第7章 树及其应用

计算机工程与科学学院 封卫兵

7.1 无向树

7.1.1 无向树的定义及其性质

7.1.2 生成树

7.1.1 无向树的定义及其性质

无向树的定义

无向树： 连通无回路的无向图；

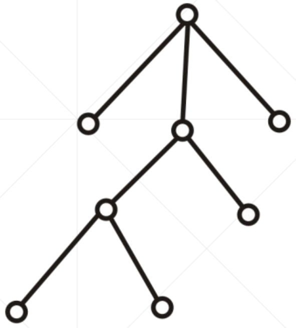
平凡树： 平凡图；

森林： 每个连通分支都是树的非连通的无向图；

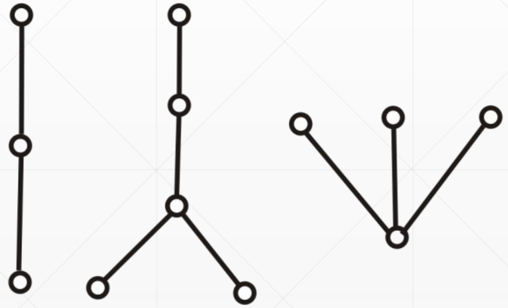
树叶： 树中度数为 1 的顶点；

分支点： 树中度数 ≥ 2 的顶点．(不是树枝)

例：



(a)



(b)

7.1.1 无向树的定义及其性质

无向树的性质

定理7.1 设 $G=\langle V,E\rangle$ 是 n 阶 m 条边的无向图，下面各命题是等价的：

1) G 是树 (连通无回路);

2) G 中任意两个顶点之间存在唯一的路径;

3) G 是连通的且 $m=n-1$;

4) G 中无回路且 $m=n-1$;

5) G 中无回路，但在任何两个不相邻的顶点之间加一条边，所得图中有唯一的一条初级回路;

6) G 是连通的且 G 中任意一条边均为桥 .

7.1.1 无向树的定义及其性质

定理7.1的证明

(1)⇒(2)：G 是树 (连通无回路) $\Rightarrow$ G 中任意两个顶点之间存在唯一的路径

由连通性，任意 2 个顶点之间有一条路径 .

又，假设某 2 个顶点之间有 2 条路径，

则这 2 条路径可组合成一条回路，与树的定义矛盾 .

7.1.1 无向树的定义及其性质

定理7.1的证明（续）

(2)⇒(3)：G 中任意两个顶点之间存在唯一的路径 $\Rightarrow$

G 是连通的且 $m = n - 1$

显然连通．要证 $m = n - 1$. 对 n 作归纳证明.

当 $n = 1$时，显然 $m = 0$，结论成立．

假设当 $n \le k\ (k \ge 1)$ 时结论成立，考虑 $n = k + 1$.

在 G 中任取一条边 $e = (u, v)$，它是 u, v 之间唯一的通路，

删去e, G 被分成 2个连通分支，设它们分别有 n_1, n_2 个顶点

和 m_1, m_2 条边，显然 $n_1 \le k, n_2 \le k$.

由归纳假设，$m_1 = n_1 - 1, m_2 = n_2 - 1$, 得 $m = m_1 + m_2 + 1 = n - 1$.

7.1.1 无向树的定义及其性质

定理7.1的证明（续）

(3)⇒(4)：G 是连通的且 $m = n - 1 \Rightarrow G$ 中无回路且 $m = n - 1$

假设有回路。

任取一个回路，删去回路中的一条边，所得图仍是连通的 .

重复这个做法 r 次，直到没有回路为止，得到一棵树，

它有 n 个顶点 $m - r$ 条边，$r > 0$. 由(1)⇒(2)⇒(3)，

得 $m - r = n - 1$，**矛盾**.

（此为破圈法）

1) G是树(连通无回路);
2) G中任意两个顶点之间存在唯一的路径;
3) G是连通的且 $m = n - 1$;

7.1.1 无向树的定义及其性质

定理7.1的证明（续）

(4)⇒(1)： G 中无回路且 $m = n - 1 \Rightarrow G$是树(连通无回路)

只需证 G 连通 .

假设 G 不连通， 有 $p\ (p > 1)$ 个连通分支.

设第 k 个连通分支有 n_k 个顶点和 m_k 条边， 由(1)⇒(2)⇒(3),

$m_k = n_k - 1$， 从而得到 $m = n - p$，

矛盾 .

1) G是树(连通无回路);
2) G中任意两个顶点之间存在唯一的路径;
3) G是连通的且 $m = n - 1$;

7.1.1 无向树的定义及其性质

定理7.1的证明（续）

(1)⇒(5)：G 是树 (连通无回路) $\Rightarrow$ G 中无回路，但在任何两个不相邻的顶点之间加一条边，所得图中有唯一的一条初级回路 .

由(1)⇒(2), 任意 2 个不相邻的顶点之间有一条唯一的路径，故在这 2 个顶点之间添加一条新边，必得到一条唯一的初级回路 .

1) G是树(连通无回路);
2) G中任意两个顶点之间存在唯一的路径;
3) G是连通的且 $m = n - 1$;

7.1.1 无向树的定义及其性质

定理7.1的证明（续）

(5)⇒(6)：G 中无回路，但在任何两个不相邻的顶点之间加一条边，所得图中有唯一的一条初级回路 $\Rightarrow G$ 是连通的且 G 中任意一条边均为桥 .

首先，任意 2 个不相邻的顶点之间都有一条通路，否则在它们之间添加一条新边不可能构成回路，故 G 连通 .

其次，若 G 中有边不是桥，则删去这条边 G 仍是连通的，这条边必在一条回路上，与 G 中无回路**矛盾**.

7.1.1 无向树的定义及其性质

定理7.1的证明（续）

(6)⇒(1)：G 是连通的且 G 中任意一条边均为桥 $\Rightarrow$ G 是树(连通无回路)

G 连通，只需证无回路 .

