树

李修成

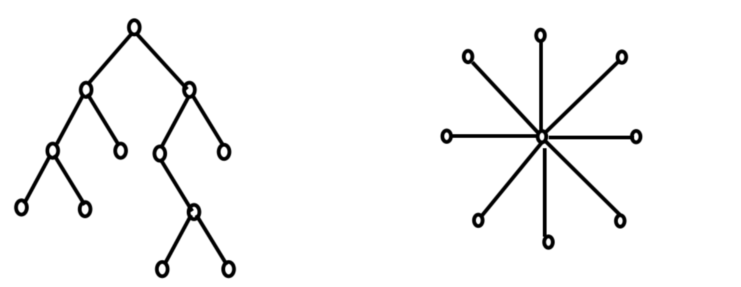
计算机科学与技术学院

Outline

无向树及其性质

定义 *1.1* (无向树). 连通无回路的无向图称为无向树，简称树. 每个连通分支都是树的无向图称为森林. 平凡图称为平凡树. 在无向树中，悬挂顶点称为树叶，度数大于或等于 2 的顶点称为分支点.

例子 *1.1*.



引理 *1.1.* 令 G 为 n 阶 m 条边的无向连通图，则 $m \geq n-1$.

证明. 使用归纳法证明. $n=1$ 时，平凡图 $m \geq 0$，结论成立.

假设结论对 $n=k(k \geq 1)$ 成立，现证对 $n=k+1$ 成立. 从 G 中去掉任意顶点，令所得连通分支量为 $G_1, G_2, \ldots, G_s$, $1 \leq s \leq k$，每个连通分量顶点数和边数分别记为 $n_i, m_i, 1 \leq i \leq s$. 则 $m \geq \sum_{i=1}^{s} m_i + s$，因为产生 s 个连通分量至少需要去掉 s 条边（考虑其逆操作，将 s 个连通分量经一个顶点连到一起）. 又由归纳假设知 $m_i \geq n_i - 1$. 故，

$$m \geq \sum_{i=1}^{s} m_i + s \geq \sum_{i=1}^{s} (n_i - 1) + s = \sum_{i=1}^{s} n_i = n - 1.$$

无向树的性质

定理 ***1.1.*** 设 $G = \langle V, E\rangle$ 是 n 阶 m 条边的无向图，则下面各命题是等价的：

1. G 是树.
2. G 中任意两个顶点之间存在唯一的路径.
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唯一的一个含新边的圈.

证明.

- $(1) \Rightarrow (2)$. 若路径不唯一，必有回路.
- $(2) \Rightarrow (3)$. 若 G 中有回路，则回路上任意两点之间的路径不唯一. 对 n 用归纳法证明 $m = n - 1$.

  当 $n = 1$ 时成立. 设 $n = k (k \geq 1)$ 时成立，现证 $n = k + 1$ 时也成立. 任取一条边 e，由于 G 中无回路，故 $G - e$ 有两个连通分支 G_1, G_2 且 $n_1 \leq k$, $n_2 \leq k$. 由归纳假设知 $m_i = n_i - 1, i = 1, 2$. 故,

$$m = m_1 + m_2 + 1 = n_1 + n_2 - 2 + 1 = n - 1.$$

- $(3) \Rightarrow (4)$. 只需证明 G 连通. 用反证法. 否则 G 有 $s (s \geq 2)$ 个连通分支, 它们都是树（连通无回路）. 于是，有 $m_i = n_i - 1$,

$$m = \sum_{i=1}^{s} m_i = \sum_{i=1}^{s} n_i - s = n - s, \quad (s \geq 2),$$

  这与 $m = n - 1$ 矛盾.

- $(4) \Rightarrow (5)$. 只需证明 G 中每条边都是桥. $\forall e \in E$ 均有 $|E(G-e)| = n-1-1 = n-2$. 由引理 1.1可知 $G-e$ 不连通，故 e 为桥.
- $(5) \Rightarrow (6)$. 由 (5) 知 G 为树（连通无回路）. 由 $(1) \Rightarrow (2)$ 知，$\forall u, v \in V, u \neq v$ 有 u 到 v 存在唯一路径，加新边 $\{u, v\}$ 可得唯一的圈.
- $(6) \Rightarrow (1)$. 只需证明 G 连通. 考虑任意两个顶点 u, v，由于在 u, v 之间加边可以产生圈，故在原图中 $u \sim v$，即 G 连通.

