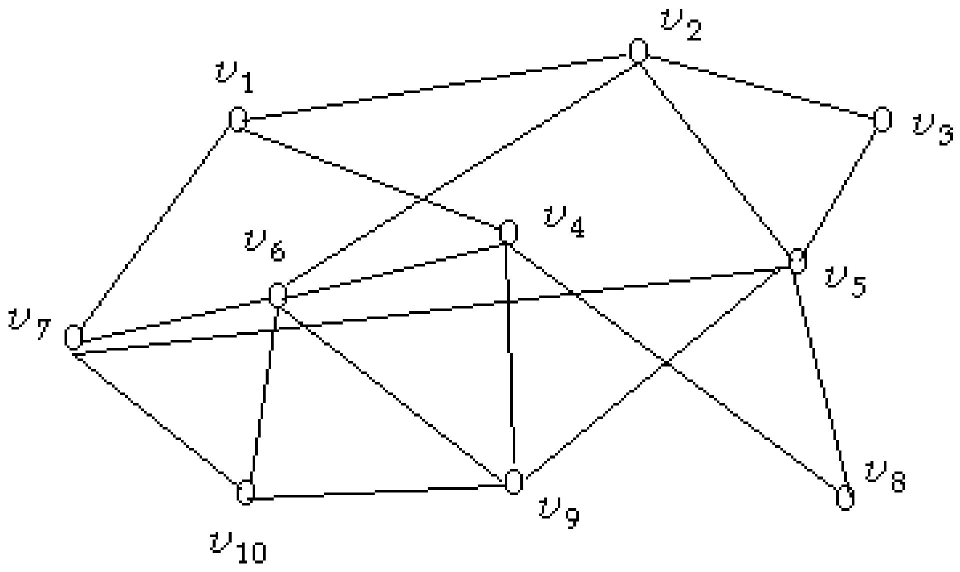
7-7 树与生成树

树与生成树

- 树是图论中的一个重要概念。 早在1847年克希霍夫就用树的理论来研究电网络， 1857年凯莱在计算有机化学中 C_2H_{2n+2}的同分异构物数目时也用到了树的理论。而树在计算机科学中应用更为广泛。
- 我们从一个问题谈起:通讯线路图。

通讯线路图

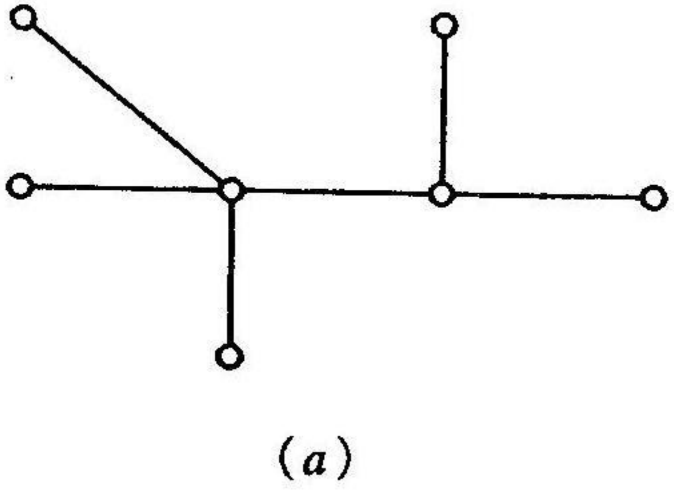
其中v_1，$v_2,…,v_{10}$是十个城市，线路只能在这里相接。不难发现，只要破坏了几条线路，立即使这个通讯系统分解成不相连的两部分。但要问在什么情况下这十个城市依然保持相通？不难知道，至少要有九条线把这十个城市连接在一起，显然这九条线是不存在任何回路的，因而九条线少一条就会使系统失去连通性。

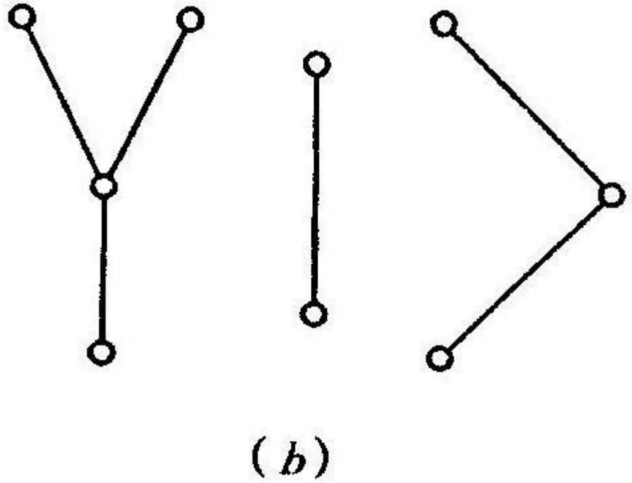


树、森林

- 一个连通且无回路的无向图称为树。
- 在树中度数为1的结点称为树叶,
- 度数大于1的结点称为分枝点或内点。
- 如果一个无回路的无向图的每一个连通分图是树，称为森林。

