第7章 树

- 7.1 无向树及生成树
- 7.2 根树及其应用

7.1 无向树及生成树

- 无向树与森林
- 生成树与余树
- 基本回路与基本回路系统
- 基本割集与基本割集系统
- 最小生成树与避圈法

无向树

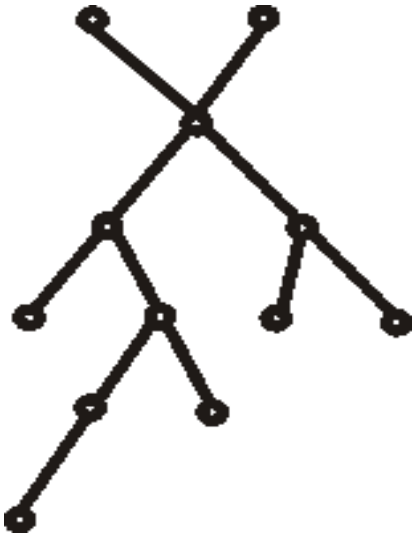
无向树: 无回路的连通无向图

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

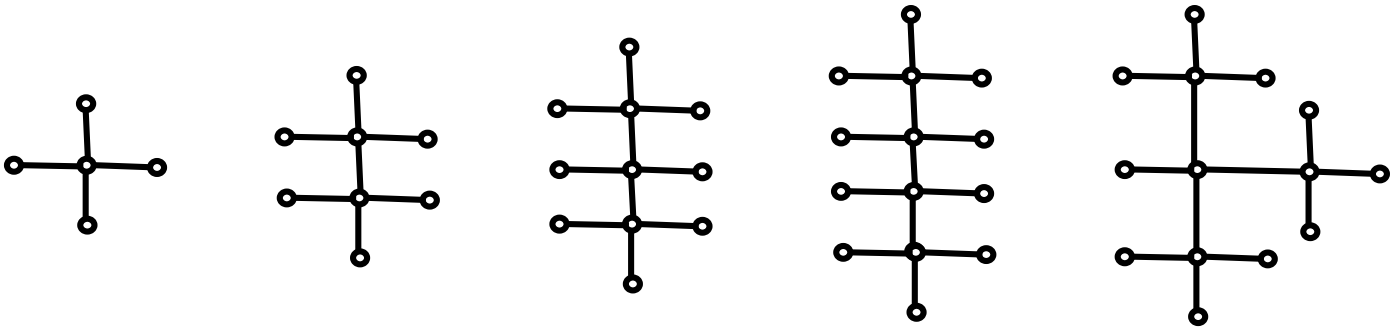


右图为一棵12阶树.

注:本章中所讨论的回路均
指简单回路或初级回路

树的应用

英国数学家凯莱(Arthur Cayley)于19世纪中叶研究饱和碳氢化合物C_nH_{2n+2}的同分异构体时提出树的概念. 当n=1,2,3时, 都只有一棵非同构的树; 当n=4时, 有2棵不同构的树.



甲烷　　乙烷　　丙烷　　丁烷　　异丁烷

无向树的性质

定理 设$G=<V,E>$是n阶m条边的无向图, 则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惟一的路径;

(3) G中无回路且$m=n-1$;

(4) G是连通的且$m=n-1$;

(5) G是连通的且G中任何边均为桥;

(6) G中没有回路, 但在任何两个不同的顶点之间加一条新边后所得图中有惟一的一个含新边的圈.

无向树的性质(续)

定理 设T是n阶非平凡的无向树，则T中至少有两片树叶.

证 设T有x片树叶, 由握手定理及前面的定理，

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

解得 $x\geq 2$.

例题

例1 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.

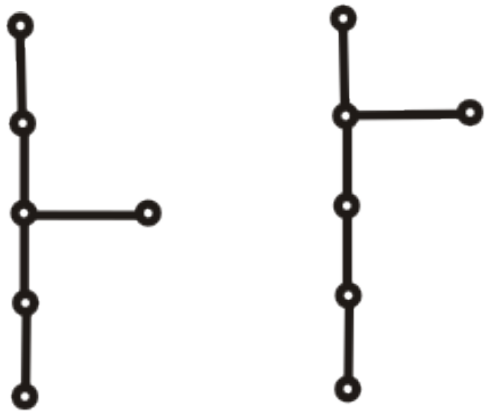
解 用树的性质$m=n-1$和握手定理.

设有x片树叶，于是

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

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

解得$x=3$，故T有3片树叶.



T的度数列为1, 1, 1, 2, 2, 3

有2棵非同构的无向树.

例题

例2 已知无向树T有5片树叶, 2度与3度顶点各1个, 其余顶点的度数均为4. 求T的阶数n, 并画出满足要求的所有非同构的无向树.

