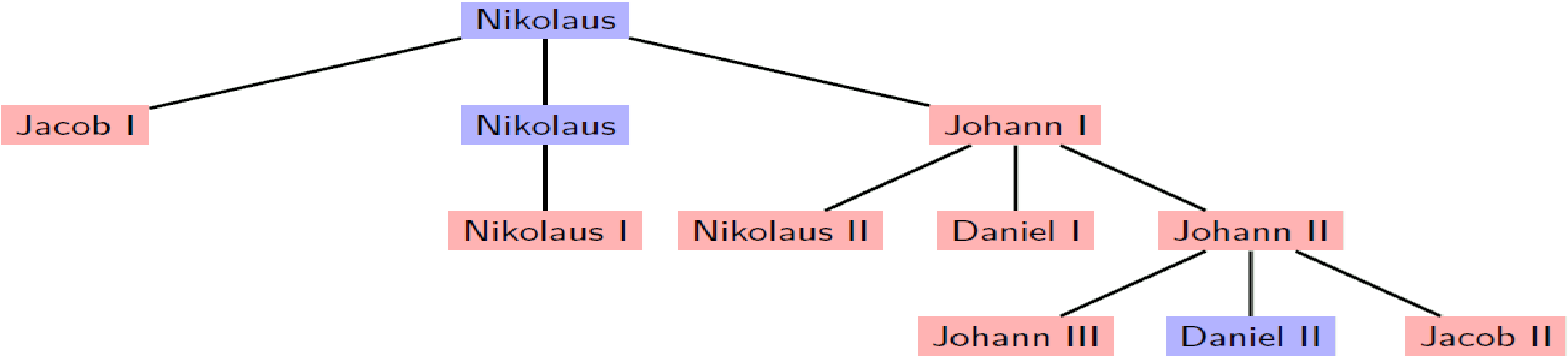
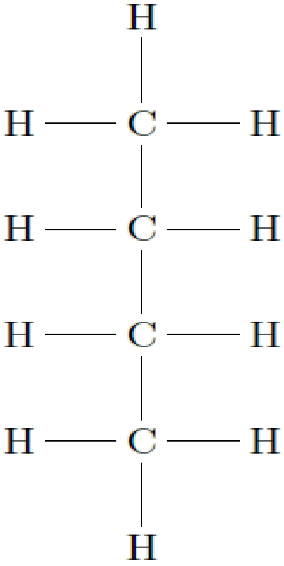
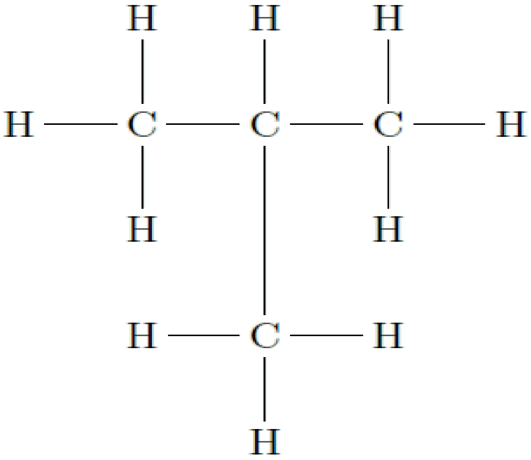
引入