树、森林

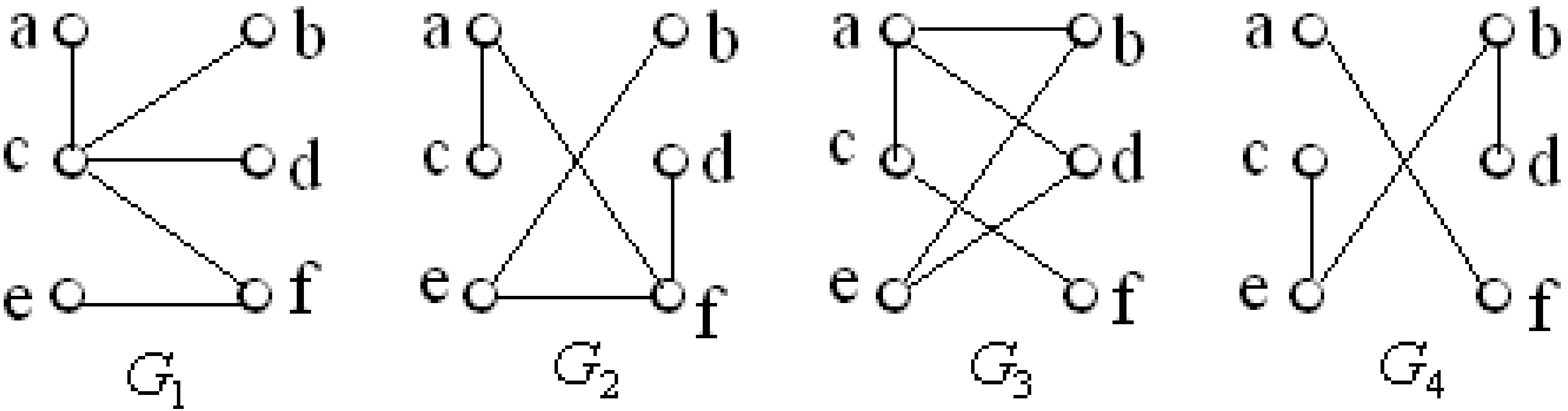




树和森林示意图

树的判断

判断下面各图是否为树？



树的等价定义

定理7-7.1 给定图T，以下关于树的定义是等价的：

(1)无回路的连通图；

(2)无回路且$e=v-1$，其中e为边数，v为结点数；

(3)连通且$e=v-1$；

(4)无回路且增加一条新边，得到一个且仅一个回路；

(5)连通且删去任何一个边后不连通；

(6)每一对结点之间有一条且仅一条路。

树的等价定义证明

(1)无回路的连通图；

(2)无回路且$e=v-1$，其中e为边数，v为结点数；

证明 (1)⇒(2)

设在图T中，当$v=2$时，连通无向图，T中的边数$e=1$，因此$e=v-1$成立。

设$v=k-1$时命题成立，当$v=k$时，因无向图且连通，故至少有一条边其一个端点u的度数为1。设该边为(u,w)，删去结点u，便得到一个$k-1$个结点的连通无向图T'，由归纳假设，图T'的边数$e'=v'-1=(k-1)-1=k-2$，于是再将结点u和关联边(u,w)加到图T'中得到原图T，此时图T的边数为$e=e'+1=(k-2)+1=k-1$，结点数$v=v'+1=(k-1)+1=k$，故$e=v-1$成立。

树的等价定义证明

(2)无回路且$e=v-1$，其中e为边数，v为结点数；

(3)连通且$e=v-1$；

(2)⇒(3)

若T不连通，并且有k(k≥2)个连通分支T_1，T_2，…，T_k，因为每个分图是连通无回路，则我们可证：如T_i有v_i个结点$v_i<v$时，T_i有v_i-1条边，而

$v=v_1+v_2+\ldots+v_k$

$e=(v_1-1)+(v_2-1)+\ldots+(v_k-1)=v-k$

但$e=v-1$，故$k=1$，这与假设G是不连通即$k\geq2$相矛盾。

树的等价定义证明

(3)连通且$e=v-1$；

(4)无回路且增加一条新边，得到一个且仅一个回路；

(3)⇒(4)

若T连通且有$v-1$条边。

当$v=2$时，$e=v-1=1$，故T必无回路。如增加一条边得到且仅得到一个回路。

设$v=k-1$时命题成立。

考察$v=k$时的情况，因为T是连通的，$e=v-1$。故每个结点u有$deg(u)\geq 1$，可以证明至少有一结点u_0，使$deg(u_0)=1$，若不然，即所有结点u有$deg(u) \geq 2$，则$2e\geq 2v$，即$e\geq v$与假设$e=v-1$矛盾。删去u_0及其关联的边，而得到图T'，由归纳假设得知T'无回路，在T'中加入u_0及其关联边又得到T，故T无回路的，如在T中增加一条边(u_i ,u_j)，则该边与T中u_i到u_j的路构成一个回路，则该回路必是唯一的，否则若删除这条新边，T必有回路，得出矛盾。

树的等价定义证明

(4)无回路且增加一条新边，得到一个且仅一个回路；

(5)连通且删去任何一个边后不连通；

(6)每一对结点之间有一条且仅一条路。

(4)⇒(5)

若图T不连通，则存在结点u_i与u_j，u_i与u_j之间没有路，显然若加边{ u_i, u_j }不会产生回路，与假设矛盾。又由于*T*无回路，故删去任一边，图就不连通。

(5)⇒(6)

由连通性可知，任两个结点间有一条路，若存在两点，在它们之间有多于一条的路，则T中必有回路，删去该回路上任一条边，图仍是连通的，与(5)矛盾。

树的等价定义证明

(1)无回路的连通图；

(6)每一对结点之间有一条且仅一条路。

(6)⇒(1)

