7-7 树与生成树 (tree & spanning tree)

树是图论中最主要的概念之一，而且是最简单的图之一。它在计算机科学中应用非常广泛。

我们从一个问题谈起，下图是通讯线路图(图7-7.1)。

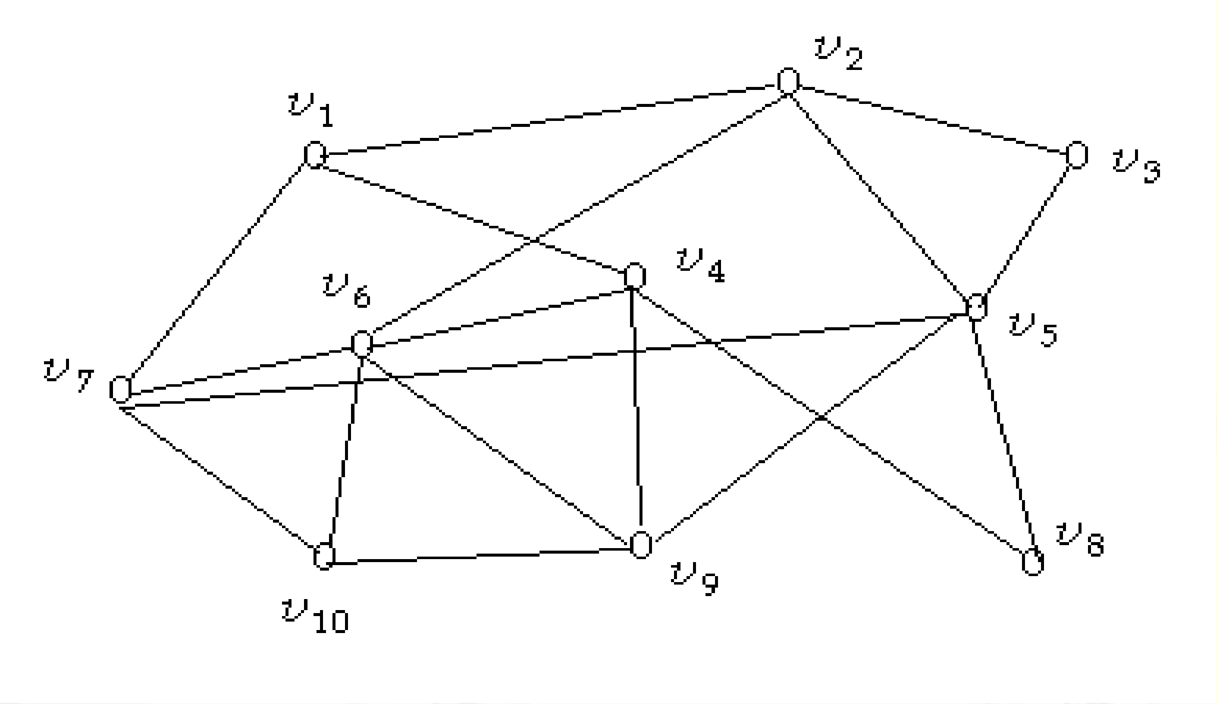

图7-7.1 通讯线路图

其中v_1，$v_2,\ldots,v_{10}$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。

定义7-7.1 树、森林(tree, forest)

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。

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。

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。

(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$k\geq 2$相矛盾。

(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)≥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) ≥2，则2e≥2v，即e≥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。

(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。

(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$deg(v_i) \geqslant 1$

若T中每一个结点的度数大于等于2，

则$\sum deg(v_i) \geqslant 2v$，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