瑞士数学家中的著名家族-伯努利家族的族谱图

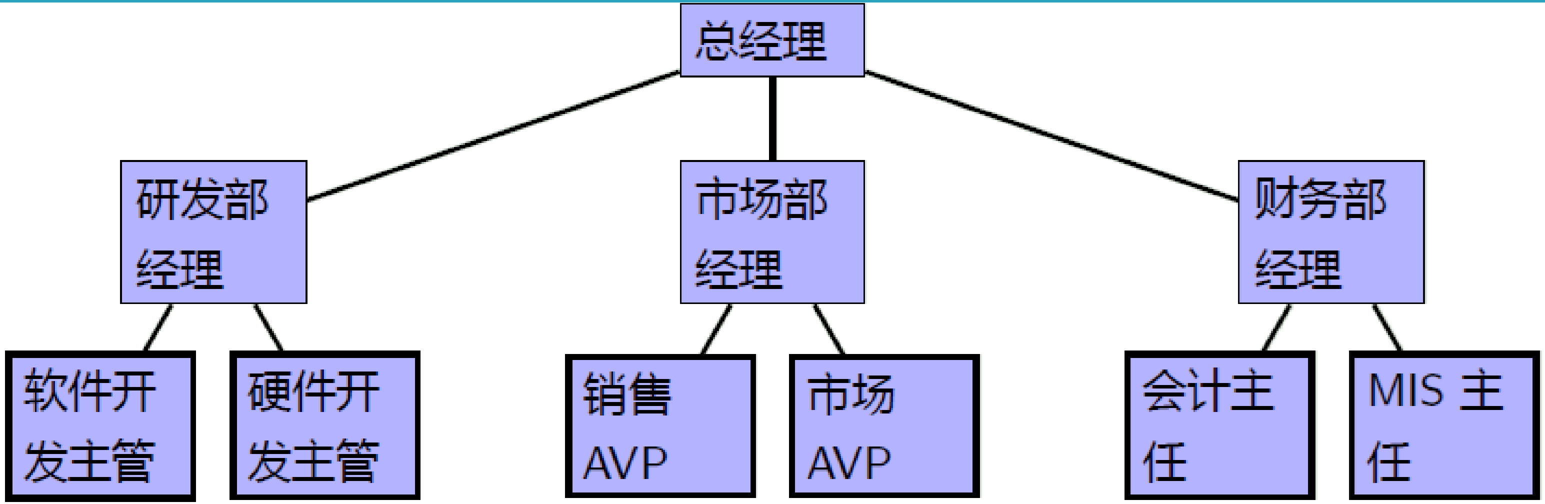


丁烷



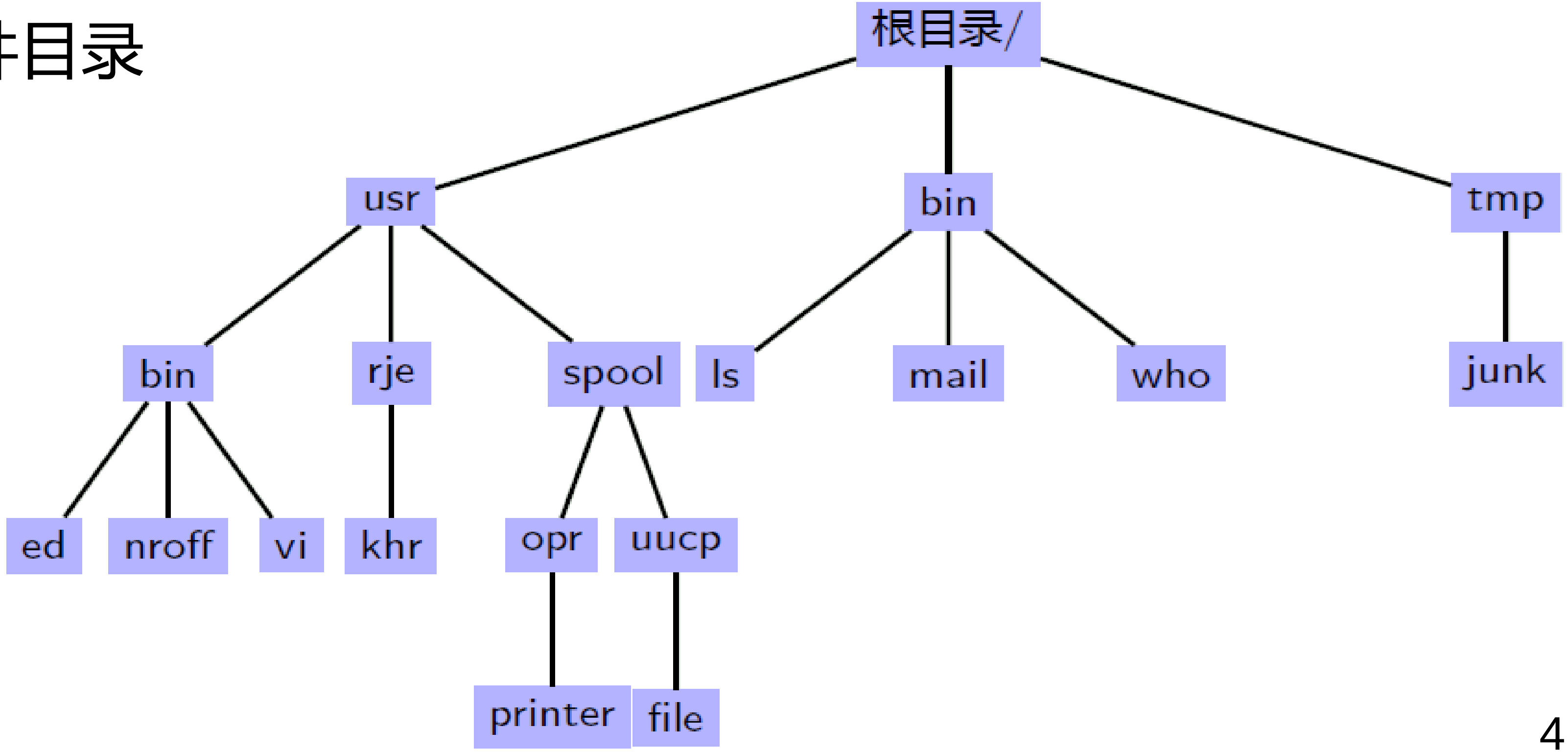
异丁烷

这是英国数学家凯莱用于表示饱和碳氢化合物(形如C_nH_{2n+2}) 的方法，从而发现了树。



大的组织机构的结构可以用树来建模，每个结点表示一个职务。

文件目录


根目录/
usr
bin
tmp
bin
rje
spool
ls
mail
who
junk
ed
nroff
vi
khr
opr
uucp
printer
file

无向树

16.1 无向树及其性质

定义16.1

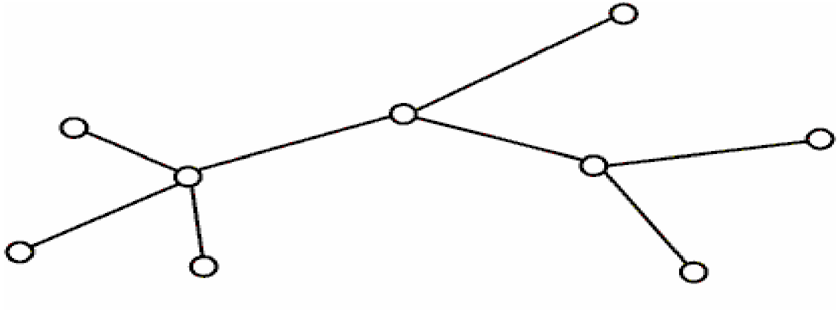
(1) 无向树——**连通无回路**的无向图

(2) 平凡树——平凡图

(3) 森林——至少由两个连通分支（每个都是树）组成

(4) 树叶——1度顶点

(5) 分支点——度数≥2的顶点



无向树的等价定义

定理16.1 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.

(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，在所得图中得到惟一的一个含新边的圈.

树的特点

- 在结点给定的无向图中
  - 树是边数最多的无回路图
  - 树是边数最少的连通图
- 在无向图$G = (n, m)$中
  - 若$m < n-1$，则G是不连通的
  - 若$m > n-1$，则G必含回路

无向树的性质

定理16.2 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，由握手定理及定理16.1可知，

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

由上式解出$x\geq 2$.

平凡树有多少片树叶?

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.

解　解本题用树的性质$m=n-1$，握手定理.