假设 G 中有回路，删去回路上任意一条边，G 仍连通，

故删除的边不是桥，**矛盾** .

7.1.1 无向树的定义及其性质

无向树的性质(续)

定理7.2 非平凡的无向树至少有两片树叶 .

证明： 设有 $n\ (n>1)$ 个顶点，由定理7.1，则边数为 $n-1$ 条，

由握手定理：

$$2(n-1)=\sum_{i=1}^{n} d(v_i)$$

设有 x 片树叶，则有 $n-x$ 个分支点 $(d\,(v_i)\geq 2)$，则 $\sum_{i=1}^{n-x} d(v_i)\geq 2(n-x)$

$$2(n-1)\geq x+2(n-x)$$

所以，解得 $x\geq 2$.

7.1.1 无向树的定义及其性质

例： 画出所有 6 阶非同构的无向树 .

解： 由条件知，该树中有 5 条边, 总度数等于 10

可能的度数列：应满足什么规则?

（至少有两片树叶）

（应有偶数个奇数）

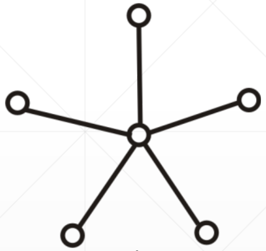
(1) 1,1,1,1,1,5

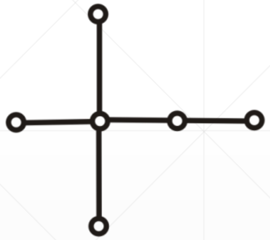
(2) 1,1,1,1,2,4

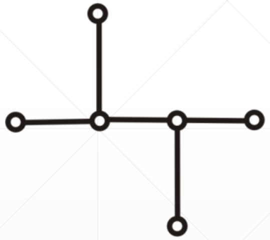
(3) 1,1,1,1,3,3

(4) 1,1,1,2,2,3

(5) 1,1,2,2,2,2


(1)


(2)


(3)


(4a)


(4b)


(5)

7.1.1 无向树的定义及其性质

例： 已知无向树 T 中, 有 1 个 3 度顶点，2 个 2 度顶点，其余顶点全是树叶. 试求树叶数，并画出满足要求的非同构的无向树.

解： 设有 x 片树叶，由定理7.1,

则边数为

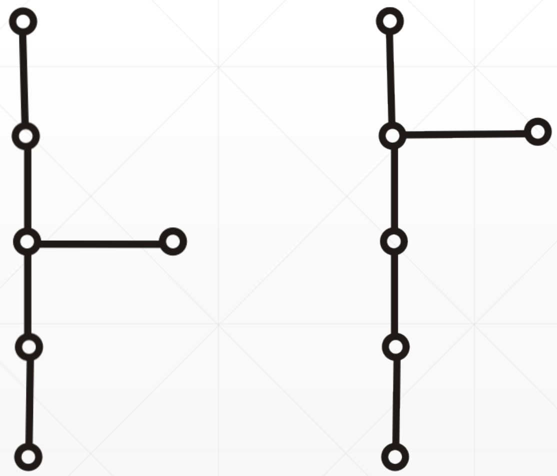
$$x+1+2-1=x+2$$

由握手定理:

$$2\times(x+2)=1\times3+2\times2+x$$

解得 $x=3$，故 T 有 3 片树叶.

T 的度数列为 1, 1, 1, 2, 2, 3

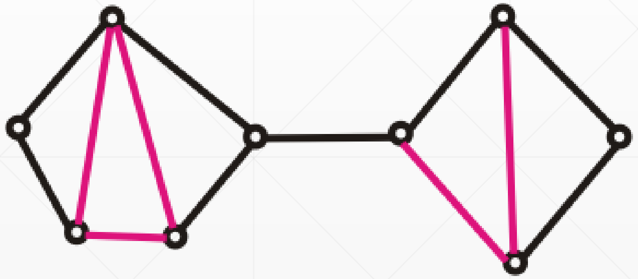


7.1.2 生成树

生成树

定义7.2 设 G 是无向连通图，若 G 的生成子图 T 是一棵树，则称 T 是 G 的生成树. G 在 T 中的边称作 T 的树枝，不在 T 中的边称作 T 的弦 . T 的所有弦的集合的导出子图称作 T 的余树 .

例： 图中黑边构成生成树红边构成余树 .



注： (1) 余树一般不是树;

(2) 生成树不唯一 .

7.1.2 生成树

生成树的存在性

定理7.3 任何无向连通图都有生成树 .

证明： （破圈法）

若图中无圈，则图本身就是自己的生成树 .

否则，删去圈上的任一条边，不破坏连通性，重复进行

直到无圈为止，得到图的一棵生成树 .

7.1.2 生成树

生成树的存在性

推论1 设 n 阶无向连通图有 m 条边，则 $m \geq n-1$.

推论2 设 n 阶无向连通图有 m 条边，则它的生成树的余树有 $m-n+1$ 条边 .

证明： 因为生成树是 n 阶树，有 $n-1$ 条边，从而余树有 $m-n+1$ 条边 .

注： n 阶无向连通简单图，则 $n-1 \leq m \leq n(n-1)/2$.

7.1.2 生成树

最小生成树

定义： 图 G 的每一条边 e 附加一个实数 $w(e)$，称作边 e 的权 .

图 G 连同附加在边上的权称作带权图，记作 $G = <V, E, W>$.

设 H 是 G 的子图，H 所有边的权的和称作 H 的权，记作$W(H)$.

最小生成树：带权图权最小的生成树 .