解 设T的阶数为n, 则边数为$n-1$, 4度顶点的个数为$n-7$.

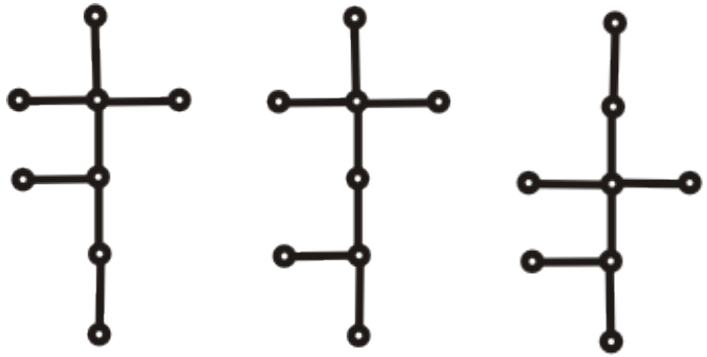
由握手定理得

$$2m=2(n-1)=5\times1+2\times1+3\times1+4(n-7)$$

解得$n=8$, 4度顶点为1个.

T的度数列为1,1,1,1,1,2,3,4

有3棵非同构的无向树



生成树

设G为无向连通图

G的生成树: G的生成子图并且是树

生成树T的树枝: G在T中的边

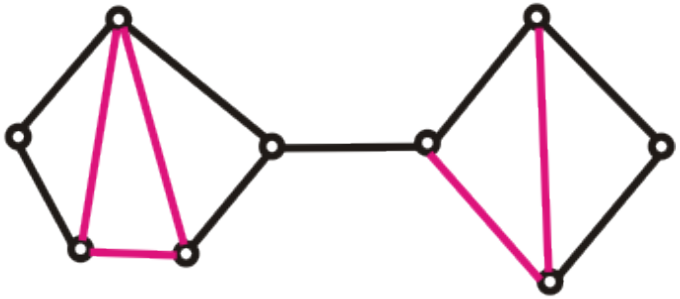
生成树T的弦: G不在T中的边

生成树T的余树 $\overline{T}$: 所有弦的集合的导出子图

注意：$\overline{T}$不一定连通, 也不一定不含回路.

黑边构成生成树

红边构成余树



生成树的存在性

定理 任何无向连通图都有生成树.

证 用破圈法. 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, 这不破坏连通性, 重复进行直到无圈为止,剩下的图是一棵生成树.

推论 设n阶无向连通图有m条边, 则$m \geq n-1$.

基本回路与基本回路系统

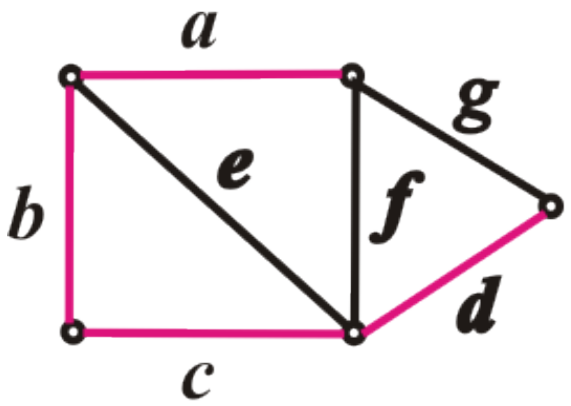
定义 设T是连通图G的一棵生成树，对每一条弦e，存在惟一的由弦e和T的树枝构成的初级回路C_e，称C_e为对应于弦e的**基本回路**. 称所有基本回路的集合为对应生成树T的**基本回路系统**.

求基本回路的算法: 设弦$e=(u,v)$, 先求T中u到v的路径Γ_{uv}, 再加上弦e.

例如

$C_e=e\ b\ c$, $C_f=f\ a\ b\ c$, $C_g=g\ a\ b\ c\ d$,

$C_{基}=\{C_e, C_f, C_g\}$.



基本割集与基本割集系统

定义 设T是连通图G的一棵生成树, 对T的每一条树枝a, 存在惟一的由树枝a, 其余的边都是弦的割集S_a, 称S_a为对应树枝a的**基本割集**, 称所有基本割集的集合为对应生成树T的**基本割集系统**.

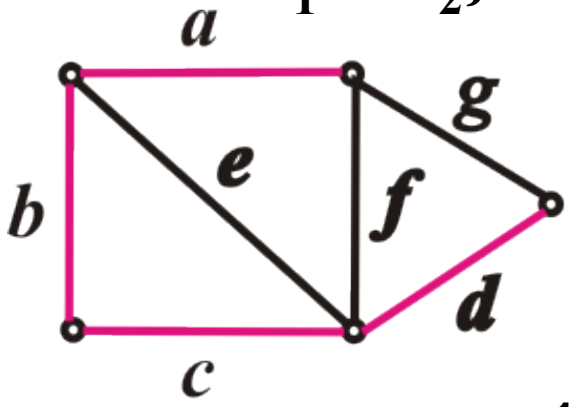
求基本割集的算法: 设a为生成树T的树枝, $T-a$由两棵子树T_1与T_2组成, 则

$S_a=\{e \mid e\in E(G)$且e的两个端点分别属于T_1与$T_2\}$.