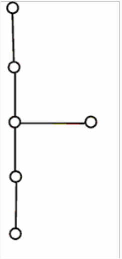
定理 *1.2.* 设 T 是 n 阶非平凡的无向树，则 T 中至少有两片树叶.

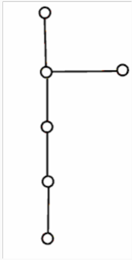
证明*.* 设 T 有 x 片树叶，由握手定理及定理1.1可知，

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

解的 $x \geq 2$.

例子 1.2. 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.





令 n 为阶，则树叶数量为 $n-3$ 且有

$$2m = 2(n-1) = 1\times 3 + 2\times 2 + n - 3$$

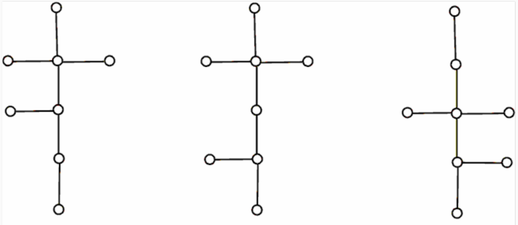
解的 $n=6$，故 T 有 3 片树叶.

例子 1.3. 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.

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,

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

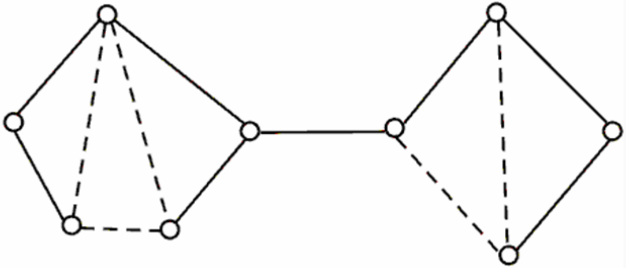
解的 $n=8$，4 度顶点为 1 个.



生成树

生成树

定义 2.1 (生成树 spanning tree). 如果无向图 G 的生成子图 T 是树，则称 T 是 G 的生成树. 设 T 是 G 的生成树，G 的在 T 中的边称为 T 的树枝，不在 T 中的边为 T 的弦. 称 T 的所有弦的导出子图为 T 的余树，记作 $\overline{T}$.



生成树存在条件

定理 *2.1.* 无向图 G 有生成树当且仅当 G 连通.

证明. （$\Leftarrow$）若 G 中无回路，则 G 为自己的生成树．若 G 中含圈，任取一圈，随意地删除圈上的一条边; 若仍有圈, 再任取一个圈并删去这个圈上的一条边，重复进行, 直到最后无圈为止. 最后得到的图无圈（当然无回路）、连通且是 G 的生成子图，因而是 G 的生成树.

推论 *2.1.* 令 G 为 n 阶 m 条边的无向连通图，则 $m \geq n - 1$.

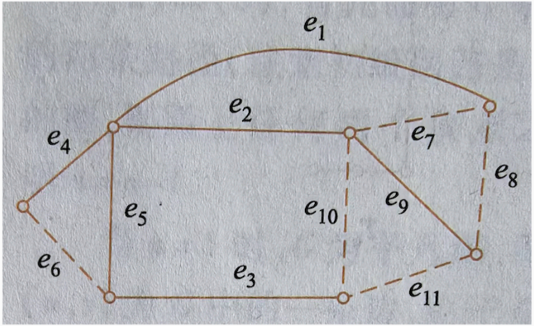
生成树

定理 ***2.2.*** 设 T 为无向连通图 G 中的一棵生成树，e 为 T 的任意一条弦，则 $T \cup e$ 中包含 G 中只有一条弦 e 且其余边均为树枝的圈，而且不同的弦对应的圈也不同.

证明. 设 $e = \{u, v\}$，由定理 1.1 可知，在 T 中，u, v 之间存在唯一的路径 P，则 $P \cup e$ 满足要求. 不同的弦对应的圈也不同.