7.1.2 生成树

最小生成树（续）

避圈法 (Kruskal)

1) 将所有非环边按权从小到大排列，设为 $e_1, e_2, \ldots, e_m$；

2) 令 $T = \varnothing$;

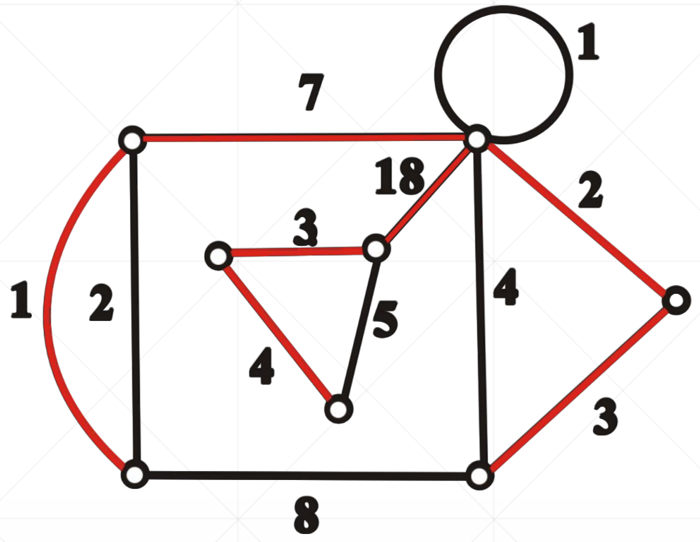
3) For $k = 1$ to m Do

若 e_k 与 T 中的边不构成回路，则将 e_k 加入 T 中 .

注：用破圈法也可以，每次删除最大权的边 .

7.1.2 生成树

例： 求图的一棵最小生成树 .



1) 排列：

$$1, 1, 2, 2, 3, 3, 4, 4, 5, 7, 8, 18$$

2) 检查环和平行边，余下：

$$1, 2, 3, 3, 4, 4, 5, 7, 8, 18$$

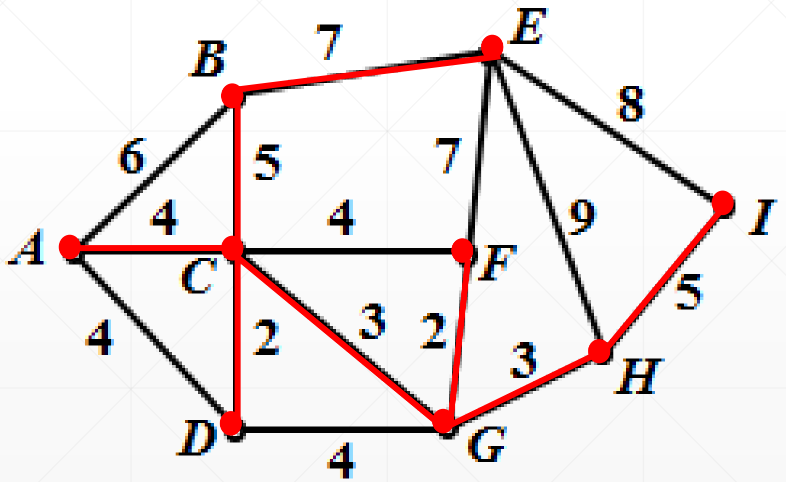
$$W(T) = 1 + 2 + 3 + 3 + 4 + 7 + 18 = 38$$ （8个顶点，7条边）

注： 最小生成树不唯一，但是权是唯一的 .

7.1.2 生成树

例：某单位建设局域网需要铺设光缆，光缆连接的建筑物的位置、建筑物之间可以铺设光缆的线路及线路的长度(单位：米)如图所示. 问：如何铺设才能使光缆的总长度最短?

解：



总长度：$2+2+3+3+4+5+5+7=31$(米)

基本回路与基本回路系统

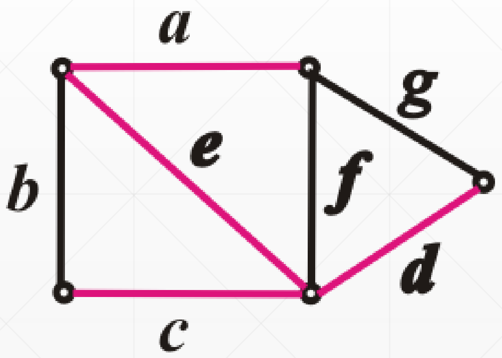
定义7.3　设G是n阶m条边的无向连通图，T是G的一棵生成树，$e_1, e_2, \ldots, e_{m-n+1}$为$T$的弦. G中仅含T的一条弦e_r的圈C_r称作对应弦e_r的基本回路. 称$\{C_1, C_2, \ldots, C_{m-n+1}\}$为对应$T$的基本回路系统