例如 $S_a=\{a, f, g\}$, $S_b=\{b, e, f, g\}$,

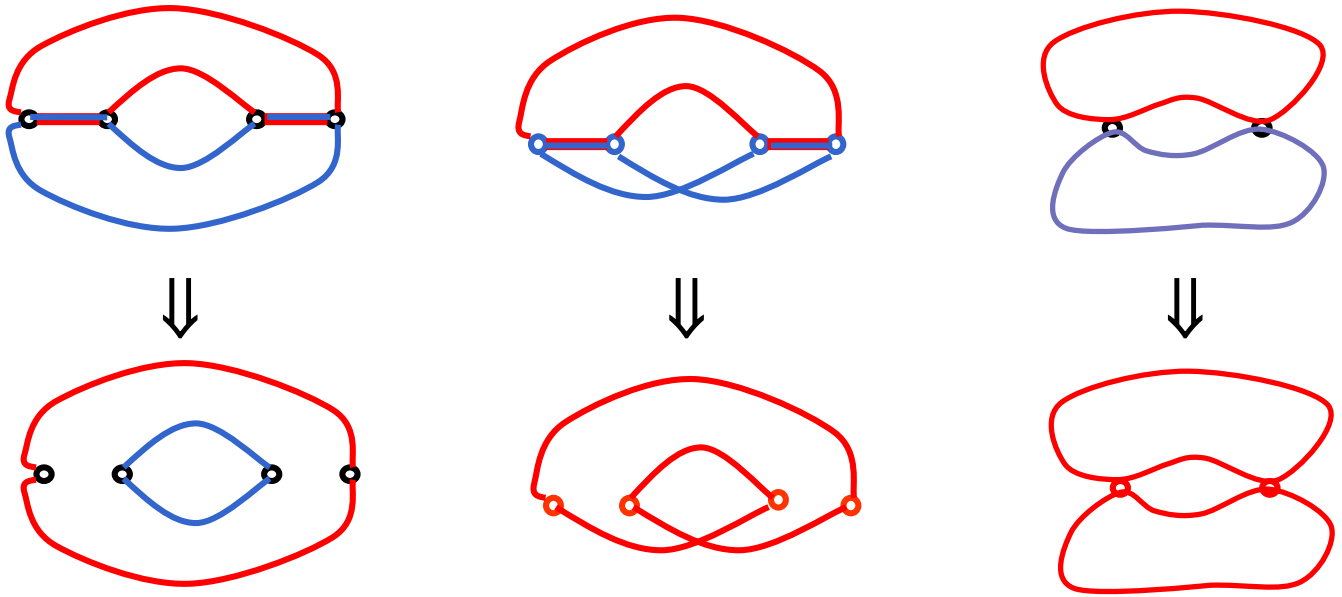
$S_c=\{c, e, f\ g\}$, $S_d=\{d, g\}$,

$S_{基}=\{S_a, S_b, S_c, S_d\}$.



回路合并

合并回路C_1和C_2($C_1 \oplus C_2$): $C_1 \oplus C_2$是C_1和C_2上的边的对称差构成的(一条或几条)回路.



基本回路的性质

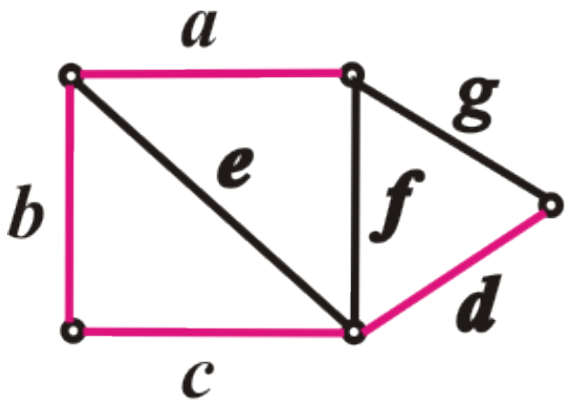
连通图中的任一条回路都可以表成对应它所含弦的基本回路的合并.