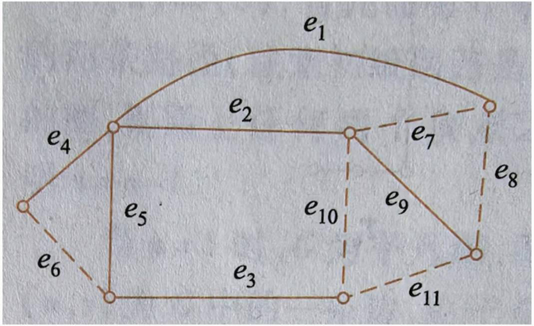
基本回路

定义 2.2. 设 T 是 n 阶 m 条边的无向连通图 G 的一棵生成树，设 $e_1', e_2', \cdots, e_{m-n+1}'$ 为 T 的弦，$C_r\,(r=1,2,\ldots,m-n+1)$ 为 T 添加弦 e_r' 产生的 G 中由弦 e_r' 和树枝构成的圈，称 C_r 为 G 的对应弦 e_r' 的基本回路或基本圈. 称 $\{C_1, C_2, \cdots, C_{m-n+1}\}$ 为 G 对应 T 的基本回路系统，称 $m-n+1$ 为 G 的圈秩，记作 $\xi(G)$.



定义 *2.3.* 无向图中的圈或若干个边不重复圈的并称作广义回路，规定 $\emptyset$ 为广义回路.

- 任一广义回路可以表示为基本回路的环和[1].



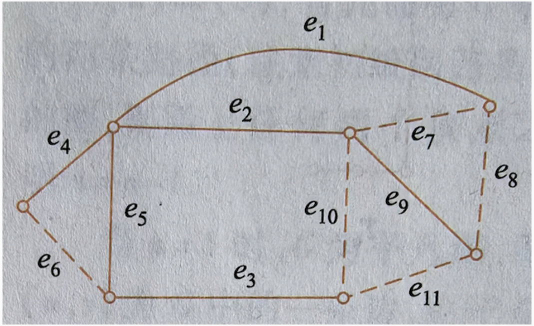
[1]以 $E_1 \oplus E_2$ 为边集，以 $E_1 \oplus E_2$ 所关联的顶点为顶点集，称为图 G_1 与 G_2 的环和，记为 $G_1 \oplus G_2$.

定理 *2.3.* 设 T 是连通图 G 的一棵生成树, e 为 T 的树枝, 则 G 中存在只含树枝 e，其余边都是弦的割集，且不同的树枝对应的割集也不同.

证明. 由定理 1.1 可知，e 是 T 的桥，因而 $T-e$ 有两个连通分支 T_1 和 T_2. 则 $S_e \triangleq E_G(V(T_1), V(T_2))$ 为 G 的割集，$e \in S_e$ 且 S_e 中除 e 外都是弦. 因为每个割集只含一条树枝, 故不同树枝对应的割集是不同的.

基本割集

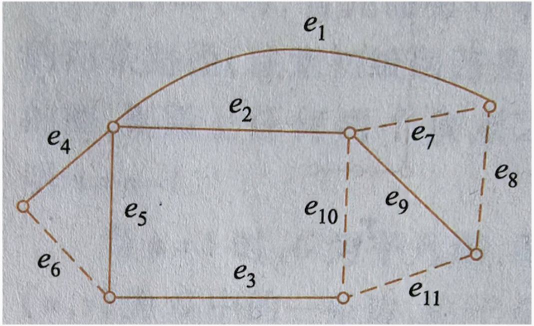
定义 2.4. 设 T 是 n 阶连通图 G 的一棵生成树，$e_1, e_2, \cdots, e_{n-1}$ 为 T 的树枝，S_i 是由树枝 e_i 和弦构成的割集 $(i = 1, 2, \cdots, n-1)$，则称 S_i 为 G 的对应树枝 e_i 的基本割集. 称集合 $\{S_1, S_2, \cdots, S_{n-1}\}$ 为 G 对应 T 的基本割集系统，称 $n-1$ 为 G 的割集秩，记作 $\eta(G)$.



广义割集

定义 2.5. 给定无向图 $G = \langle V, E\rangle$, $V_1 \subset V$, $V_1 \neq \emptyset$，记 $\overline{V_1} = V - V_1$，称边集 $E(V_1, \overline{V_1})$ 为广义割集，$E(V_1, \overline{V_1}) \triangleq \{\{u, v\} \mid \{u, v\} \in E \wedge u \in V_1 \wedge v \in \overline{V_1}\}$. 规定 $\emptyset$ 为广义割集.

- 割集是广义割集，但广义割集不一定是割集.
- 任一广义割集可以表示为基本割集的对称差.