例3

图中红边为一棵生成树,

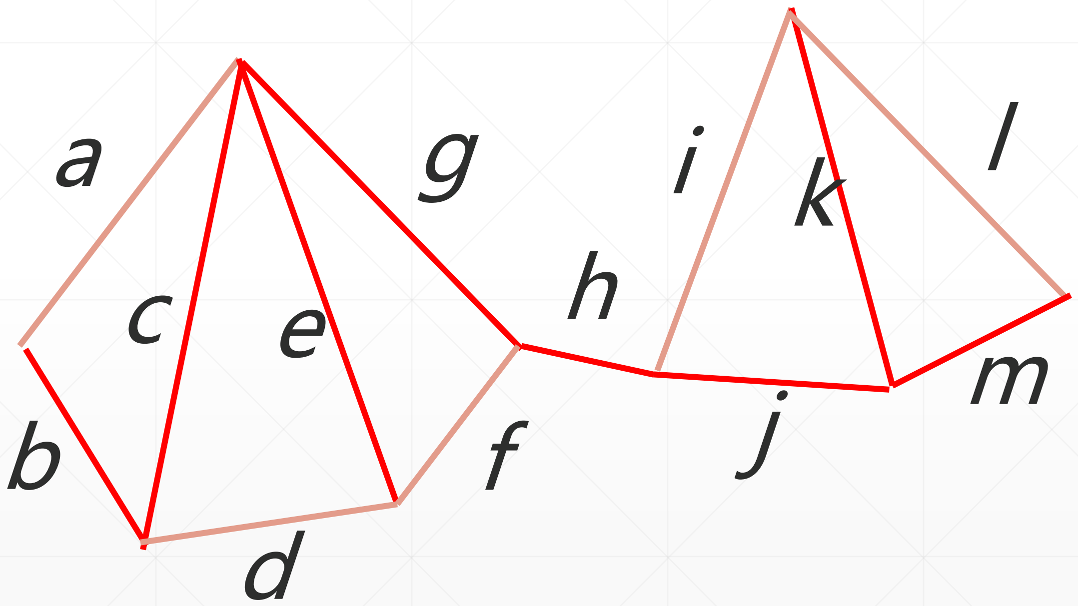
对应它的基本回路系统为

$\{bce,\ fae,\ gaed\}$



练习

图中红边为一棵生成树，对应它的基本回路系统为



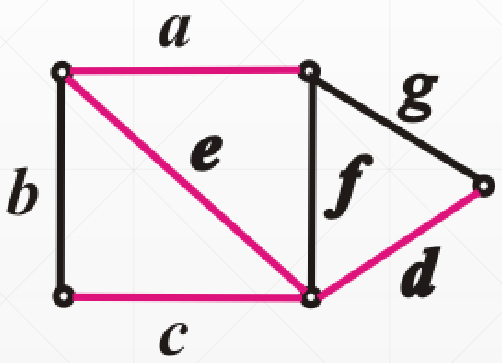
$\{abc,\ dce,\ feg,\ ijk,\ lkm\}$

基本割集与基本割集系统

定义7.4 设T是n 阶连通图G的一棵生成树, e_1', e_2', …, e'_{n-1}为T的树枝，S_i是G的只含树枝e_i', 其他边都是弦的割集, 称S_i为对应树枝e_i'的基本割集. 称{S_1, S_2, …, S_{n-1}}为对应T的基本割集系统

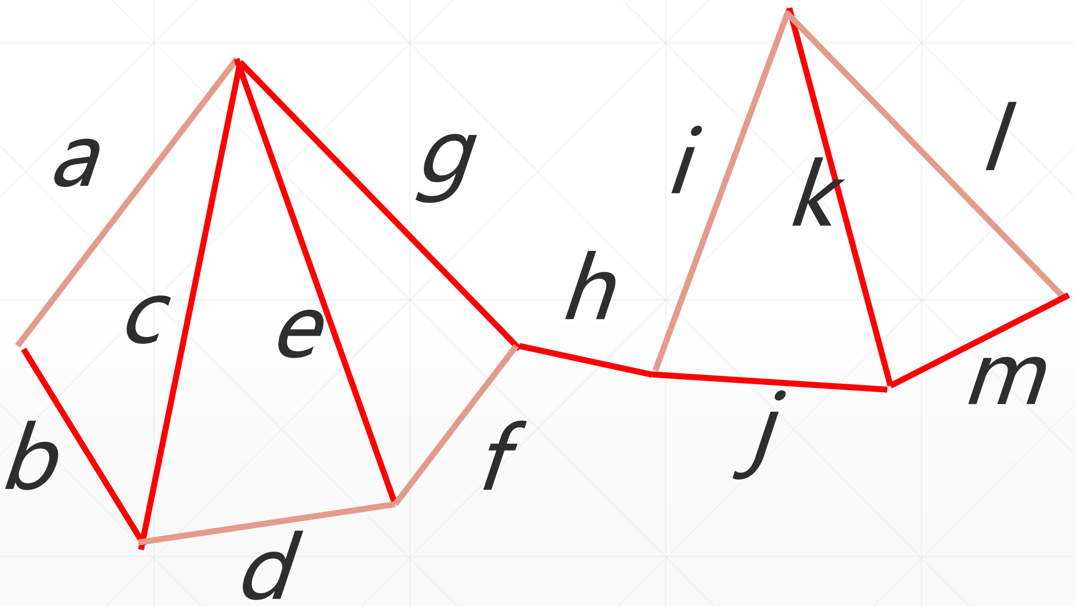
例4 图中红边为一棵生成树, 对应它的基本割集系统为

{{a,f,g}, {e,b,f,g}, {c,b}, {d,g} }



练习

图中红边为一棵生成树，对应它的基本割集系统为



$\{ba, cad, edf, fg, hi, jil, kl, m\}$

图论小结（二部图）

二部图

判别定理：

无向图$G=<V,E>$是二部图当且仅当G中无奇长度的回路

或：AB法

存在完备匹配的条件：

设二部图$G=<V_1,V_2,E>$，$|V_1|\le|V_2|$，

Hall定理：G中存在V_1到V_2的完备匹配当且仅当V_1中任意$k(1\le k\le|V_1|)$个顶点至少与V_2中的k个顶点相邻(相异性条件)；

或：如果存在正整数 t，使得V_1中每个顶点至少关联 t 条边，而V_2中每个顶点至多关联 t 条边(t 条件)，则G中存在V_1到V_2的完备匹配 （充分条件)

图论小结（欧拉图）

欧拉图：含有经过所有顶点且每条边恰好经过一次的回路（简单回路）

判别：

无向图G是欧拉图<u>当且仅当</u>G是连通的且无奇度顶点

有向图D是欧拉图<u>当且仅当</u>D是连通的且所有顶点的入度等于出度.