任意两点间必有唯一一条路，则T必连通，若有回路，则回路上任两点间有两条路，与(6)矛盾。

定理

定理7-7.2 任一棵树至少有两片树叶。

证明 设树*T*=〈*V*，*E*〉，|V|=*v*，

则$\sum deg(v_i)=2(v-1)$

因为*T*是连通图，对于任意$v_i \in T$，

有$deg(v_i) \geq 1$

若*T*中每一个结点的度数大于等于2，

则$\sum deg(v_i) \geq 2v$，得出矛盾。

若T中只有一个结点度数为1，其它结点的度数大于等于2，则

$\sum deg(v_i) \geq 2(v-1)+1=2v-1$，得出矛盾。

故*T*至少有两个结点度数为1。

练习

T是一棵树,有两个2度结点，一个3度结点，三个4度结点，T有几片树叶？

解： 设树T有x片树叶，则T的结点数

$$n=2+1+3+x$$

T的边数

$$m=n-1=5+x$$

又由

$$2m=\sum_{i=1}^{n}\deg(v_i)$$

得

$$2\cdot(5+x)=2\cdot 2+3\cdot 1+4\cdot 3+x$$

所以$x=9$，即树T有9片树叶。

练习

已知无向树T有5片树叶，2度和3度顶点各一个，其余顶点的度数均为4，求4度顶点的个数？

答案：1个（解法略）

生成树、树枝

定义7-7.2 生成树、树枝

若图*G*的生成子图是一棵树，则该树称为*G*的**生成树**。

设图*G*有一棵生成树*T*，则*T*中的边称作**树枝**。

图*G*中不在生成树上的边称为**弦**。所有弦的集合称为**生成树*T*相对于*G*的补**。

生成树

图7-7.3中，可以看出该图的生成树*T*为粗线所表达。其中***e*1,*e*7,*e*5,*e*8,*e*3**都是*T*的树枝，***e*2,*e*4,*e*6**是*T*的弦，{ ***e*2,*e*4,*e*6**}是生成树*T*的补。

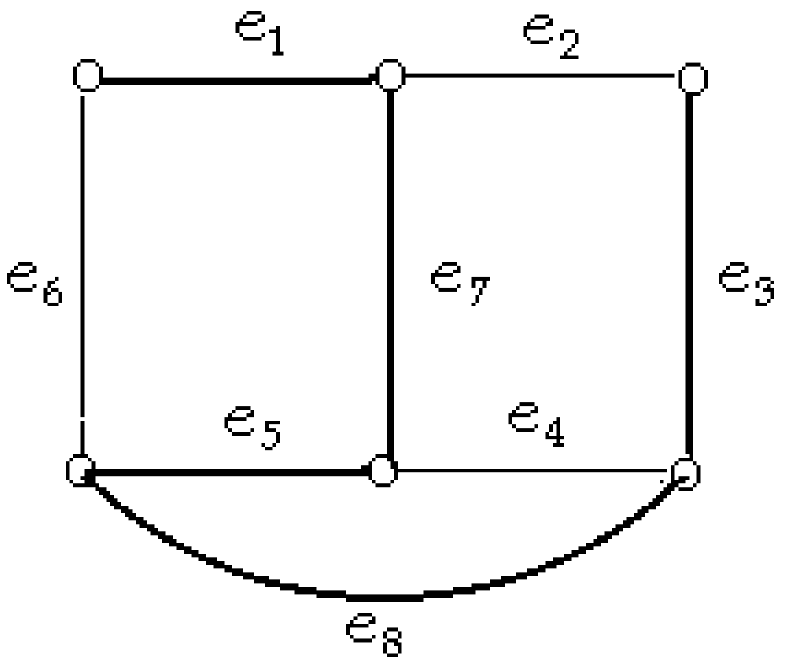


图7-7.3 生成树

定理

定理7-7.3 连通图至少有一棵生成树。

证明

设连通图G没有回路，则它本身就是一棵生成树。若G至少有一个回路，我们删去回路上的一条边，得到G_1，它仍然是连通的，并与G有相同的结点集。若G_1没有回路，则G_1就是G的生成树。若G_1仍然有回路，再删去G_1回路上的一条边，重复上面的步骤，直到得到一个连通图H，它没有回路，但与G有相同的结点集，因此H为G的生成树。

由定理7-7.3的证明过程中可以看出，一个连通图有许多生成树。因为取定一个回路后，就可以从中去掉任何一条边，去掉的边不一样，故可以得到不同的生成树。一般的，图的生成树不唯一。