广义回路空间

- 记无向图 G 的广义回路的全体为 C^*；C^* 对环和 [2] 运算具有封闭性.
- 定义 C^* 上的数乘运算：$0 \cdot C = \emptyset, 1 \cdot C = C$.
- C^* 对环和运算与数乘运算构成数域 $F = \{0, 1\}$ 上维度为 $m - n + 1$ 的向量空间 vector space，称为广义回路空间.
- 对任意圈 $C \in C^*$ 定义向量 $\mathbf{z}_C \in \{0, 1\}^{|E|}$

$$\mathbf{z}_C(e_i) = \begin{cases} 1 & \text{if } e_i \in C \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

- 任意基本回路系统为其一个基.

[2] 以 $E_1 \oplus E_2$ 为边集，以 $E_1 \oplus E_2$ 所关联的顶点为顶点集，称为图 G_1 与 G_2 的环和，记为 $G_1 \oplus G_2$.

广义割集空间

- 记 G 广义割集的全体为 S^*, 定义 S^* 上的数乘运算：$0 \cdot S = \emptyset, 1 \cdot S = S$.
- S^* 对对称差运算与数乘运算构成数域 $F = \{0, 1\}$ 上维度为 $n-1$ 的向量空间 vector space，称为广义割集空间.
- 对任意圈 $S \in S^*$ 定义向量 $\mathbf{z}_S \in \{0,1\}^{|E|}$

$$\mathbf{z}_S(e_i) = \begin{cases} 1 & \text{if } e_i \in S \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

- 任意基本割集系统为其一个基.

最小生成树

定义 2.6 (最小生成树 *minimum spanning tree*). 设无向连通带权图 $G = \langle V, E, W \rangle$，$T$ 是 G 的一棵生成树，T 的各边权重之和称为 T 的权重，记作 $W(T)$. G 的所有生成树中权最小的生成树称为 G 的最小生成树.

Kruskal 算法

Algorithm 1: Kruskal Algorithm

Input: A connected undirected weighted graph $G = \langle V, E, W \rangle$.

Output: A minimum spanning tree T of G.

$T \leftarrow \{\}$;
Sort the edges E by weight W in decreasing order;
for $\{u, v\} \in E$ **do**
  if no cycle in $T \cup \{u, v\}$ **then**
    $T \leftarrow T \cup \{u, v\}$;
return T;

Kruskal 算法

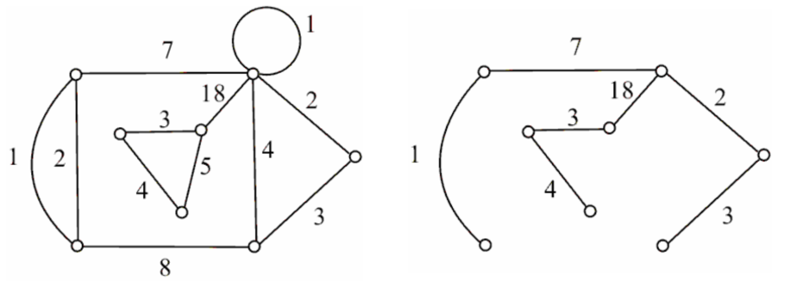
Algorithm 2: Kruskal Algorithm with Disjoint Set

Input: A connected undirected weighted graph $G = \langle V, E, W \rangle$.

Output: A minimum spanning tree T of G.

for $v \in V$ **do**
    makeset(v);
$T \leftarrow \{\}$;
Sort the edges E by weight W in decreasing order;
for $\{u, v\} \in E$ **do**
    if find(u) $\neq$ find(v) **then**
        add $\{u, v\}$ to T;
        union(u, v);
return T;

例子 2.1. 求图的一棵最小生成树.



$W(T) = 38$

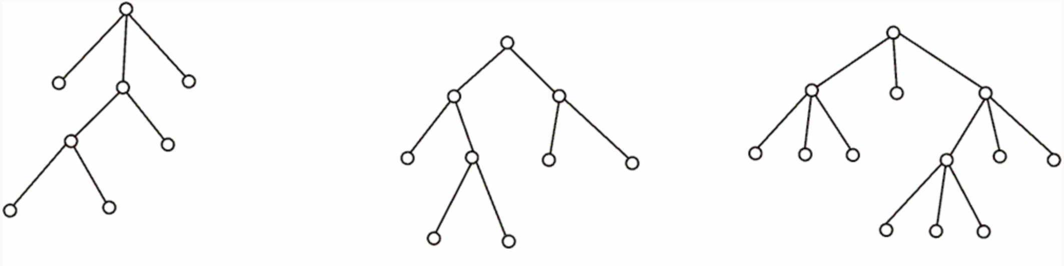
根树及其应用

根树及其应用

- 若有向图的基图是无向树, 则称这个有向图为有向树.
- 一个顶点的入度为 0，其余顶点的入度为 1 的有向树称为根树.
- 入度为 0 的顶点称为树根 root，入度为 1 出度为 0 的顶点称为树叶 leaf，入度为 1 出度不为 0 的顶点称为内点 internal nodes，内点和树根统称为分支点 branching nodes.
- 从树根到顶点 v 的路径的长度 (即, 路径中的边数) 称为 v 的层数.
- 所有顶点的最大层数称为树高.