图论小结（哈密顿图）

哈密顿图：含有经过图中所有顶点一次且仅一次的回路（初级回路）

判别（必要条件）：

- 若无向图$G=<V, E>$是哈密顿图，则对于V 的任意非空真子集V_1 均有 $p(G-V_1)\leq|V_1|$.
- 有割点的图不是哈密顿图

判别（充分条件）：

设G是$n(n\geq3)$阶无向简单图,

- 若任意两个不相邻的顶点的度数之和大于等于 n, 则 G 为哈密顿图.
- 若$\delta(G)\geq n/2$, 则 G 是哈密顿图
- 设D是 n $(n\geq 3)$阶有向完全图, 则D是哈密顿图

图论小结（平面图）

平面图：能将图G除顶点外边不相交地画在平面上

性质：

- **平面图各面的次数之和等于边数的2倍**
- **欧拉公式：设G为n阶m条边 r个面的<u>连通平面图</u>, 则**

$$n-m+r=2$$

- **设平面图G有 p ($p\geq 2$) 个连通分支, 则** $n-m+r=p+1$
- **设G为n阶<u>连通平面图</u>, 有m条边, 且每个面的次数不小于 l ($l\geq 3$), 则**

$$m \leq \frac{l}{l-2}(n-2)$$

判定定理：

一个图是平面图<u>当且仅当</u>：

- **它既不含与K_5同胚的子图, 也不含与$K_{3,3}$同胚的子图，或：**
- **它既无可收缩为K_5的子图, 也无可收缩为$K_{3,3}$的子图**

图论小结（树）

树：连通无回路的无向图

性质：

- 连通无回路;
- G中任意两个顶点之间存在唯一的路径;
- G是连通的且$m=n-1$;
- G中无回路且$m=n-1$;
- G中无回路, 但在任何两个不相邻的顶点之间加一条边，所得图中有唯一的一条初级回路.
- G是连通的且G中任意一条边均为桥.

- 非平凡的无向树至少有两片树叶

- 任何无向连通图都有生成树

- 带权图权最小的生成树为最小生成树，不唯一

例题

例 证明：树都是二部图

证：树是连通无回路的无向图，这里的回路是指初级回路或简单回路。

树中若有回路，一定是复杂回路，且在回路上的每条边均出现偶数次。

所以树中没有奇数长度的回路。

由二部图的判定定理，可知树都是二部图。

例题

例 证明：除平凡树外，树都不是欧拉图

证：利用T中有奇度顶点

设T为一棵非平凡的无向树，由定理7.2，T至少有两片树叶，因而T有奇度顶点。

由无向欧拉图的判别定理，T不是欧拉图。

例 证明：除平凡树外，树都不是哈密顿图

证：若T是2阶树，同构意义下，T为K_2，K_2显然不是哈密顿图。

在阶数≥3的树中，设u为T中非树叶顶点，u与v和w相邻，设$e_1=(v,u)$, $e_2=(u,w)$, 则e_1, e_2均为桥，于是 $p(T-u)\geq 2$, 故u为割点。

即在阶数≥3的树中，存在割点（非树叶顶点），根据6.10推论，T不是哈密顿图。

例题

例 证明：树都是平面图

证：设 T 为任何一棵树，则 T 中既没有简单回路，也没有初级回路。

而与 K_5 或 $K_{3,3}$ 同胚的图均有初级和简单回路，

因而 T 中既没有与 K_5 同胚的子图，也没有与 $K_{3,3}$ 同胚的子图，

由库拉图斯基定理，T 是平面图

注：T 是特殊的平面图，T 中只有一个外部面，没有任何内部面，外部面 R_0 的边界是由所有边构成的复杂回路组成，每条边在回路中恰好出现两次。

例题

例 哪些完全二部图是树?

解：在$K_{r,s}$中，若$r\geq 2$且$s\geq 2$，则$K_{r,s}$中必含圈，所以$r\geq 2$, $s\geq 2$时，$K_{r,s}$不是树。

所以完全二部图$K_{1,r}$ ($r\geq 1$)和$K_{s,1}$($s\geq 1$) 为树。

例题

例 下面两组数中，哪些可以为无向树的度数列?

(1)　1, 1, 2, 3, 3, 4　　　　(2) 1, 1, 1, 1, 1, 1, 3, 3, 4

解：(1) 若可以为无向树的度数列，则：

阶数n=　　, 边数m=　　　　　,

由握手定理应该有：6 2m=10=1+1+2+3+3+4=14 6-1=5

矛盾。所以不可能是无向树的度数列。

(2) 若可以为无向树的度数列，阶数n=9, 边数m=9-1=8, 由握手定理应该有：
2m=16=1*6+3+3+4=16

成立。所以是无向树的度数列。

作 业

- 7.8
- 7.16 （v2: 7.19)

研 讨 题

1. 画出具有7个结点的所有非同构的树。
2. 对任意一个图$G=<V，E>$，设$|V|=n$，$|E|=m$，$p(G)=p$，试证明G中至少包含$m-n+p$条不同的回路。
3. 若连通图G的顶点数大于2，则G中至少有2个顶点，证明：将它们去掉后G仍然是连通。

7.2 根树及其应用

- 7.2.1 根树及其分类
- 7.2.2 最优树与哈夫曼算法
- 7.2.3 最佳前缀码
- 7.2.4 根树的周游及其应用
  - 中序行遍法、前序行遍法和后序行遍法
  - 波兰符号法与逆波兰符号法