生成树

例如图7-7.4(a)中，相继删去边2、3和5，就得到生成树T_1，如图7-7.4(b)，若相继删去2、4和6，可得生成树T_2，如图7-7.4 (c)。

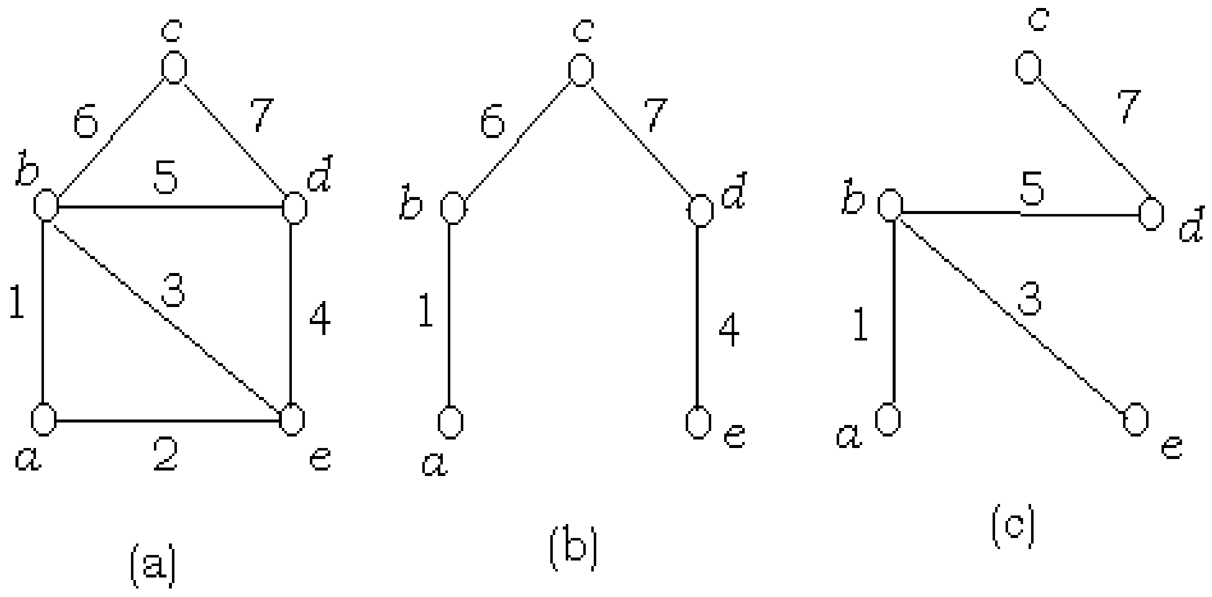


图7-7.4 生成树

带权的生成树

下面我们讨论带权的生成树。

设图G中的一个结点表示一些城市，各边表示城市间道路的连接情况，边的权表示道路的长度，如果我们要用通讯线路把这些城市连接起来，要求沿道路架设线路时，所用的线路最短，这就是要求一棵生成树，使该生成树是图G的所有生成树中边权的和为最小。

边e的权、最小生成树

定义：

假定图*G*是具有*n*个结点的连通图。对应于*G*的每一条边*e*，指定一个正数*C*(*e*)，把*C*(*e*)称作边e的权，(可以是长度、运输量、费用等）。

*G*的生成树也具有一个树权*C*(*T*)，它是*T*的所有边权的和。

在带权的图*G*的所有生成树中，树权最小的那棵生成树，称作最小生成树。

Kruskal算法

Kruskal算法是一种用来寻找最小生成树的算法，由Joseph Kruskal在1956年发表。此方法又称为“避圈法”。其要点是，在与已选取的边不成圈的边中选取最小者。

定理7-7.6(Kruskal，贪心算法) 设图G有n个结点，以下算法产生最小生成树。

(1)选择最小权边e_1,置边数$i \leftarrow 1$；

(2)$i=n-1$结束，否则转(3)；

(3)设定已选定e_1，e_2，…，e_i，在G中选取不同于e_1，e_2，…，e_i的边e_{i+1}，使$\{e_1, e_2, \ldots, e_i, e_{i+1}\}$无回路且$e_{i+1}$是满足此条件的最小权边。

(4)$i \leftarrow i+1$,转(2)。

Kruskal算法

证明

设T_0为由以上算法构造的一个图，它的结点是图G中的n个结点，T_0的边是e_1 ，$e_{2,}$，…，e_{n-1}。根据构造，T_0没有回路，根据定理7-7.1（2）可知T_0是一棵树，且为图G的生成树。

Kruskal算法

下面证明T_0是最小生成树。

设图G的最小生成树是T，若T与T_0相同，则T_0是G的最小生成树。若T与T_0不同，则T_0中至少有一条边e_{i+1}，使得e_{i+1}不是T的边，但 e_1，e_2，…，e_i 是T的边。因为T是树，我们在T中加上一条边e_{i+1}，必有一条回路r，而T_0是树，所以r中必存在某条边f不在T_0中。对于树T，若以边e_{i+1}置换f，则得到新的一棵树T'，但T'的权$C(T')=C(T)+C(e_{i+1})-C(f)$，因为$T$是最小生成树，故$C(T)\leq C(T')$，