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

故T至少有两个结点度数为1。

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。

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。

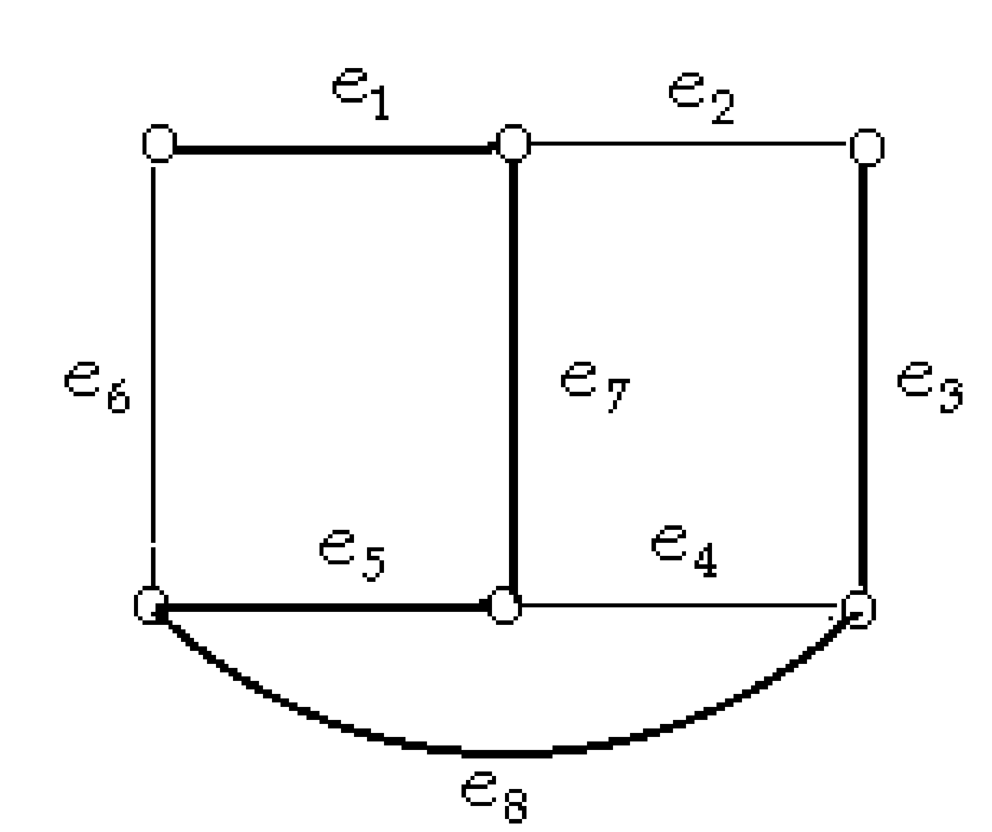

图7-7.3 生成树

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。

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)。

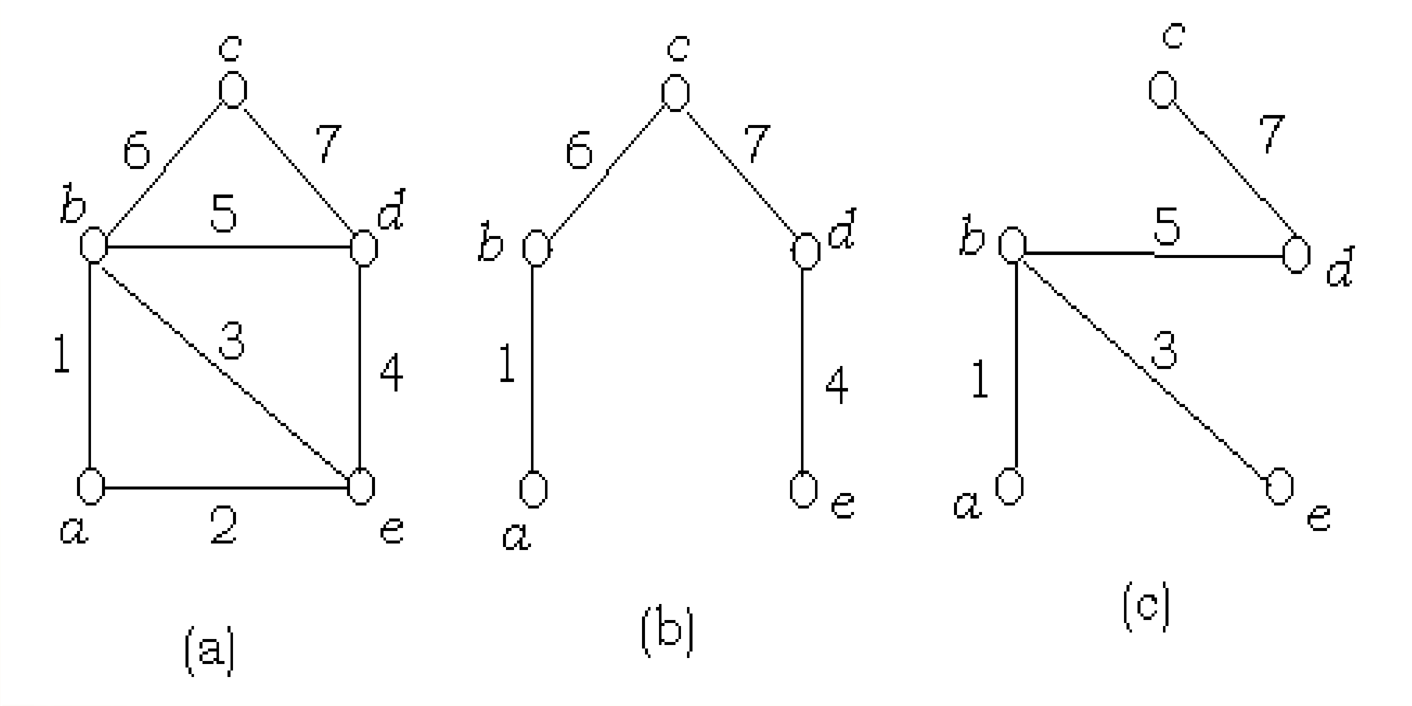

图7-7.4 生成树

假定G是一个有n个结点和m条边的连通图，则G的生成树正好有n-1条边。因此要确定G的一棵生成树，必须删去G中的$m-(n-1)=m-n+1$条边。该数$m-n+1$称为连通图G的秩。