根树的画法——树根放上方，省去所有有向边上的箭头.

例子 3.1.



家族树与根树的分类

定义 3.1. 设 T 为一棵非平凡的根树，

- $\forall v_i, v_j \in V(T)$, 若 v_i 可达 v_j, 则称 v_i 为 v_j 的祖先 ancestor，v_j 为 v_i 的后代 descendant.
- 若 v_i 邻接到 v_j, 则称 v_i 为 v_j 的父亲 parent，v_j 为 v_i 的儿子 child.
- 若 v_j, v_k 有相同的父亲 parent，则称 v_j 与 v_k 是兄弟 sibling.

家族树与根树的分类

将根树 T 中层数相同的顶点都标定次序，称 T 为有序树. 根树的分类:

1. 若 T 的每个分支点至多有 r 个儿子，则称 T 为r 叉树.
2. 若 T 的每个分支点都恰好有 r 个儿子, 则称 T 为r 叉正则树.
3. 若 T 是 r 叉正则树, 且所有树叶的层数相同, 则称 T 为r 叉完全正则树.

有序的 r 叉树、r 叉正则树、r 叉完全正则树分别称作r 叉有序树、r 叉正则有序树、r 叉完全正则有序树.

根子树与最优二叉树

定义 3.2 (根子树). 设 T 为一棵根树, $\forall v \in V(T)$, 称 v 及其后代的导出子图 T_v 为以 v 为根的根子树. 二叉正则有序树的每个分支点的两个儿子导出的根子树分别称为该分支点的左子树和右子树.

定义 3.3 (最优 2 叉树). 设 2 叉树 T 有 t 片树叶 $v_1, v_2, \ldots, v_t$, 权分别为 $w_1, w_2, \ldots, w_t$, 称 $W(T) = \sum_{i=1}^{t} w_i \ell_i$ 为 T 的权重, 其中 ℓ_i 是 v_i 的层数. 在所有 t 片树叶的 2 叉树中, 权重最小的 2 叉树称为最优二叉树.

Huffman 编码

Algorithm 3: Huffman Encoding

Input: An array of weights $w_1, w_2, \ldots, w_t$.

Output: An encoding tree with t leaves.

```
Create t tree leaves with weight w_1, w_2, ..., w_t;
for k ← 1 to t − 1 do
    Select vertex i, j with the minimum weights from the vertices of 0 in-degree;
    Create a vertex with children i, j and weight w_i + w_j;
return T;
```

- $W(T)$ equals to the sum of all branching nodes.
- Considering a leaf i in level ℓ_i, which branching nodes will count its weight?
- Its ancestors. The number of its ancestors is exactly ℓ_i.
- Hence, the sum of all branching nodes equals to $\sum_i^t \ell_i w_i$.

Huffman 编码

Algorithm 4: Huffman Encoding with priority queue

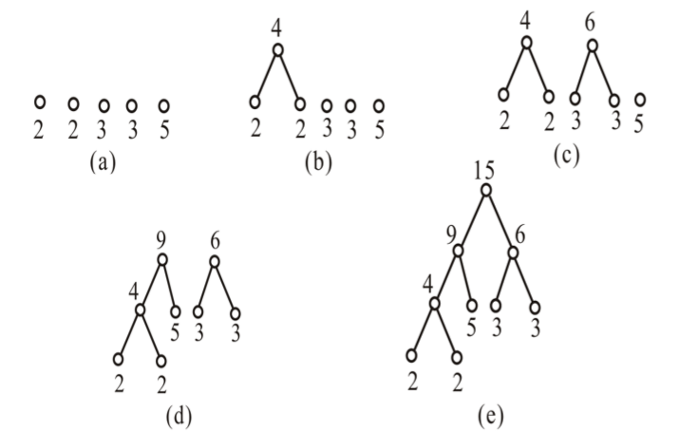
Input: An array of frequency $w_1, w_2, \ldots, w_t$.

