离散数学

图论

7.2 根树及其应用

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

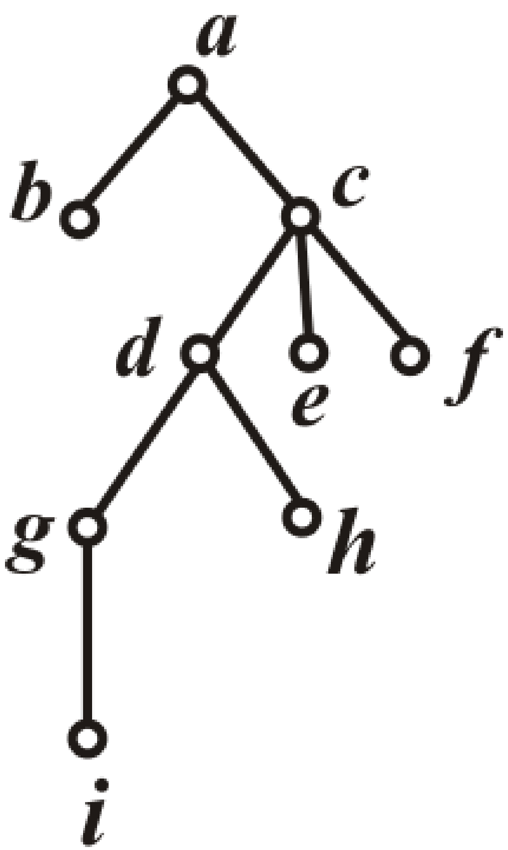
c,d,g是内点

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

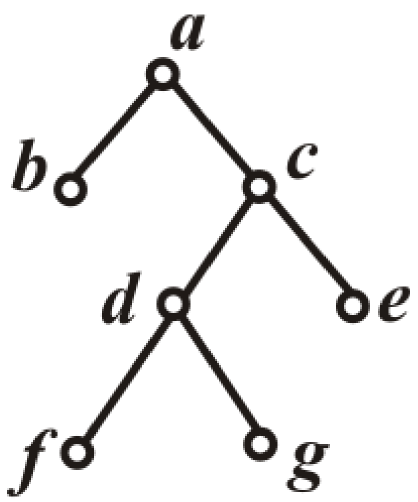
r叉完全正则有序树: 有序的r叉完全正则树

根树的分类

- **定理** 若T是完全r叉树，其叶子数为t，分枝点数为i，则$(m-1)i=t-1$
- 由握手定理，所有顶点的入度和等于出度和。除了根每个顶点的入度为1，所以所有顶点的入度和为$t+i-1$。所有顶点的出度和为$m*i$。整理可得结论

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

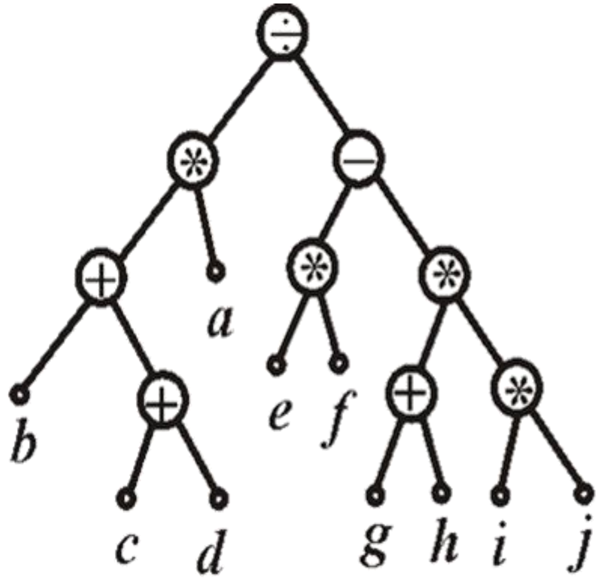
后序行遍: $b\ ((f\ g\ \underline{d})\ e\ \underline{c})\ \underline{a}$

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$((b+(c+d))*a)\div((e*f)-(g+h)*(i*j))$的2叉有序树



中序行遍:

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

前序行遍:

$\div(*(+b(+cd))a)(-(*ef)(*(+gh)(*ij)))$

后序行遍:

$((b(cd+)+)a*)((ef*)((gh+)(ij*)*)-)\div$

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

- 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.