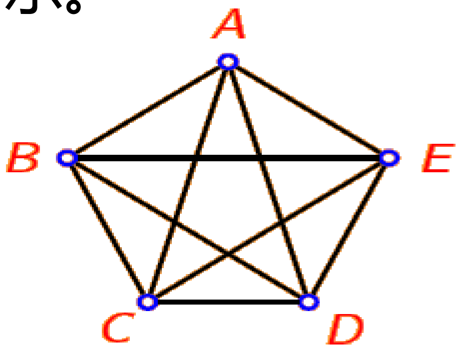
设有x片树叶，于是 $n = 1+2+x = 3+x$,

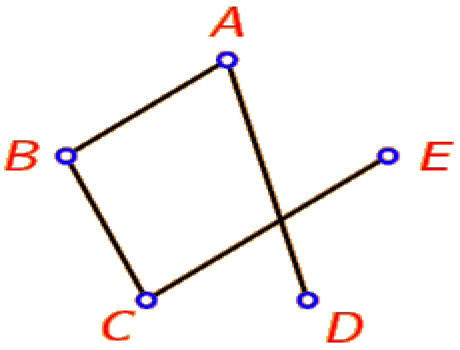
$$2m = 2(n-1) = 2\times(2+x) = 1\times3+2\times2+x$$

解出$x = 3$，故T有3片树叶.

生成树

- 考虑构建一个包含5 个信息中心A,B,C,D,E 的通信系统，可能的光纤连接如下左图所示。
- 由于费用限制，要求铺设尽可能少的光纤线路，但又必须保持网络畅通。这实际上就是要求出一个边最少的连通图，这恰好符合树的特点。因而问题转化成在一个连通图中找到一棵树，一种可能的方式如右图所示。





16.2 生成树

定义16.2 设G为无向图

(1) G的树——T 是G 的子图并且是树

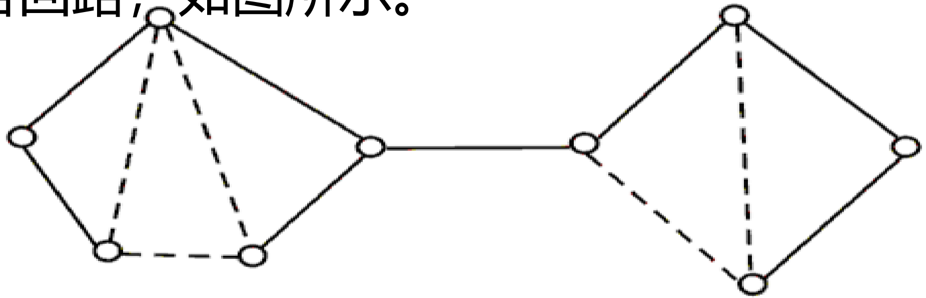
(2) G的生成树——T **是G 的生成子图并且是树**

(3) 生成树T的树枝——T 中的边

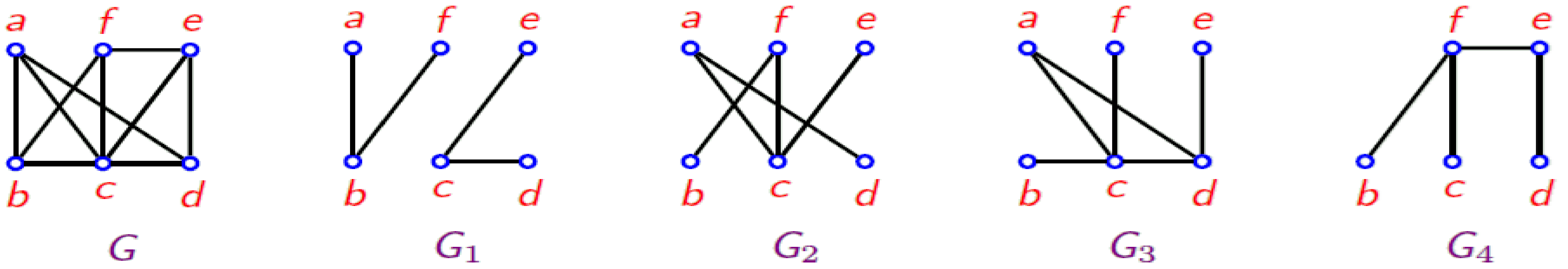
(4) 生成树T的弦——不在T 中的边

(5) 生成树T的余树 $\bar{T}$ ——全体弦组成的集合的导出子图

$\bar{T}$不一定连通，也不一定不含回路，如图所示。



- G_1，G_2，G_3，G_4中，哪一个是图G的生成树?