定理7-7.4 一条回路和任意一棵生成树的补至少有一条公共边。

证明 若有一条回路和一棵生成树的补没有公共边，那么这回路包含在生成树之中，然而这是不可能的，因为一棵生成树不能包含回路。

定理7-7.5 一个边割集和任何生成树至少有一条公共边。

证明 若有一条边割集和一棵生成树没有公共边，那么删去这个边割集后，所得的子图必然包含该生成树，这意味着删去边割集后仍然连通，与边割集定义矛盾。

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。

现在讨论一般带权图的情况。

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。

定义7-7.3 最小生成树

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。

定理7-7.6(Kruskal) 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边。

(4)$i \leftarrow i+1$,转(2)。

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可知T_0是一棵树，且为图G的生成树。

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$C(T)\leqslant C(T')$，

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

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。

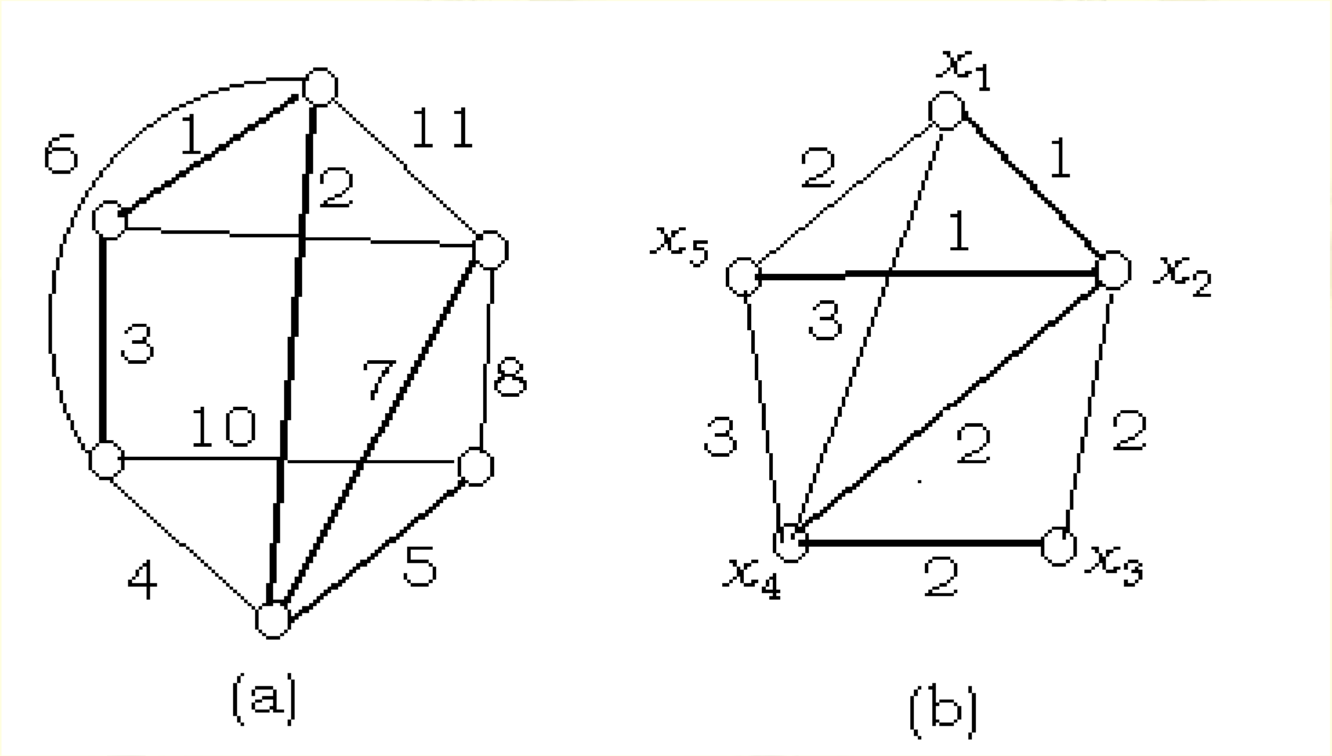

图7-7.5 带权的图

例如图7-7.5(a)中给出一个带权连通图。粗线表示按上述算法得到的最小生成树。

以上算法假设G中边权不相同，实际上，这种算法完全适用于任意边权的情况，若有两条边的权相同，我们可以用其中一条边的权改变一个很小的量，因为G中的边是有限的，总可选择这个改变量而不影响最小生成树的最小性。

图7-7.5(b)中的粗线表示了最小生成树。

作业

P327 (2)

(6) a)