例如，$abcf=C_f$

$aef=C_e\oplus C_f$

$aedg=C_e\oplus Cg$

$bcdgfe=C_e\oplus C_f\oplus Cg$



基本割集的性质

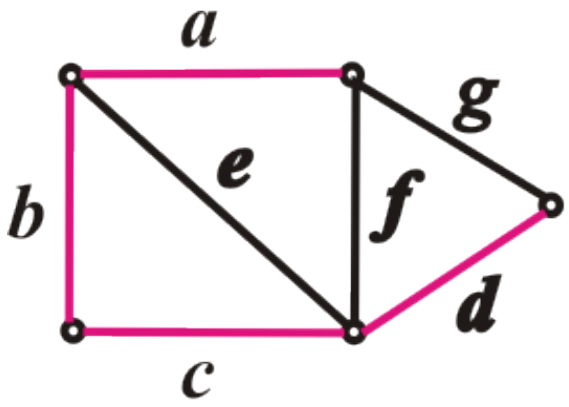
连通图中的任一割集都可以表成对应它所含树枝的基本割集的对称差.

例如 $\{g,d\}=S_d$

$\{a,b,e\}=S_a \oplus S_b$

$\{a,e,c\}=S_a \oplus S_c$

$\{b,e,f,d\}=S_b \oplus S_d$



无向图与最小生成树

对无向图或有向图的每一条边e附加一个实数$w(e)$, 称作边e的权. 图连同附加在边上的权称作带权图, 记作$G=<V,E,W>$.
设T是G的生成树, T所有边的权的和称作T的权, 记作$W(T)$.

最小生成树: 带权图权最小的生成树

避圈法 (Kruskal) ——求最小生成树的算法

设G是n阶无向连通带权图G.

(1) 按权从小到大排列边(环除外), 设$W(e_1)\le W(e_2)\le\ldots\le W(e_m)$.

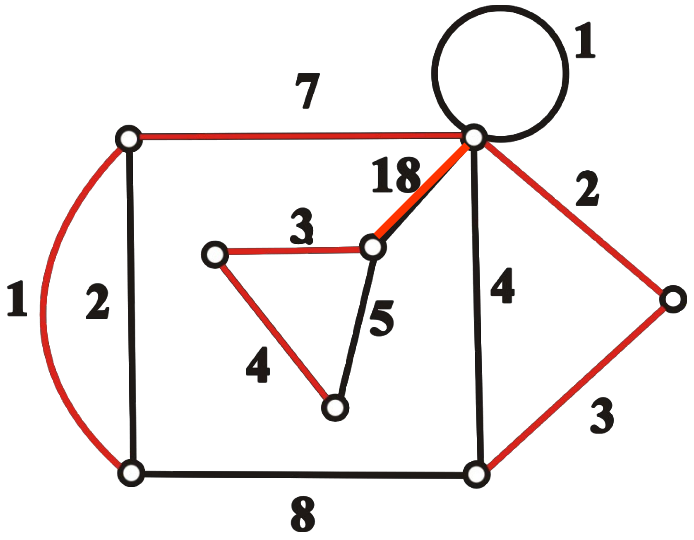
(2) 令$T\leftarrow\varnothing$, $i\leftarrow1$, $k\leftarrow0$.

(3) 若e_i与T中的边不构成回路，则令$T\leftarrow T\cup\{e_i\}$, $k\leftarrow k+1$.

(4) 若$k<n-1$, 则令$i\leftarrow i+1$, 转(3).

实例

例 求图的一棵最小生成树



w(T)=38

7.2 根树及其应用

- 有向树与根树
- 家族树与根子树
- 有序树
- 根树与有序树的分类

  *r*叉(有序)树, *r*叉正则(有序)树,

  *r*叉完全正则(有序) 树
- 最优2叉树与Huffman算法
- 前缀码与最佳前缀码
- 中序行遍法、前序行遍法、后序行遍法
- 波兰符号法与逆波兰符号法

有向树与根树

有向树: 基图为无向树的有向图

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

根树(续)

根树的画法:树根放上方, 省去所有有向边上的箭头

如右图所示

*a*是树根

*b,e,f,h,i*是树叶

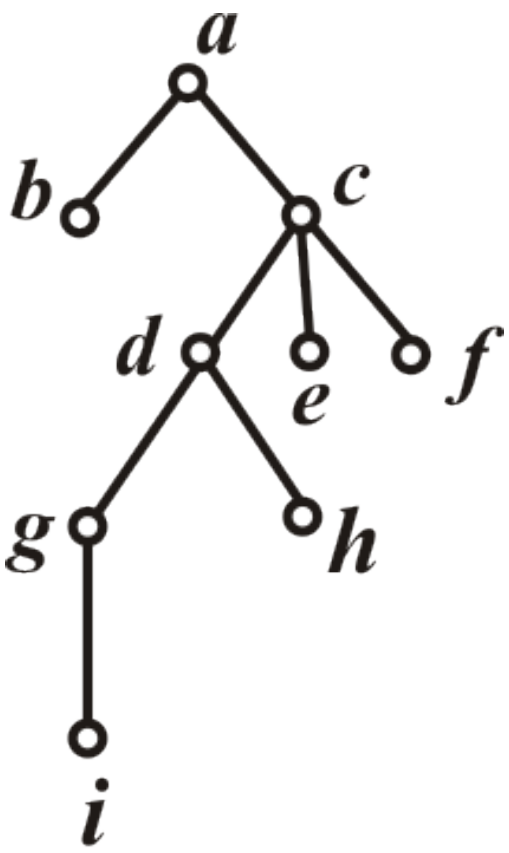
*c,d,g*是内点

*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

定义 把根树看作一棵家族树:

(1) 若顶点 a 邻接到顶点 b, 则称 b 是 a 的儿子, a 是 b 的父亲;

(2) 若b和c为同一个顶点的儿子, 则称b和c是兄弟;

(3) 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.

根树的分类

有序树: 将根树同层上的顶点规定次序

*r*叉树:根树的每个分支点至多有*r*个儿子

*r*叉正则树: 根树的每个分支点恰有*r*个儿子

*r*叉完全正则树: 树叶层数相同的*r*元正则树

*r*叉有序树: 有序的*r*叉树

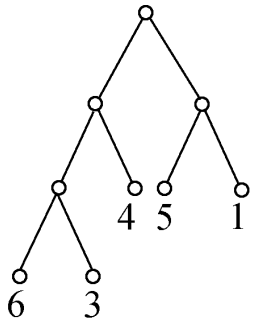
*r*叉正则有序树: 有序的*r*叉正则树

*r*叉完全正则有序树: 有序的*r*叉完全正则树

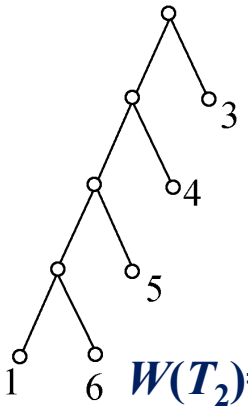
最优2叉树

定义 设2叉树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, 其中$l(v_i)$是$v_i$的层数. 在所有权为$w_1, w_2, \ldots, w_t$的$t$片树叶的2叉树中, 权最小的2叉树称为最优 2叉树.

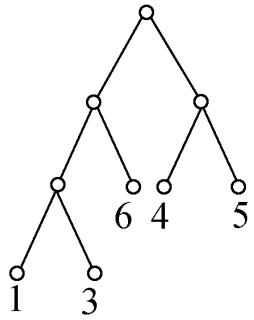
例如



$W(T_1)=47$



$W(T_2)=54$



$W(T_3)=42$

求最优2叉树的算法

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.

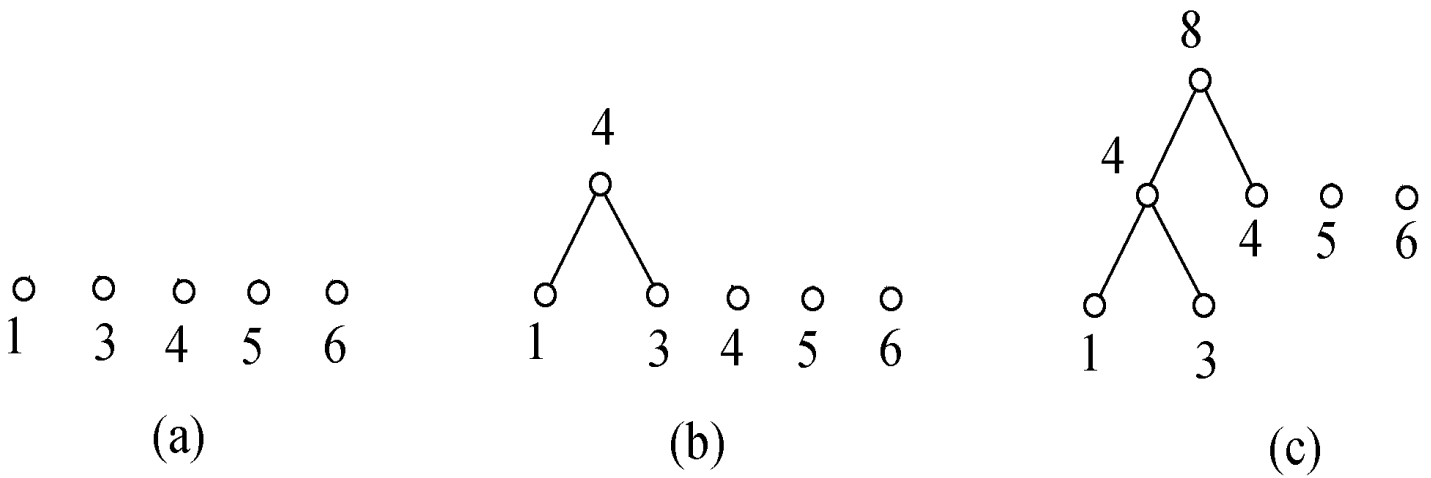
② 在所有入度为0的顶点(不一定是树叶)中选出两个权最小的顶点, 添加一个新分支点, 以这2个顶点为儿子, 其权等于这2个儿子的权之和.