生成树存在条件

定理16.3 无向图G具有生成树当且仅当G连通.

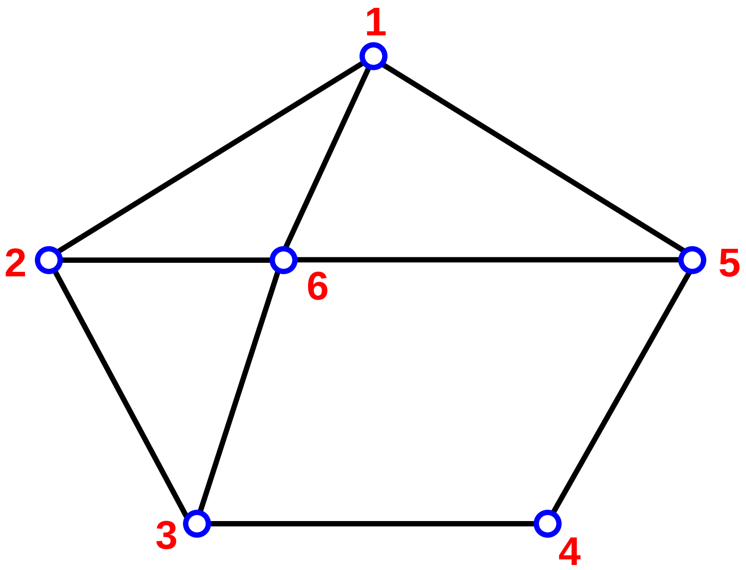
证 必要性显然.

充分性用破圈法（注意：在圈上删除任何一条边，不破坏连通性)

小贴士——求连通图G = (n, m)的生成树

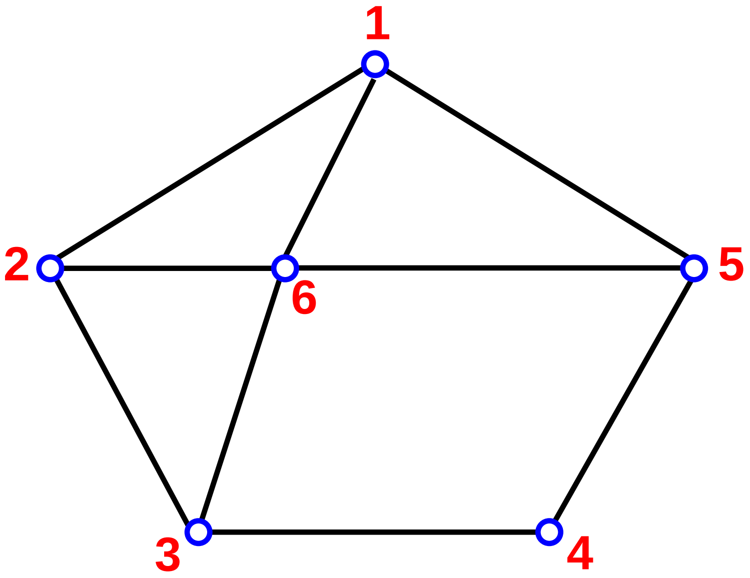
- **破圈法**：找出一条回路，并删除该回路中的一条边，直到图中没有回路为止，删除的边的总数为m-n+1。
- **避圈法**：选取一条边，验证该边与已选取的边不构成回路，选取的边的总数为n-1。