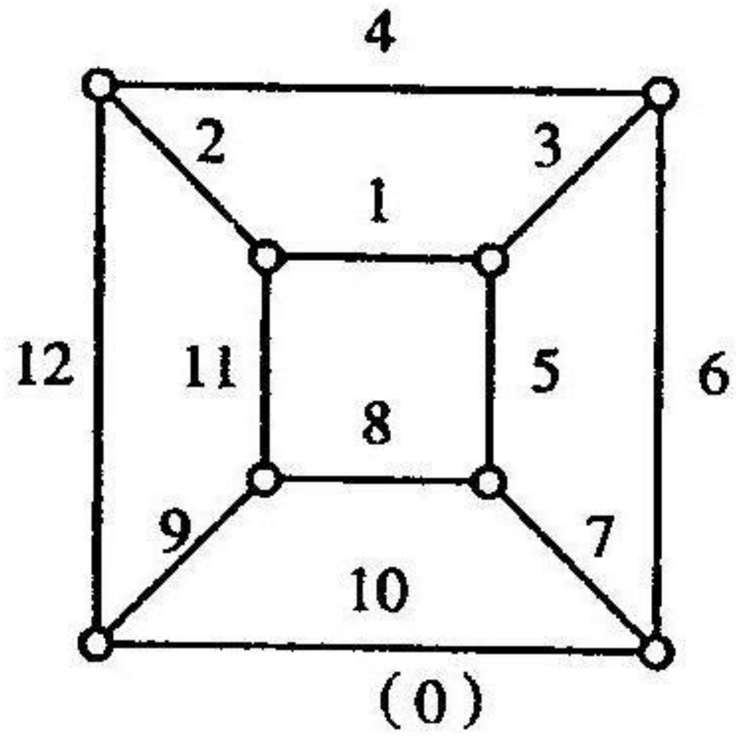
Kruskal算法

即　$C(e_{i+1})-C(f)\geq 0$　或 $C(e_{i+1})\geq C(f)$

因为 e_1，e_2，…，e_i，是T’的边，且在{ e_1，e_2，…，e_i，e_{i+1}} 无回路，故$C(e_{i+1})>C(f)$不可能成立，因为否则在T_0中，自e_1，e_2，…，e_i之后将取f而不取e_{i+1}，与题设矛盾。于是$C(e_{i+1})=C(f)$，因此T'也是G的一棵最小生成树，但是T'与T_0的公共边比T与T_0的公共边多1，用T'代替T，重复上面的讨论，直至得到与T_0有n-1条公共边的最小生成树，这时我们断定T_0是最小生成树。