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

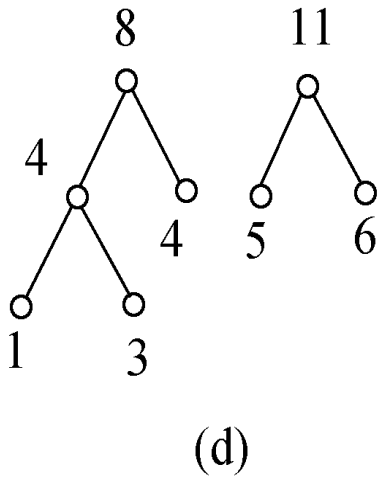
实例

例 求权为 1, 3, 4, 5, 6的最优树.

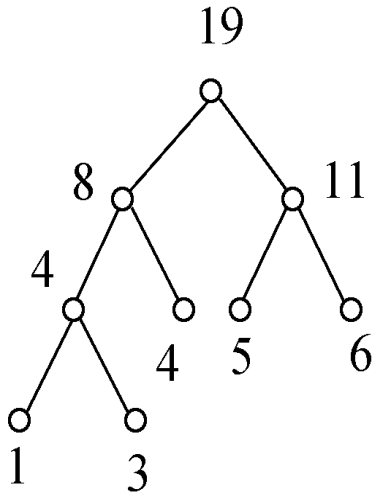


实例

例(续)



(d)



(e)

$W(T)=42$, 前面的T_3也是最优的.

前缀码

设$\alpha=\alpha_1\alpha_2\ldots\alpha_{n-1}\alpha_n$是长度为$n$的符号串

α的前缀: $\alpha_1\alpha_2\ldots\alpha_k$, k=1,2,…,n-1,n

前缀码: $\{\beta_1, \beta_2, \ldots, \beta_m\}$, 其中$\beta_1, \beta_2, \ldots, \beta_m$为非空字符串, 且任何两个互不为前缀

2元前缀码: 只有两个符号(如0与1)的前缀码

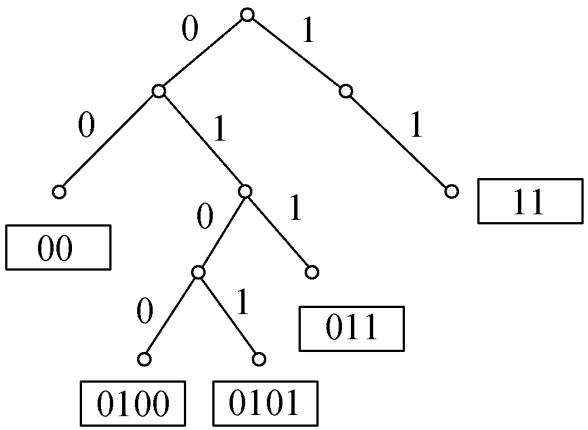
如 {0,10,110, 1111}, {10,01,001,110}是2元前缀码

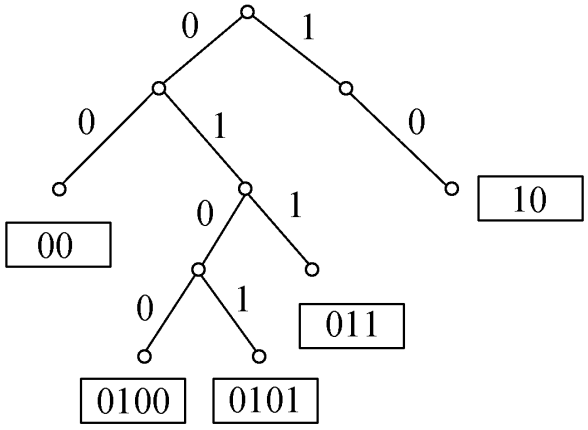
{0,10,010, 1010} 不是前缀码

前缀码(续)

一棵2叉树产生一个二元前缀码:
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





最佳前缀码

设要传输的电文中含有t个字符, 字符a_i出现的频率为p_i, 它的编码的长度为l_i, 那么100个字符的电文的编码的期望长度是$100\sum_{i=1}^{t} l_i p_i$. 称编码期望长度最小的2元前缀码为最佳2元前缀码.