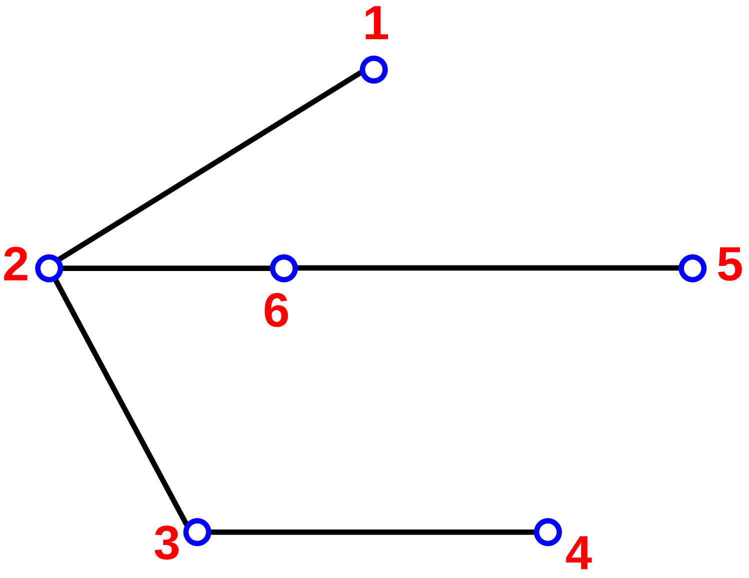
用破圈法求生成树



➢ 由于$n = 6$，$m = 9$，所以$m - n + 1 = 4$，故要删除的边数为4，因此只需4步即可。

用避圈法求生成树





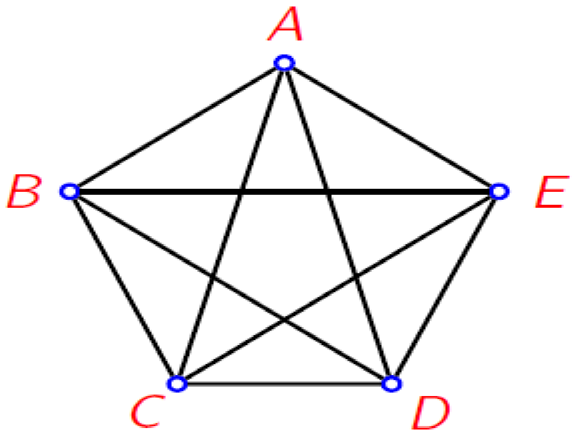
- 由于n = 6，所以n-1 = 5，故要选取5条边，因此需5步即可。

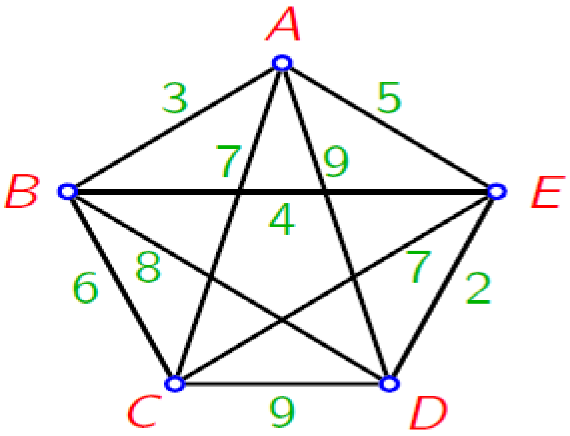
说明

- 由于生成树的形式不惟一，故上述两棵生成树都是所求的。
- 破圈法和避圈法的计算量较大，主要是需要找出回路或验证不存在回路。

最小生成树

- 构建一个包含5 个信息中心A,B,C,D,E 的通信系统的问题，如下左图所示。通常情况下，各中心之间的光纤连接长度并不相同，这会影响总体费用。所以我们建立一个带权图(以百公里为单位，如右图所示)，希望能从这个图中找出一棵生成树，而且总权值最小。