$\div**+1\underline{+22}3-*33*+12*12$, $\div*\underline{+14}3-*33*+12*12$

$\div\underline{*53}-*33*+12*12$, $\div(15)-\underline{*33}*+12*12$

$\div(15)-9*\underline{+12}*12$, $\div(15)-9*3\underline{*12}$

$\div(15)-9\underline{*32}$, $\div(15)\underline{-96}$, $\underline{\div(15)3}$, 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) $bcd++a*ef*gh+ij**-\div$

- 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) 设a=3, b=1, c=d=2, e=f=3, g=i=1, h=j=2.

$122++3*33*12+\underline{12*}*-\div$

$122++3*33*\underline{12+}2*-\div$,　　$122++3*33*\underline{32*}-\div$

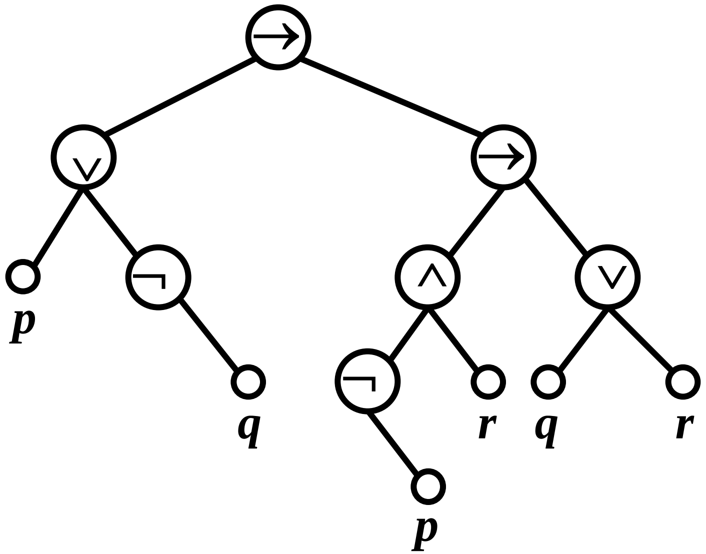
$122++3*\underline{33*}6-\div$,　　$122++3*\underline{96-}\div$

$1\underline{22+}+3*3\div$,　　$\underline{14+}3*3\div$,　　$\underline{53*}3\div$,　　$\underline{(15)3\div}$,　　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.

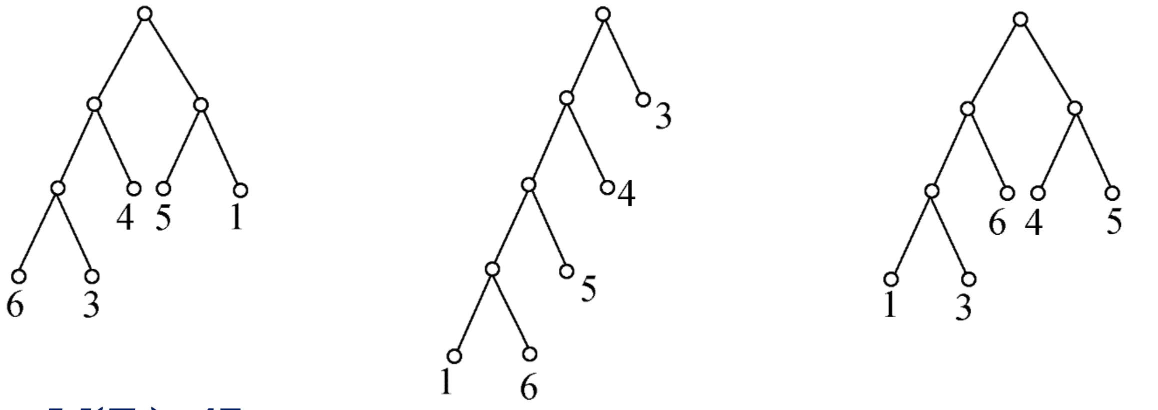
练习

- 计算- + * 4 2 1 * / 6 3 2的值，并画出对应的有序树T，最后写出其对应的后序遍历结果。

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.