在用2叉树产生2元前缀码时, 每个二进制串的长度等于它所在树叶的深度, 因而权为$100p_1$, $100p_2$, …, $100p_t$的最优2叉树产生的2元前缀码是最佳2元前缀码. 于是, 给定字符出现的频率, 可以用Huffman算法产生最佳2元前缀码.

实例

例 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 , 并求传输$10^n(n\geq 2)$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叉树. 这里 $w_1=5, w_2=5, w_3=10, w_4=10, w_5=10, w_6=15, w_7=20, w_8=25.$

例(续)

编码:

0---01

1---11

2---001

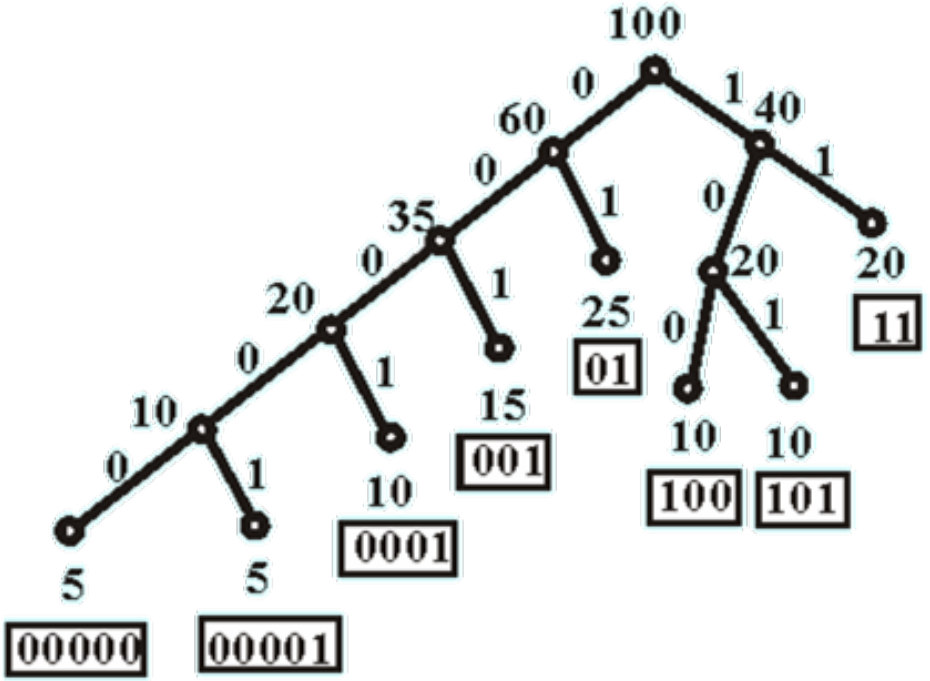
3---100

4---101

5---0001

6---00000

7---00001



传100个按比例出现的八进制数字所需二进制数字的个数为 $W(T)=285$.

传$10^n(n\geq 2)$个所用二进制数字的个数为2.85×10^n, 而用等长码(长为3)需要用 3×10^n个数字.

行遍2叉有序树

行遍(周游)根树T：对T的每个顶点访问且仅访问一次.

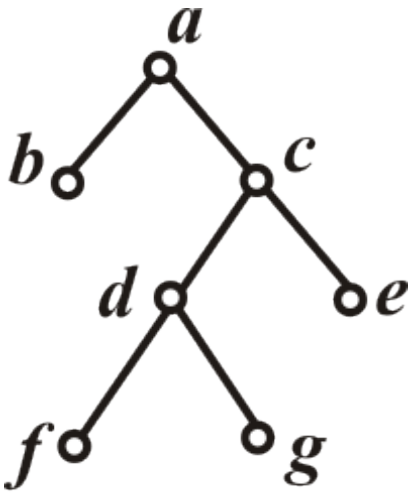
行遍2叉有序树的方式：

① 中序行遍法: 左子树、根、右子树

② 前序行遍法: 根、左子树、右子树

③ 后序行遍法: 左子树、右子树、根

当不是正则树时, 左子树或右子树可缺省



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

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

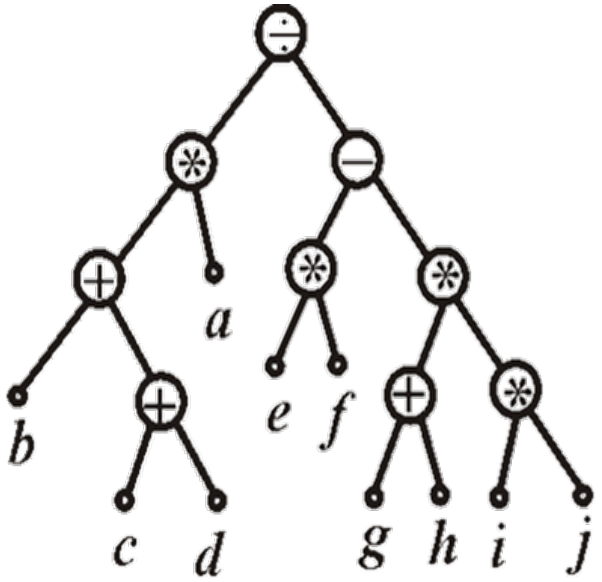
后序行遍: $b\ ((f\ g\ \underline{d})\ e\ \underline{c})\ \underline{a}$