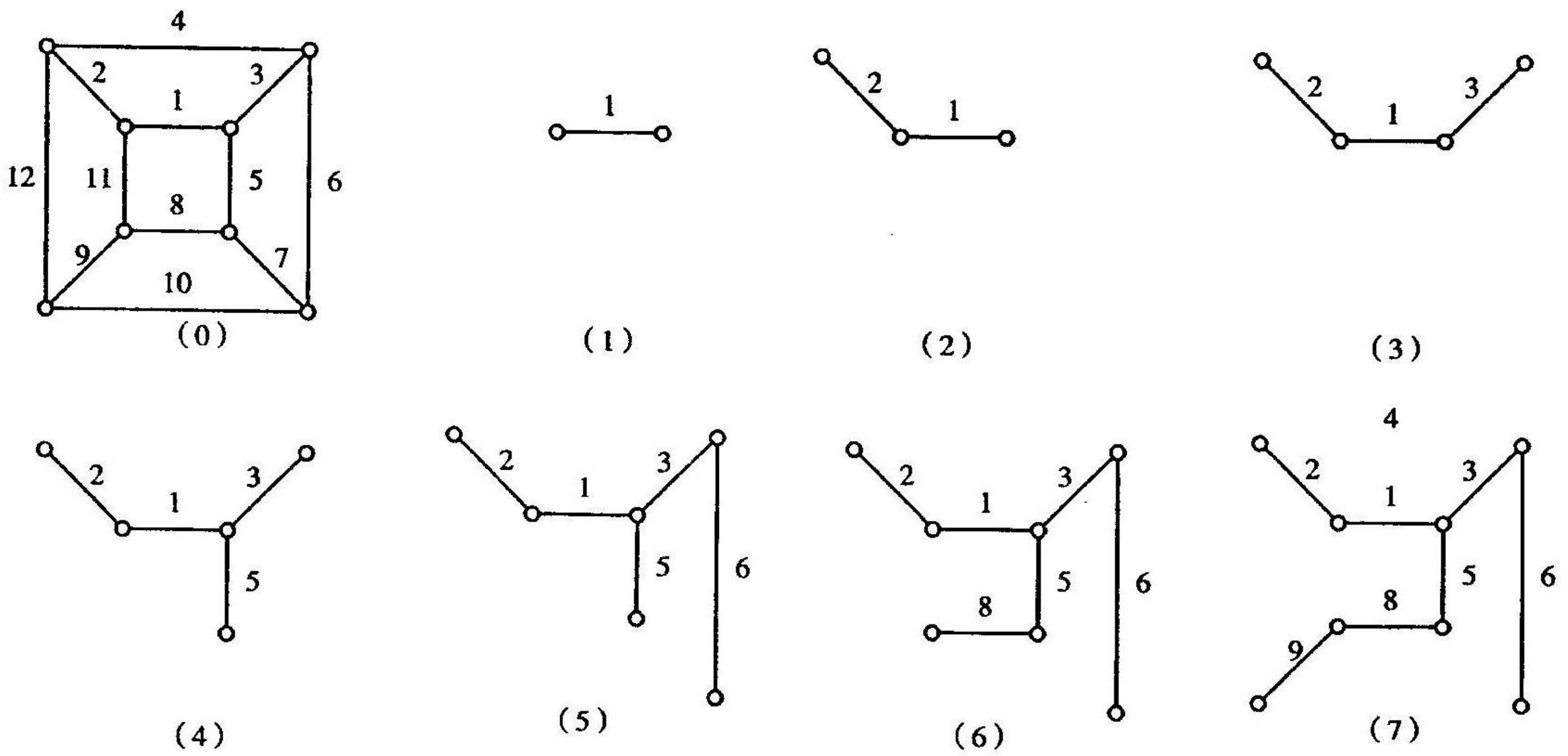
举例

求下图中有权图的最小生成树。



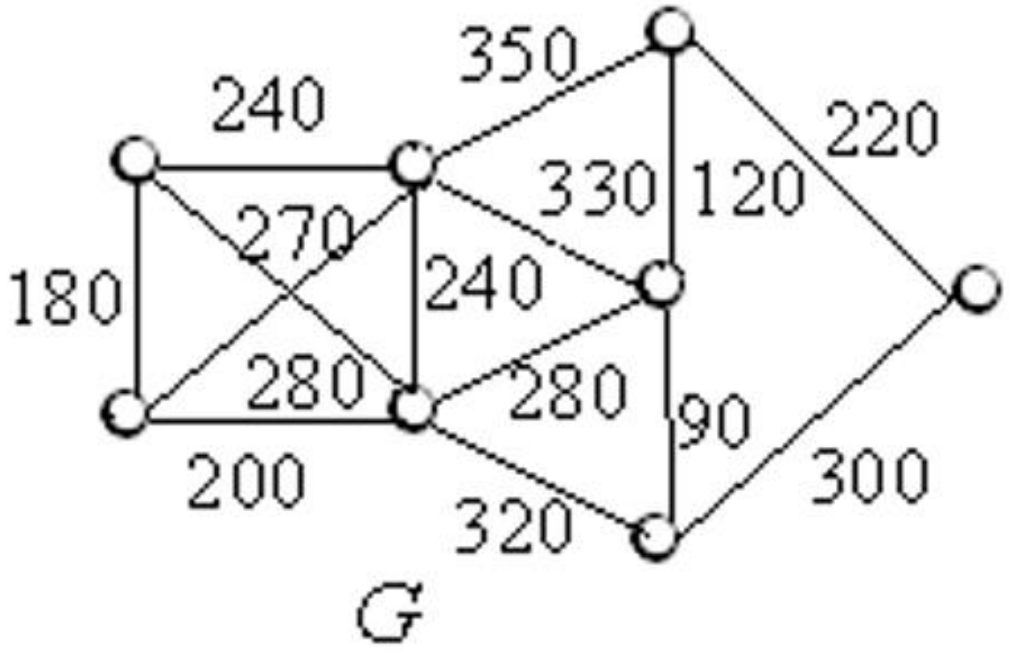
举例

解: 因为 图中n＝8， 所以按算法要执行n－1＝7次。



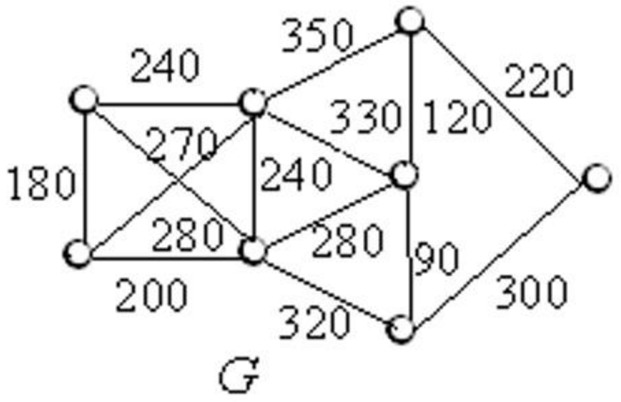
练习

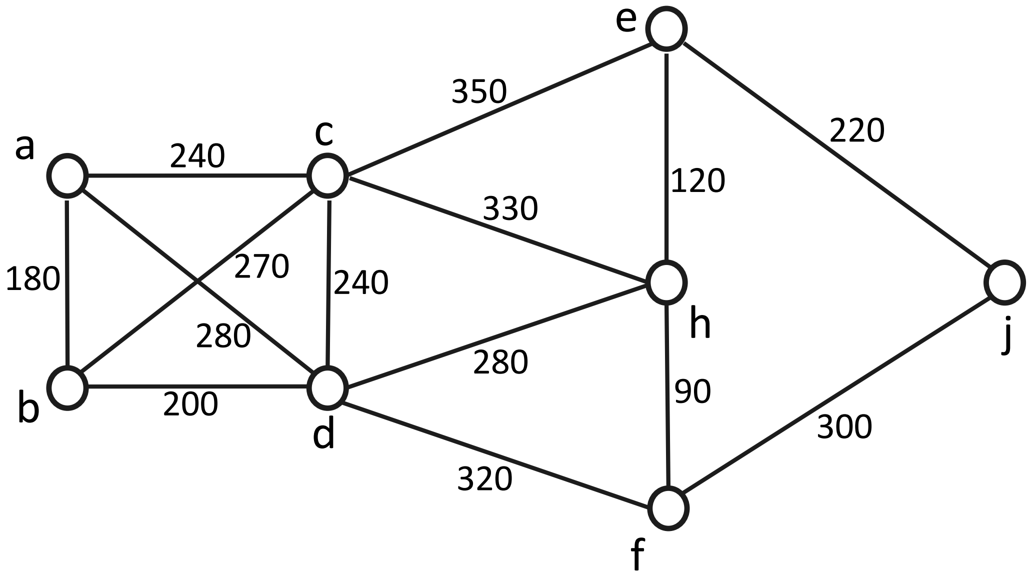
求下图中有权图*G*的最小生成树。



练习

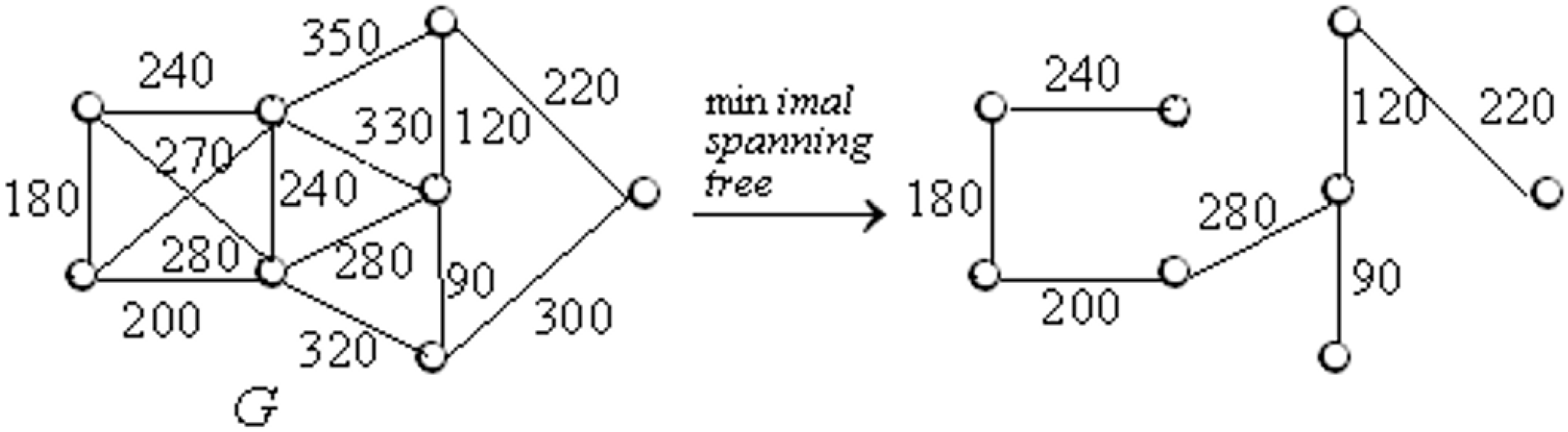
求下图中有权图G的最小生成树。





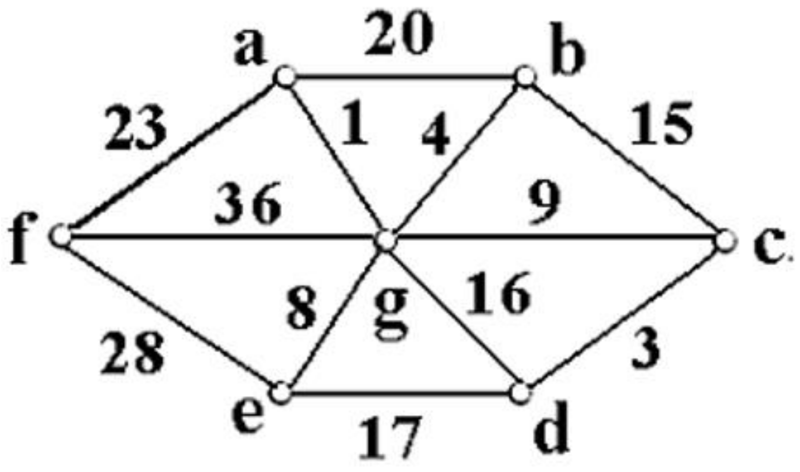
练习

解: 因为 图中$n=8$， 所以按算法要执行$n-1=7$次。



练习