根树的定义

有向树: 略去方向后为无向树的有向图

根树: 有一个顶点入度为0, 其余的入度均为1的非平凡的有向树

树根: 有向树中入度为0的顶点

树叶: 有向树中入度为1, 出度为0的顶点

内点: 有向树中入度为1, 出度大于0的顶点

分支点: 树根与内点的总称

顶点 v 的层数: 从树根到 v 的通路长度

树高: 有向树中顶点的最大层数

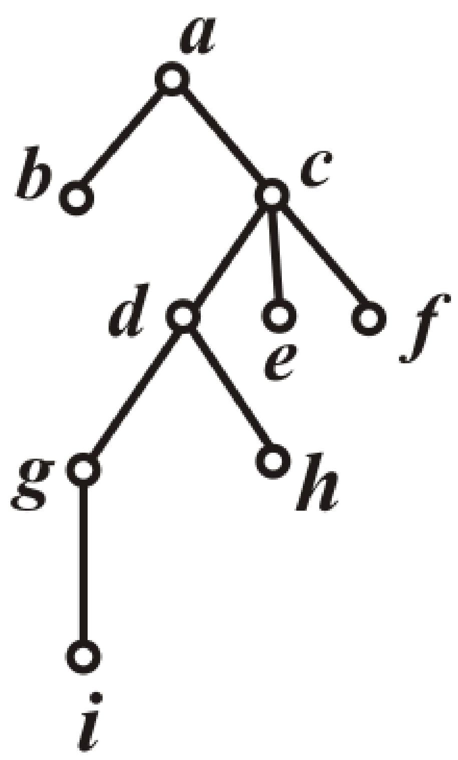
实例

根树的画法：树根放最上方，省去所有有向边上的箭头

右图中

- *a*是树根
- *b,e,f,h,i*是树叶
- *c,d,g*是内点
- *a,c,d,g*是分支点
- *a*为0层;1层有*b,c*; 2层有*d,e,f*;
- 3层有*g,h*; 4层有*i*.

树高为4



家族树

把根树看作一棵**家族树**:

若顶点a邻接到顶点b, 则称b是a的**儿子**, a是b的**父亲**

若b和c为同一个顶点的儿子, 则称b和c是**兄弟**

若$a \neq b$且a可达b, 则称a是b的**祖先**, b是a的**后代**

设v为根树的一个顶点且不是树根, 称v及其所有后代的导出子图为以v为根的**根子树**

将根树每一层上的顶点规定次序后称作**有序树**

根树的分类

r元树:根树的每个分支点至多有r个儿子

r元正则树: 根树的每个分支点恰有r个儿子

r元完全正则树: 所有树叶层数相同的r元正则树

r元有序树: 有序的r元树

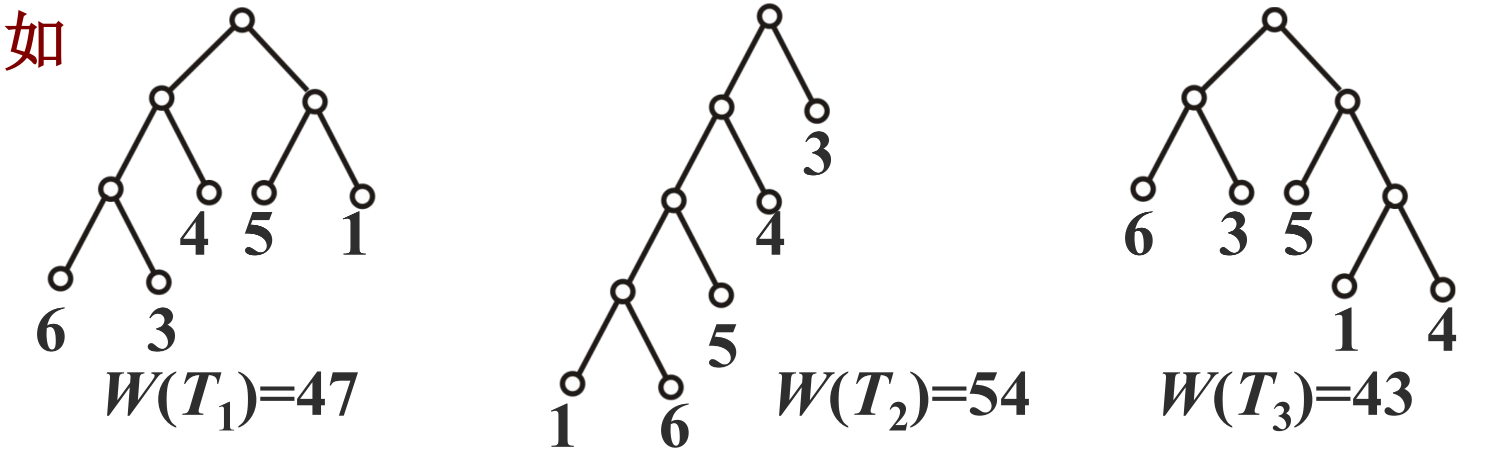
r元正则有序树: 有序的r元正则树

r元完全正则有序树: 有序的r元完全正则树

最优2元树

定义7.10 设2元树T有t片树叶$v_1, v_2, \ldots, v_t$, 树叶的权分别为$w_1, w_2, \ldots, w_t$, 称 $W(t)=\sum_{i=1}^{t} w_i l(v_i)$为$T$的权, 记作$W(T)$, 其中$l(v_i)$是$v_i$的层数. 在所有有$t$片树叶, 带权$w_1, w_2, \ldots, w_t$ 的 2元树中, 权最小的2元树称为最优2元树

