集合论与图论11

树及其应用

何英华
hyh@tju.edu.cn

第7章 树及其应用

- 无向树
- 根树及其应用

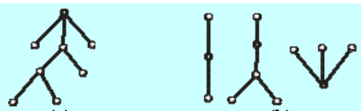
7.1 无向树

- 无向树的定义及其性质
- 生成树与基本回路和基本割集
- 最小生成树

无向树的定义

无向树: 连通无回路的无向图，常用T表示
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

例如



(a) (b)

无向树的性质

定理7.1 设$G=<V,E>$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惟一的路径;
(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惟一的一条初级回路.
(6) G是连通的且G中任意一条边均为桥.

定理7.1的证明

(1)⇒(2) 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.

(2)⇒(3) 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$n=k+1$. 任取一条边$e=(u,v)$, 它是u,v之间惟一的通路, 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, $n_1\le k$, $n_2\le k$.
由归纳假设, $m_1=n_1-1$, $m_2=n_2-1$, 得$m= m_1+m_2+1= n-1$.

定理7.1的证明(续)

(3)⇒(4) 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, 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.

(4)⇒(1) 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), $m_k= n_k-1$. 得到$m= n-p$, 矛盾.

定理7.1的证明(续)

(1)⇒(5) 由(1)⇒(2), 任意2个不相邻的顶点之间有一条惟一的路径, 故在这2个顶点之间添加一条新边, 必得到一条惟一的初级回路.

(5)⇒(6) 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, 若删去一条边G仍是连通的, 这条边必在一条回路上, 与G中无回路矛盾.

(6)⇒(1) G中无回路, 否则删去回路上任意条边, G仍连通.

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 设有$n(n>1)$个顶点, x片树叶, 由握手定理和定理7.1, 有

$$2(n-1)=\sum d(v_i)\geq x+2(n-x)$$

解得 $x\geq2$.

实例

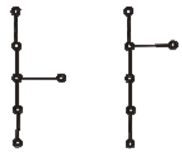
例1 已知无向树T中, 有1个3度顶点, 2个2度顶点, 其余顶点全是树叶. 试求树叶数, 并画出满足要求的非同构的无向树.

解 设有x片树叶,

$2\times(3+x-1)=1\times3+2\times2+x$

解得$x=3$，故T有3片树叶.

T的度数列为1, 1, 1, 2, 2, 3



实例

例2 画出所有6阶非同构的无向树

解 5条边, 总度数等于10

可能的度数列:

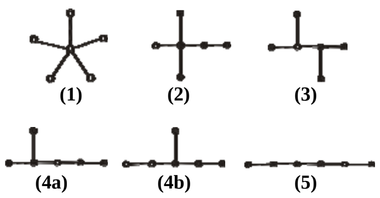
(1) 1,1,1,1,1,5

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

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

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

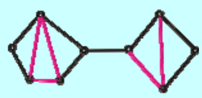
(5) 1,1,2,2,2,2



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

例如 图中黑边构成生成树
红边构成余树



注意: 余树一般不是树

生成树的存在性

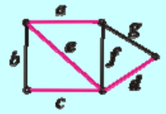
定理7.3 任何无向连通图都有生成树.
证 用破圈法. 若图中无圈, 则图本身就是自己的生成树. 否则删去圈上的任一条边, 不破坏连通性, 重复进行直到无圈为止, 得到图的一棵生成树.

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.

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

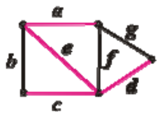
例3 图中红边为一棵生成树,
对应它的基本回路系统为
$\{bce, fae, gaed\}$



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, $e_1', e_2', \ldots, e'_{n-1}$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$\{S_1, S_2, \ldots, S_{n-1}\}$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
$\{\{a,f,g\}, \{e,b,f,g\}, \{c,b\}, \{d,g\}\}$



最小生成树

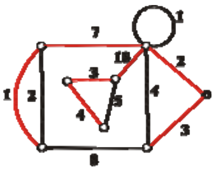
图G的每一条边e附加一个实数$w(e)$, 称作边e的权. 图G连同附加在边上的权称作带权图, 记作$G=<V,E,W>$. 设H是G的子图, H所有边的权的和称作H的权, 记作$W(H)$.

最小生成树: 带权图中权最小的生成树

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

实例

例5 求图的一棵最小生成树



$W(T)$=38

7.2 根树及其应用

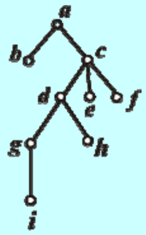
- 根树及其分类
- 最优树与哈夫曼算法
- 最佳前缀码
- 根树的周游及其应用
  - 中序行遍法、前序行遍法和后序行遍法
  - 波兰符号法与逆波兰符号法

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

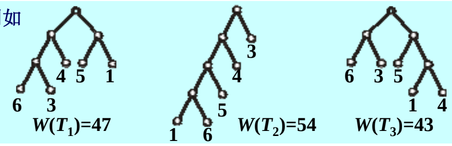
根树的分类

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



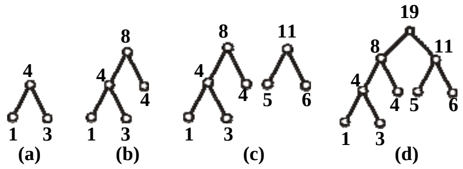
求最优2元树的算法

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



(a) (b) (c) (d)

$W(T)$=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,…,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

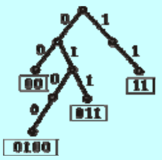
例如 { 0, 10, 110, 1111 }, { 10, 01, 001, 110 }是2元前缀码

{ 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:

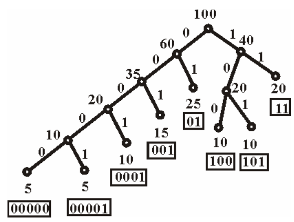
0: 25, 1: 20, 2: 15, 3: 10, 4: 10, 5: 10, 6: 5, 7: 5

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

对一棵根树的每个顶点访问一次且仅访问一次称作对根树的行遍或周游

行遍2元有序正则树的方式:

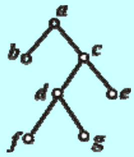
① 中序行遍法: 左子树、树根、右子树

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

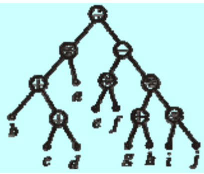
后序行遍: $b\ ((f\ g\ \underline{d})\ e\ \underline{c})\ \underline{a}$

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, 其结果不加括号, 规定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, 其结果不加括号,规定每个运算符与前面紧邻两数运算.

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