赋权图*G*表示七个城市a,b,c,d,e,f,g及架起城市间直接通讯线路的预测造价。试给出一个设计方案使得各城市间能够通讯且总造价最小，并计算出最小造价。

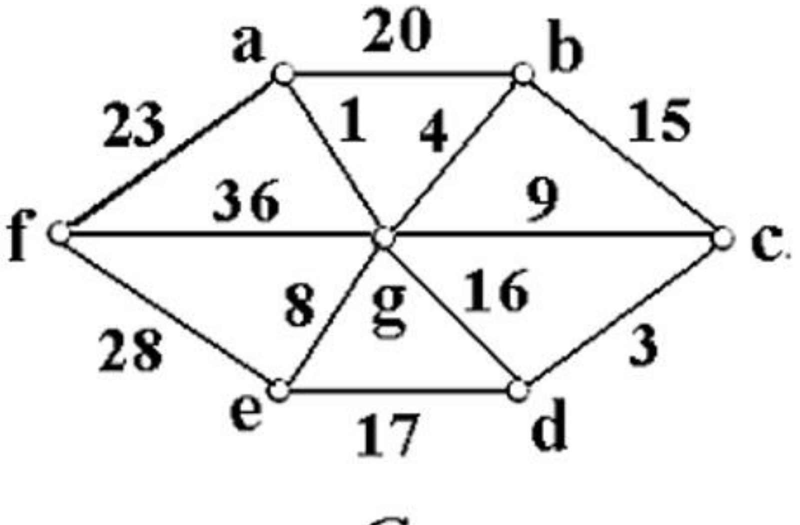


G

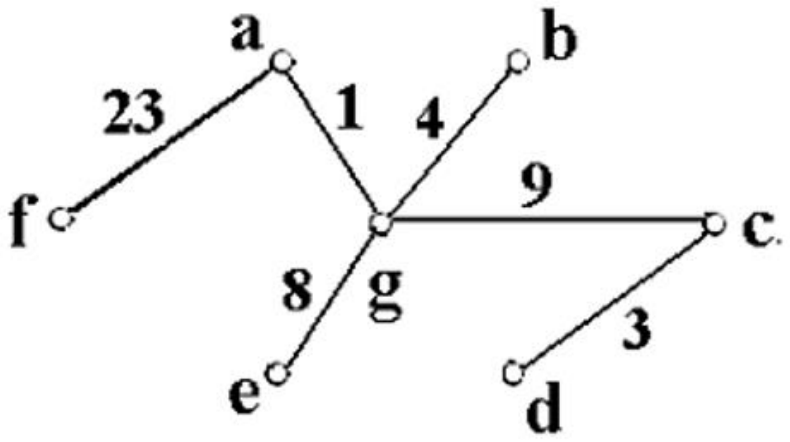
练习

解： 该问题相当于求图的最小生成树问题，此图的最小生成树为图中的T_G ,因此如图T_G架线使各城市间能够通讯，且总造价最小，最小造价为：

$$W(T)=1+3+4+8+9+23=48$$



G



T_G

普里姆算法

该算法于1930年由捷克数学家沃伊捷赫·亚尔尼克（英语：Vojtěch Jarník）发现；并在1957年由美国计算机科学家罗伯特·普里姆（英语：Robert C. Prim）独立发现；1959年，艾兹格·迪科斯彻再次发现了该算法。因此，在某些场合，普里姆算法又被称为DJP算法、亚尔尼克算法或普里姆－亚尔尼克算法。