Output: An encoding tree with t leaves.

$H \leftarrow$ makequeue(W) (using w_i as keys);
for $k \leftarrow t+1$ to $2t-1$ **do**
  $i \leftarrow$ deletemin(H), $j \leftarrow$ deletemin(H);
  Create a vertex with children i, j;
  $w_k \leftarrow w_i + w_j$;
  insert(H, k, w_k);
return T;

- Huffman encoding and Kruskal algorithm are all greedy algorithm.
- Every decision it makes is the one with the most obvious immediate advantage.

例子 *3.2.* 求权为 2, 2, 3, 3, 5 的最优树.



$W(T) = 34.$

前缀码

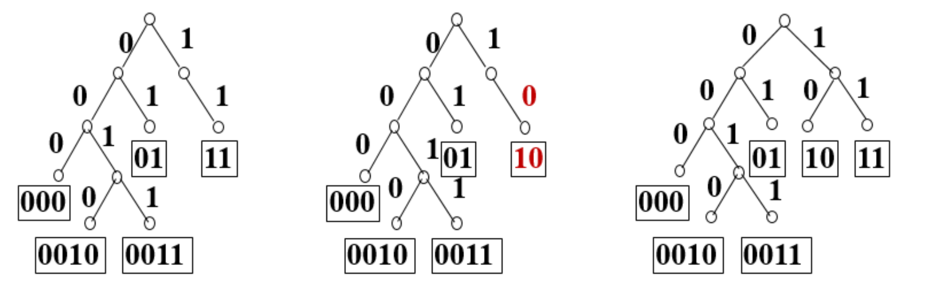
- 考虑使用二进制编码传输 4 个字母 A, B, C, D.
- 等长编码 fixed-length encoding, $A:00, B:01, C:10, D:11$.
- 传输 10 个字母代价为?
- 若不同字母出现的频率不同, $p: 0.6, 0.1, 0.2, 0.1$, 是否有更为经济的编码方式?
- 变长编码 variable-length encoding, $A:0, B:01, C:1, D:10$.
- 传输 10 个字母代价为?
- 变长编码有风险, 考虑传输字符串 001.

前缀码

定义 *3.4* (前缀码). 设 $a_1a_2\dots a_{n-1}a_n$ 是长为 n 的符号串，称其子串 $a_1, a_1a_2, \dots,$ $a_1a_2\dots a_n$ 为该符号串的前缀. 设 $A = \{\beta_1, \beta_2, ..., \beta_m\}$ 是一个符号串集合，若 A 的任意两个符号串都互不为前缀，则称 A 为前缀码 prefix-free encoding. 由 0-1 符号串构成的前缀码称作二元前缀码.

- 前缀码：$\{1, 00, 011, 0101, 01001, 01000\}$.
- 非前缀码：$\{1, 00, 011, 0101, 0100, 01001, 01000\}$.
- 如何构造前缀码?

例子 3.3. 用 2 叉树产生二元前缀码.



- 按左枝标 0、右枝标 1，一棵正则 2 叉树产生唯一的前缀码.
- 从根节点到叶子结点路径标号序列唯一.

最佳前缀码

设符号 A_i 在传输中出现的频率为 p_i, 其二元前缀码的长度为 $\ell_i, 1 \le i \le t$. 则

- 传输 m 个符号需要 $m\sum_{i=1}^{t} p_i\ell_i$ 个二进制位.
  - 记传输第 k $(1 \le k \le m)$ 个符号的开销为 X_k，则 X_k 服从何种分布? $\mathbb{E}[X_k] =?$
  - $\mathbb{E}[X_1 + X_2 + \ldots X_m] = \mathbb{E}[X_1] + \mathbb{E}[X_2] + \ldots \mathbb{E}[X_m] = m\mathbb{E}[X_1]$.
- $\sum_{i=1}^{t} p_i\ell_i$ 最小的二元前缀码称作最佳前缀码.
- 最佳前缀码可以使用 Huffman 算法计算以频率为权的最优二叉树产生.