例如



求最优2元树的算法

哈夫曼(Huffman)算法

给定实数$w_1, w_2, \ldots, w_t$

① 作t片树叶, 分别以$w_1, w_2, \ldots, w_t$为权

② 在所有入度为0的顶点(不一定是树叶)中选出两个权最小的顶点, 添加一个新分支点, 以这2个顶点为儿子, 其权等于这2个儿子的权之和

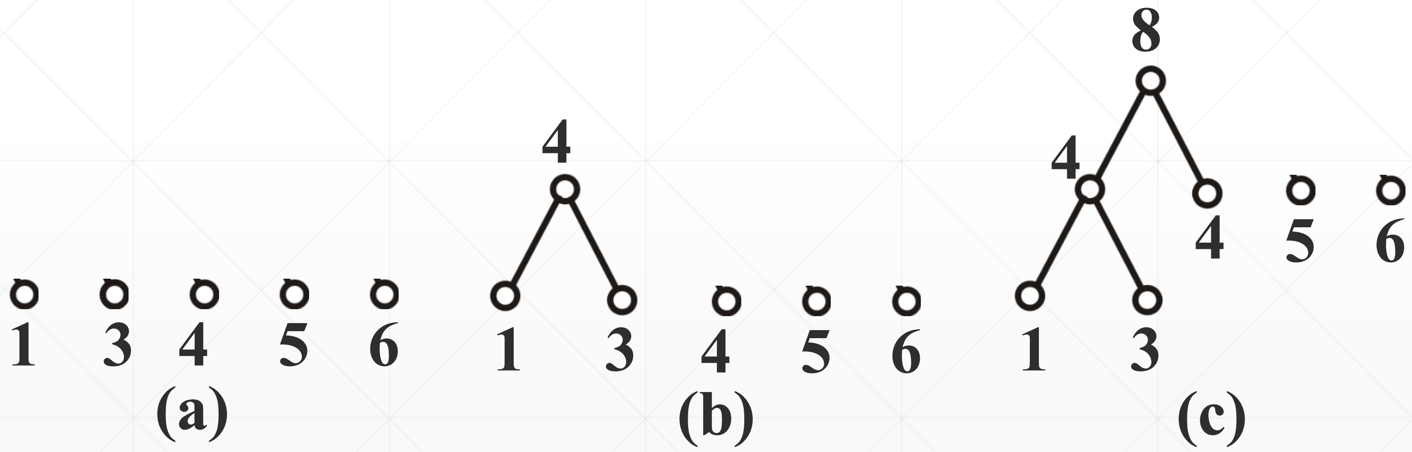
③ 重复②, 直到只有1个入度为0 的顶点为止

$W(T)$等于所有分支点的权之和

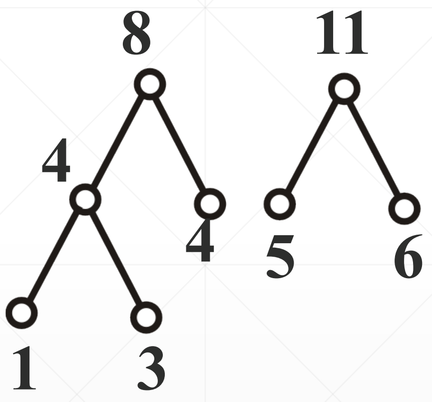
实例

例1 求以1,3,4,5,6为权的最优2元树, 并计算它的权

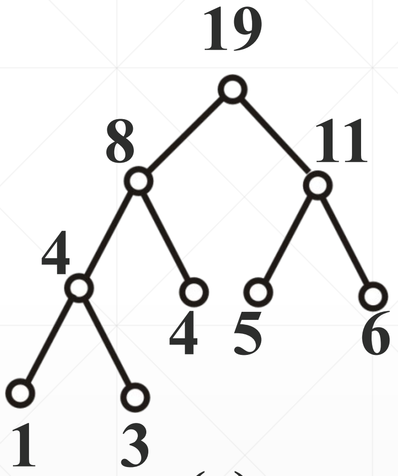
解



实例(续)



(d)



(e)

$W(T)$=4+8+11+19=42

最佳前缀码

定义7.11 设$\alpha=\alpha_1\alpha_2\ldots\alpha_{n-1}\alpha_n$是长度为$n$的符号串, $\alpha_1\alpha_2\ldots\alpha_k$称作$\alpha$的长度为$k$的**前缀**, $k=1,2,\ldots,n-1$.

若非空字符串$\beta_1, \beta_2, \ldots, \beta_m$中任何两个互不为前缀, 则称$\{\beta_1, \beta_2, \ldots, \beta_m\}$为**前缀码**

只出现两个符号(如0与1)的前缀码称作**2元前缀码**

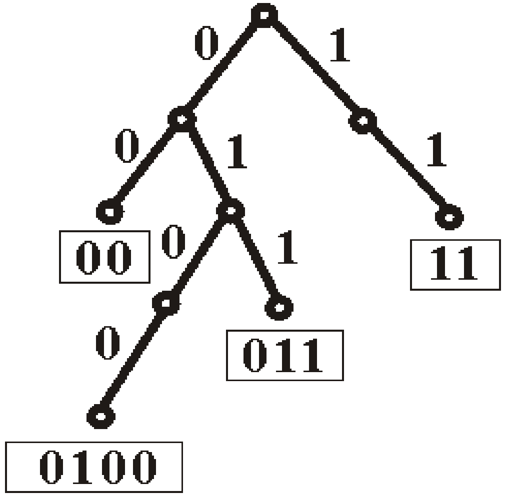
例如 {0,10,110, 1111}, {10,01,001,110}是2元前缀码

{0,10,010, 1010} 不是前缀码

用2元树产生2元前缀码的方法

对每个分支点, 若关联2条边, 则给左边标0, 右边标1; 若只关联1条边, 则可以给它标0(看作左边), 也可以标1(看作右边). 将从树根到每一片树叶的通路上标的数字组成的字符串记在树叶处, 所得的字符串构成一个前缀码.