用2叉有序树表示算式

每一个分支点放一个运算符. 二元运算符所在的分支点有2个儿子, 运算对象是以这2个儿子为根的根子树表示的子表达式, 并规定被减数和被除数放在左子树上; 一元运算符所在的分支点只有一个儿子, 运算对象是以这个儿子为根的根子树表示的子表达式.数字和变量放在树叶上.

实例

例1 表示((*b*+(*c*+*d*))**a*)÷((*e***f*)–(*g*+*h*)*(*i***j*))的2叉有序树



中序行遍:

((*b*+(*c*+*d*))**a*)÷((*e***f*)–(*g*+*h*)*(*i***j*))

前序行遍:

÷(*(+*b*(+*cd*))*a*)(–(**ef*)(*(+*gh*)(**ij*)))

后序行遍:

((*b*(*cd*+)+)*a**)((*ef**)((*gh*+)(*ij**)*)–)÷

注:中序行遍的结果是原式

波兰符号法

波兰符号法(前缀符号法): 按前序行遍法访问表示算式的2叉有序树, 并舍去所有括号.

例1(续) ÷*+*b*+*cda*–**ef***+*gh***ij*

计算方法: 从左到右, 每个运算符号对它后面紧邻的2个(或1个)数进行运算.

例1(续) 设*a*=3, *b*=1, *c*=*d*=2, *e*=*f*=3, *g*=*i*=1, *h*=*j*=2.

÷*+1+223–*33*+12*12, ÷*+143–*33*+12*12

÷*53–*33*+12*12, ÷(15)–*33*+12*12

÷(15)–9*+12*12, ÷(15)–9*3*12

÷(15)–9*32, ÷(15)–96, ÷(15)3, 5

逆波兰符号法

逆波兰符号法(后缀符号法)：按后序行遍法访问表示算式的2叉有序树, 并舍去所有括号.

例1(续)　*bcd++a*ef*gh+ij**– ÷*

计算方法：从右到左，每个运算符号对它前面紧邻的2个(或1个)数进行运算.

例1(续)　122++3*33*12+12**– ÷

122++3*33*12+2*– ÷ ,　122++3*33*32*– ÷

122++3*33*6– ÷,　122++3*96– ÷

122++3*3 ÷,　14+3*3 ÷,　53*3 ÷,　(15)3 ÷,　5

实例

例2 用2叉有序树表示下述命题公式, 并写出它的波兰符号法和逆波兰符号法表达式.

$$(p\vee\neg q)\rightarrow((\neg p\wedge r)\rightarrow(q\vee r))$$

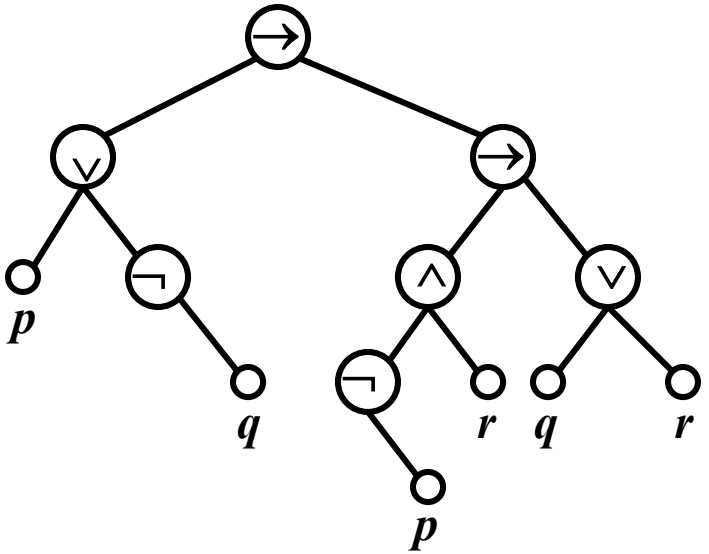
解

波兰符号法表达式

$$\rightarrow\vee p\neg q\rightarrow\wedge\neg pr\vee qr$$

逆波兰符号法表达式

$$pq\neg\vee p\neg r\wedge qr\vee\rightarrow\rightarrow$$



注: 当一元运算符在运算对象前面时, 应画成右儿子.