实例

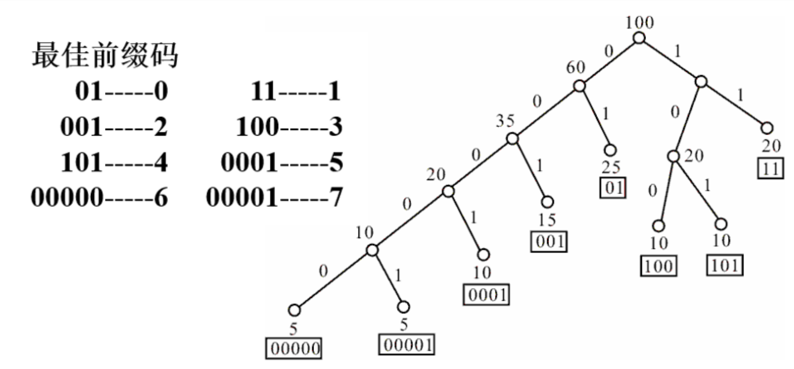
例子 3.4. 设在通信中，八进制数字出现的频率如下：

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

求传输它们的最佳前缀码，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？若用等长的码字传输需要多少个二进制数字？

用等长码需 3×10^n 位.

实例

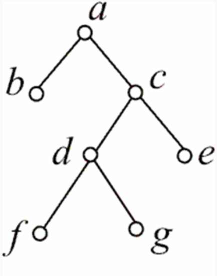


$W(T)=285$，传输 $10^n\,(n\geq 2)$ 个八进制数字, 用最佳前缀码需 2.85×10^n 位.

有序树的行遍方式

行遍有序树——对每个顶点访问且仅访问一次. 对 2 叉有序正则树的周游方式:

1. 中序行遍法 inorder traversal. 访问次序为：左子树、根、右子树.
2. 前序行遍法 preorder traversal. 访问次序为：根、左子树、右子树.
3. 后序行遍法 postorder traversal. 访问次序为：左子树、右子树、根.



例子 3.5. 用中序, 前序, 后序行遍法访问的结果分别为:

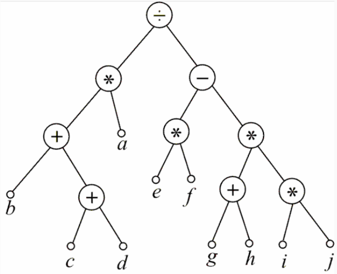
- $b\ a\ f\ d\ g\ c\ e$
- $a\ b\ c\ d\ f\ g\ e$
- $b\ f\ g\ d\ e\ ca$

用 2 叉有序树存放算式

用 2 叉有序树表示含有 2 元运算和 1 元运算的算式:

- 每个分支点放一个运算符，其运算对象是以它的儿子为树根的子树所表示的子算式.
- 规定运算对象的排列顺序，如被除数、被减数放在左边.
- 所有的变量和常量都放在树叶上.

例子 *3.6.* 用中序行遍法访问还原算式
$((b+(c+d))*a) \div ((e*f)-(g+h)*(i*j))$



波兰符号法

- 波兰符号法 (前缀符号法): 按前序行遍法访问存放算式的 2 叉有序树, 且不加括号.
- 如对上页的算式 $\div * + b + c\ d\ a - * e\ f * + g\ h * i\ j$.
- 运算规则: 从右到左每个运算符号对其后面紧邻的两个或一个对象进行运算.

- 逆波兰符号法 (后缀符号法): 按后序行遍法访问, 且不加括号.
- 如对上页的算式 $b\ c\ d + + a * e\ f * g\ h + i\ j * * - \div$.
- 运算规则: 从左到右每个运算符对其前面紧邻的两个或一个对象进行运算.

习题

主要内容

- 无向树及其性质
- 生成树、最小生成树
- 根树及其分类、最优二叉树、最佳前缀码、波兰符号法、逆波兰符号法

基本要求

- 深刻理解无向树的定义及性质
- 熟练地求解无向树
- 准确地求出给定带权连通图的最小生成树
- 理解根树及其分类等概念
- 熟练掌握求最优二叉树及最佳前缀码的方法
- 掌握波兰符号法与逆波兰符号法

练习 1

例子 4.1. 无向树 T 有 n_i 个 i 度顶点，$i = 2, 3, \ldots, k$，其余顶点全是树叶，求 T 的树叶数.

解：用树的性质，边数 $m = n - 1$（n 为阶数），及握手定理. 设有 t 片树叶,

$$2m = 2\left(\sum_{i=2}^{k} n_i + t\right) - 2 = \sum_{i=1}^{n} d(\nu_i) = \sum_{i=2}^{k} i n_i + t,$$

解得

$$t = \sum_{i=3}^{k} (i-2) n_i + 2.$$

练习 2

例子 *4.2.* 设 n 阶非平凡的无向树 T 中，$\Delta(T) \geq k, k \geq 1$. 证明 T 至少有 k 片树叶.