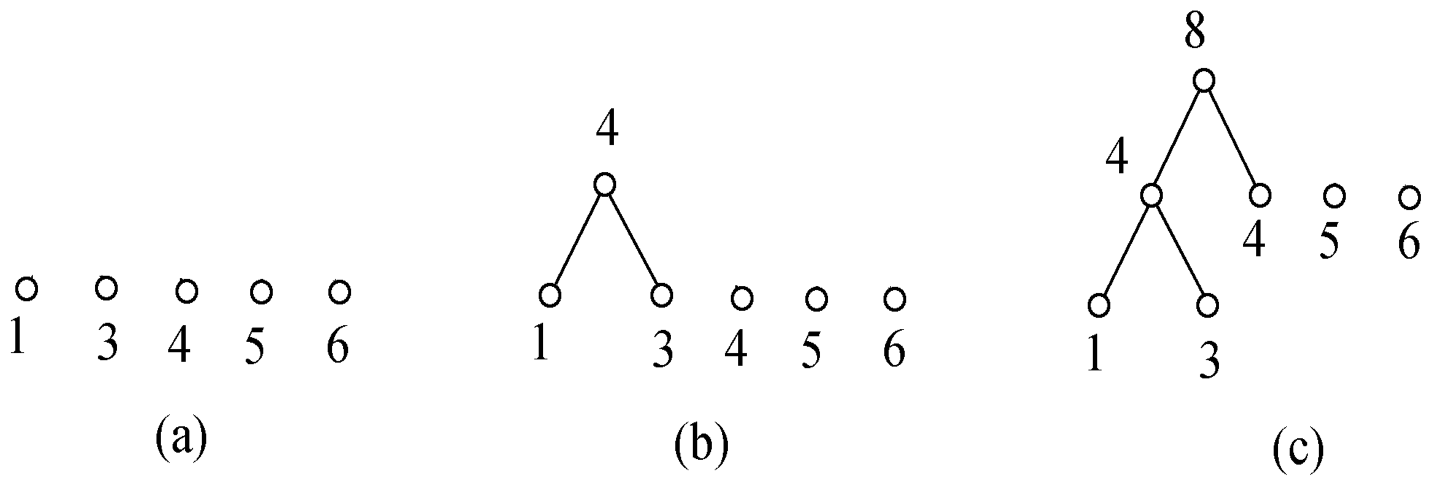
② 在所有入度为0的顶点(不一定是树叶)中选出两个权最小的顶点, 添加一个新分支点, 以这2个顶点为儿子, 其权等于这2个儿子的权之和.