Prim算法，贪心算法

1).输入：一个加权连通图，其中顶点集合为V，边集合为E；

2).初始化：Vnew = {x}，其中x为集合V中的任一节点（起始点），Enew = {},为空；

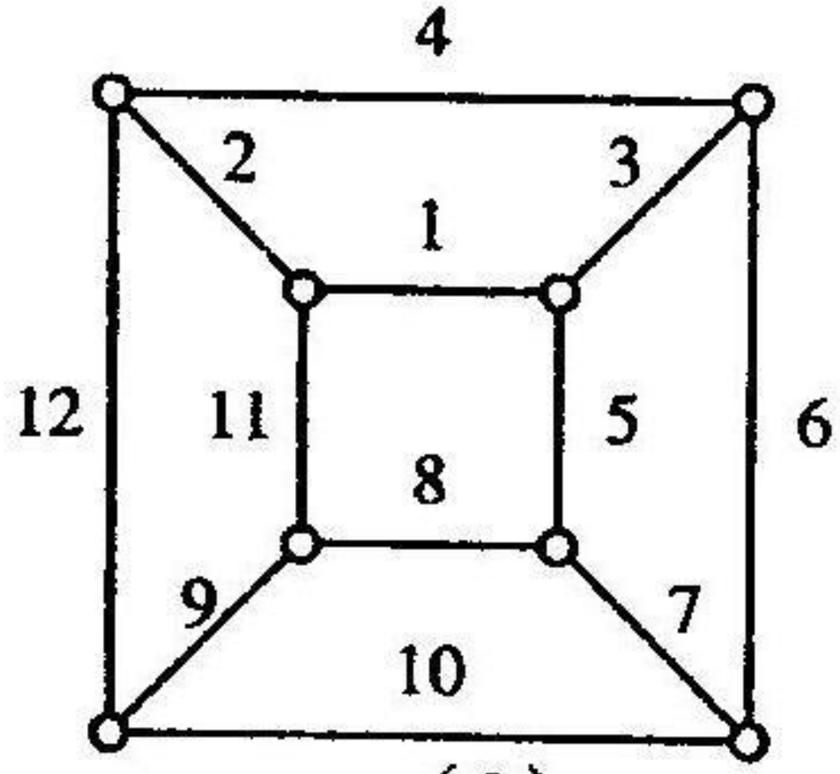
3).重复下列操作，直到Vnew = V：

a.在集合E中选取权值最小的边<u, v>，其中u为集合Vnew中的元素，而v不在Vnew集合当中，并且v∈V（如果存在有多条满足前述条件即具有相同权值的边，则可任意选取其中之一）；

b.将v加入集合Vnew中，将<u, v>边加入集合Enew中；

4).输出：使用集合Vnew和Enew来描述所得到的最小生成树。

Prim算法，贪心算法



(0)

例题：单链聚类——最小生成树

在数据分析中经常用到各类聚类分析（无监督），所谓聚类：把数据集D中的时间按照题目之间的相似度聚集成若干个子类。

单链聚类（层次聚类方法中的一种，单链技术擅长于处理非椭圆形状的簇，但对噪声和离群点很敏感）：设有一组离散数据D={$a_1, a_2,\dots, a_n$}，D上定义了一个相似度函数d 。对于任何两个数据a_i, a_j ∈D, a_i, a_j的相似度函数的值为$d(i,j)$，通常取$d(i,j)$ ≥0，并且d具有对称性，即$d(i,j)$} = $d(j,i)$} 。

给定正整数k(1< k <n),D的一个k聚类是D的一个k划分$\pi = \{C_1，C_2，\dots, C_k\}$，我们希望同一子类中数据尽可能接近，而不同子类中数据尽可能远离。因此定义如下的π的最小间隔D(π)。

对任意两个不同的子类C_s，C_t，定义他们之间的聚类D(C_s，C_t)是C_s中数据与C_t中数据相似度的最小值，即

$$\mathrm{D}(C_s，C_t) = \min\{d(i,j) | a_i \in C_s, a_j \in C_t\}$$

k聚类$\pi = \{C_1，C_2，\dots, C_k\}$的最小间隔

$$\mathrm{D}(\pi) = \min\{\mathrm{D}(C_s，C_t) | C_s, C_t \in \pi，1 \le i < j \le k\}$$

问题：给定数据集D和D上的相似度函数d以及正整数k，如何求使得D(π)达到最大值的k聚类π？

例题：单链聚类——最小生成树

解题思路：可以利用最小生成树的Kruskal算法解决这个问题。

定义带权完全图G=<V,E,d>，其中V={1,2,...,n}，对于任意i,j ∈V，i≠j,边（i,j）的权为$d(i,j)$。根据Kruskal算法，先将边按照权从小到大顺序排序为$e_1, e_2,..., e_{n(n-1)/2}$。初始T中没有边，由n个孤立顶点构成的森林，即T有n个连通分支。接着，依次按照权从小到大的顺序考察G的每条边，只要不构成圈就把它加入T中。在加入边的过程中计数T的连通分支数，直到T恰好含有k个连通分支时算法停止。这时所得的k个连通分支恰好就是所求的k个子类C_1，C_2，$...,C_k$，他的最小间隔达到最大。

谢 谢