证明. 设 T 有 s 片树叶，由于 $\Delta(T) \geq k$，有

$$2m = 2n - 2 = \sum_{i=1}^{n} d(\nu_i) \geq 2(n - s - 1) + k + s,$$

解得 $s \geq k$.

练习 3

例子 *4.3.* 设 G 为 n 阶无向简单图，$n \geq 5$，证明 G 或 $\overline{G}$ 中必含圈.

证明 *(1).* 反证法. 否则 G 与 $\overline{G}$ 的各连通分支都是树. 设 G 与 $\overline{G}$ 的连通分支的顶点数和边数分别为 $n_i, m_i\,(1 \leq i \leq s)$ 与 $n'_j, m'_j\,(1 \leq i \leq s)$. 于是

$$\frac{n(n-1)}{2} = \sum_{i=1}^{s} m_i + \sum_{j=1}^{t} m'_j = \sum_{i=1}^{s}(n_i - 1) + \sum_{j=1}^{t}(n'_j - 1) = 2n - (s+t) \leq 2n - 2.$$

整理得

$$n^2 - 5n + 4 \leq 0,$$

解得 $1 \leq n \leq 4$，与 $n \geq 5$ 矛盾.

练习 3

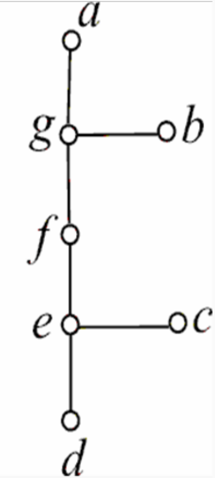
证明 *(2)*. 在 G 与 $\overline{G}$ 中有一个 (不妨设为 G) 边数

$$m \geq \frac{n(n-1)}{4} \geq n$$

若 G 是森林, 则 $m \leq n-1$, 矛盾.

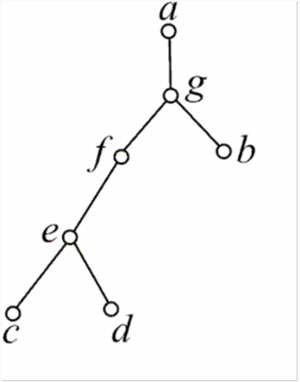
练习 4

例子 *4.4.* 画出基图为图所示无向树的所有非同构的根树.

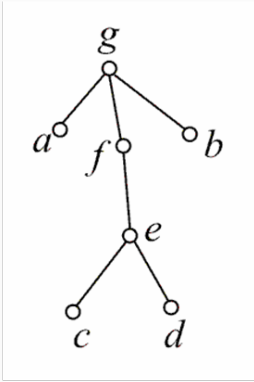


练习 4

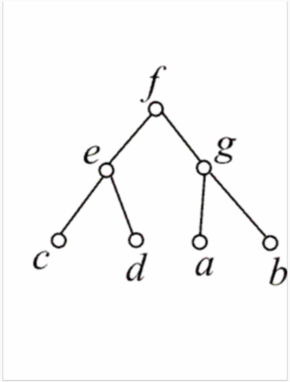
解　以 a, b, c, d 为根的根树同构, 选 a 为根, 如 (1)；以 e, g 为根的根树同构，取 g 为根，如 (2)；以 f 为根，如 (3).



(1)



(2)



(3)

练习 5

例子 *4.5.* 设 T 是正则 2 叉树，T 有 t 片树叶，证明 T 的阶数 $n=2t-1$.

证明 *(1).* 利用正则 2 叉树的定义及树的性质直接证明.

1. $n=t+i$ (i 为分支点数)
2. $n=m+1$ （m 为 T 的边数）
3. $m=2i$ （正则 2 叉树定义）

由 (2)、(3) 得 $i=\frac{n-1}{2}$，代入（1）得 $n=2t-1$.

证明 *(2).* 利用握手定理及树的性质证. T 的树根为 2 度顶点，所有内点为 3 度顶点，叶为 1 度顶点，有

1. $2m=2+3(i-1)+t$
2. $m+1=n=i+t$

由 (1) 和 (2) 可解得 $n=2t-1$.

作业

作业

习题 16:

- 4, 5, 9, 13.
- 24, 25, 29, 30.
- 36, 41, 42.