例如



前缀码

{00, 11, 011, 0100}

实例

例2 在通信中,设八进制数字出现的频率(%)如下:

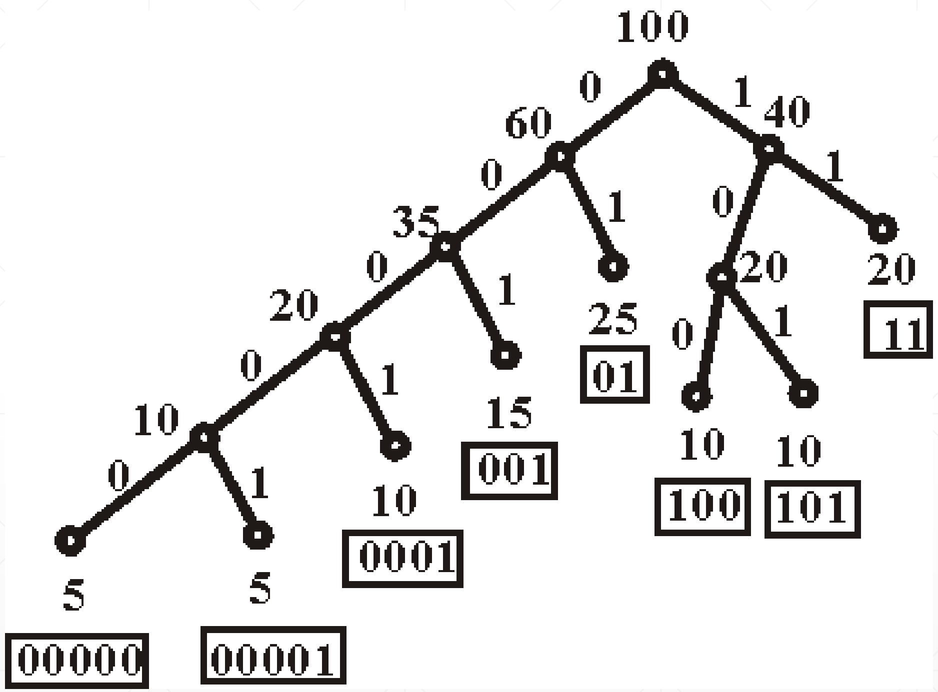
0: 25，1: 20，2: 15，3: 10，4: 10，5: 10，6: 5，7: 5

采用2元前缀码, 求传输数字最少的2元前缀码 (称作最佳前缀码), 并求传输100个按上述比例出现的八进制数字需要多少个二进制数字？若用等长的 (长为3) 的码字传输需要多少个二进制数字?

解 用Huffman算法求以频率(乘以100)为权的最优2元树.

这里$w_1=5, w_2=5, w_3=10, w_4=10, w_5=10, w_6=15, w_7=20, w_8=25$.

实例(续)



编码:

0---01

1---11

2---001

3---100

4---101

5---0001

6---00000

7---00001

传100个按比例出现的八进制数字需$W(T)$=285个二进制数字, 用等长码(长为3)需要用 300个数字.

根树的周游及其应用

对根树的**行遍**或**周游**: 每个顶点访问一次且仅访问一次

行遍2元有序正则树的方式:

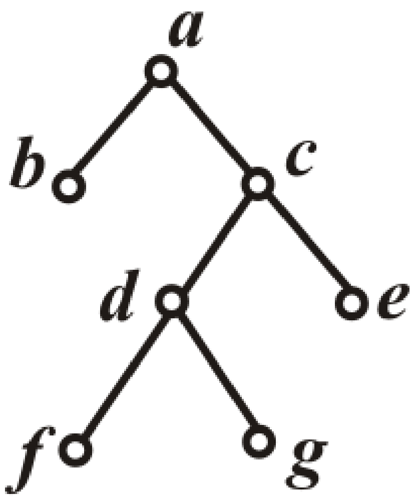
① **中序行遍法**: 左子树、树根、右子树

② **前序行遍法**:树根、左子树、右子树

③ **后序行遍法**: 左子树、右子树、树根

实例

例3



中序行遍：$b\ \underline{a}\ ((f\ \underline{d}\ g)\ \underline{c}\ e)$

前序行遍：$\underline{a}\ b\ (\underline{c}\ (\underline{d}\ f\ g)\ e)$

后序行遍：$b\ ((f\ g\ \underline{d})\ e\ \underline{c})\ \underline{a}$

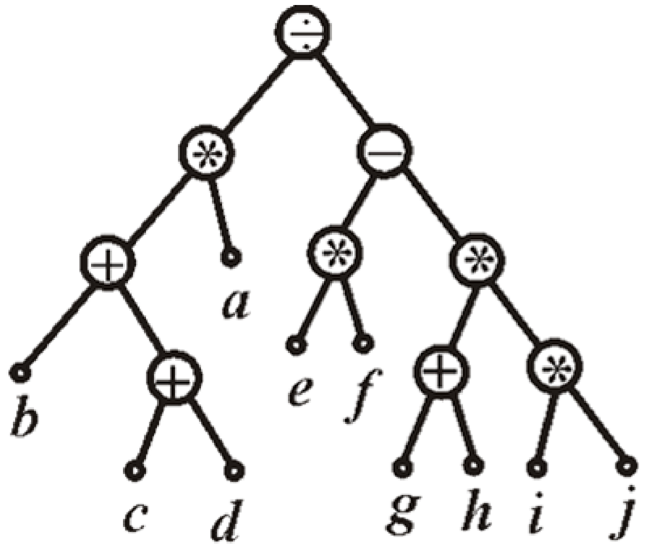
带下划线的是(子)树根，
一对括号内是一棵子树

波兰符号法与逆波兰符号法

用2元有序正则树表示算术运算算式如下: 以中序行遍方式将运算符和数标记在顶点上, 即将运算符放在分支点上, 数放在树叶上, 每个运算符对它所在分支点的2棵子树进行运算, 并规定左子树是被除数或被减数.

例4 右图表示算式

$((b+(c+d))*a)\div((e*f)-(g+h)*(i*j))$



波兰符号法与逆波兰符号法(续)

波兰符号法(前缀符号法): 按前序行遍法访问, 不加括号.

从右到左进行, 每个运算符对其后面紧邻两个数进行运算.

逆波兰符号法(后缀符号法): 按后序行遍法访问, 不加括号.

从左到右进行, 每个运算符对前面紧邻两数运算.

例4(续)

波兰符号法

$\div * + b + c\, d\, a - * e f * + g\, h * i\, j$

逆波兰符号法

$b\, c\, d + + a * e f * g\, h + i\, j * * - \div$