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

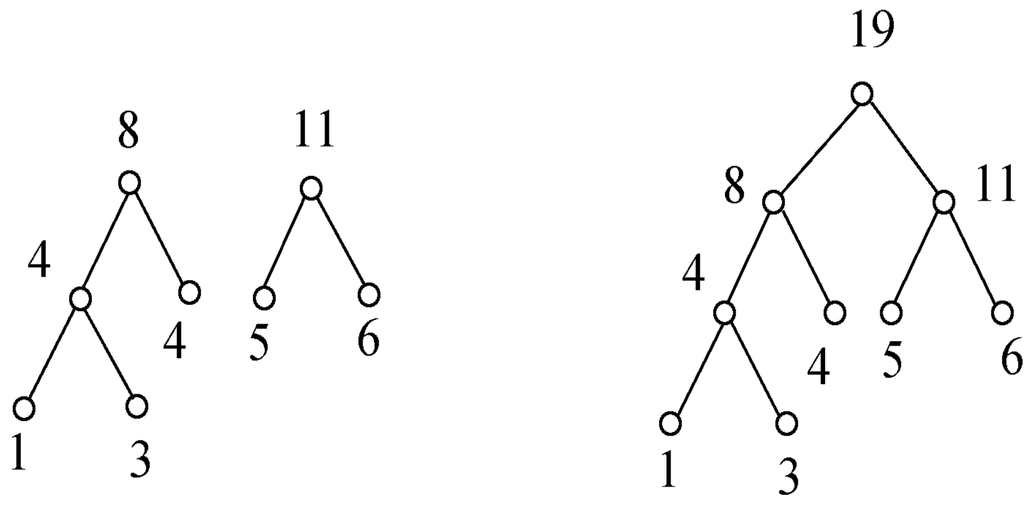
实例

- 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

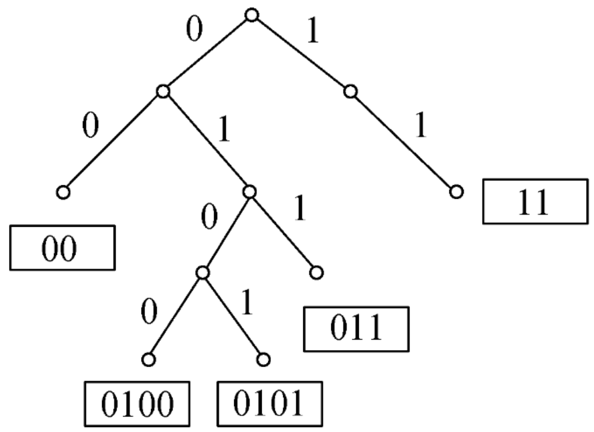
如 {0,10,110, 1111}, {10,01,001,110}是2元前缀码

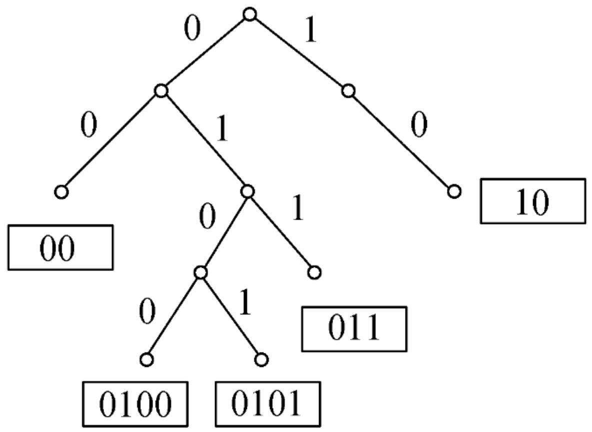
{0,10,010, 1010} 不是前缀码

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≥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

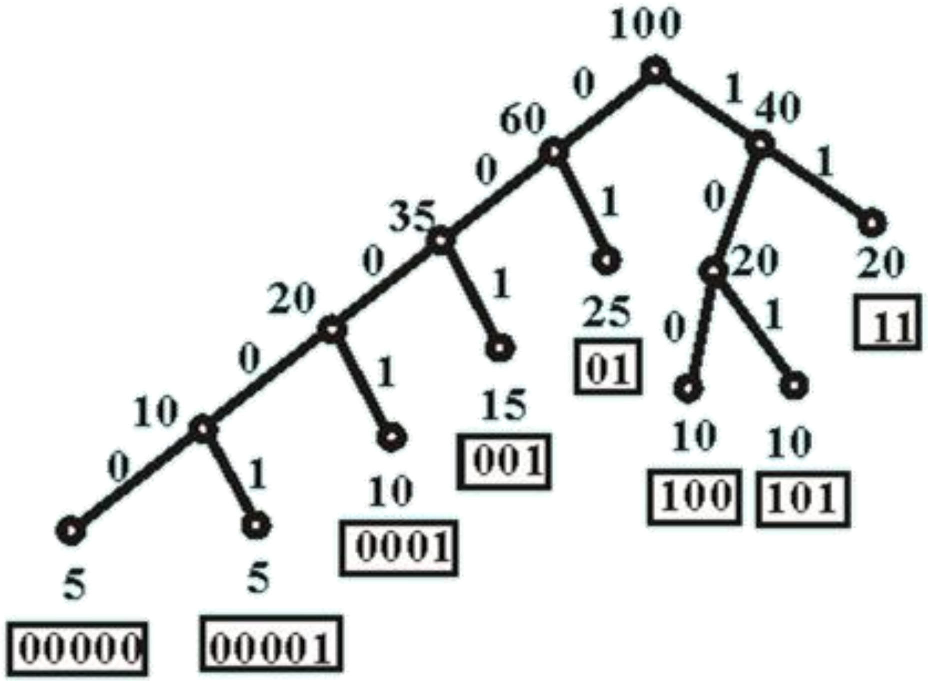
3---100

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≥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.

练习

- 利用霍夫曼算法对“a banana”进行编码，并画出对应的最优二叉树。

作业

- P172
- 7.12
- 7.14
- 7.15

问题？