最小生成树

定义16.3 T是$G=<V,E,W>$的生成树

(1) $W(T)$——T各边权之和

(2) 最小生成树——G的**所有生成树中权最小**的

- 求最小生成树的算法
  - Kruskal算法
  - Prim算法

Kruskal算法和Prim算法

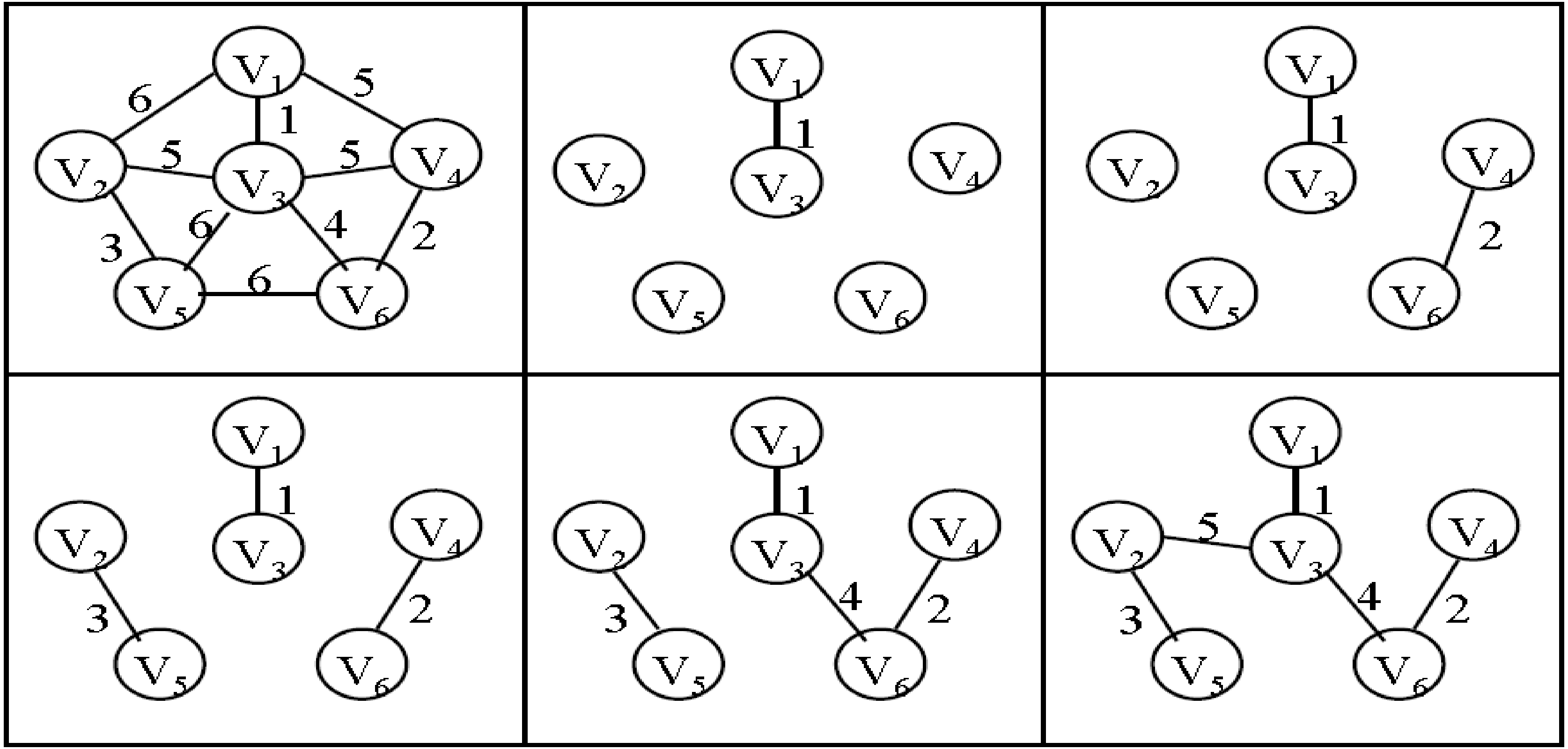
- Kruskal算法过程

- 采用贪心策略
- 每次从所有边中选取最小的边，限制条件是加入这条边后不能使当前的结点形成回路。

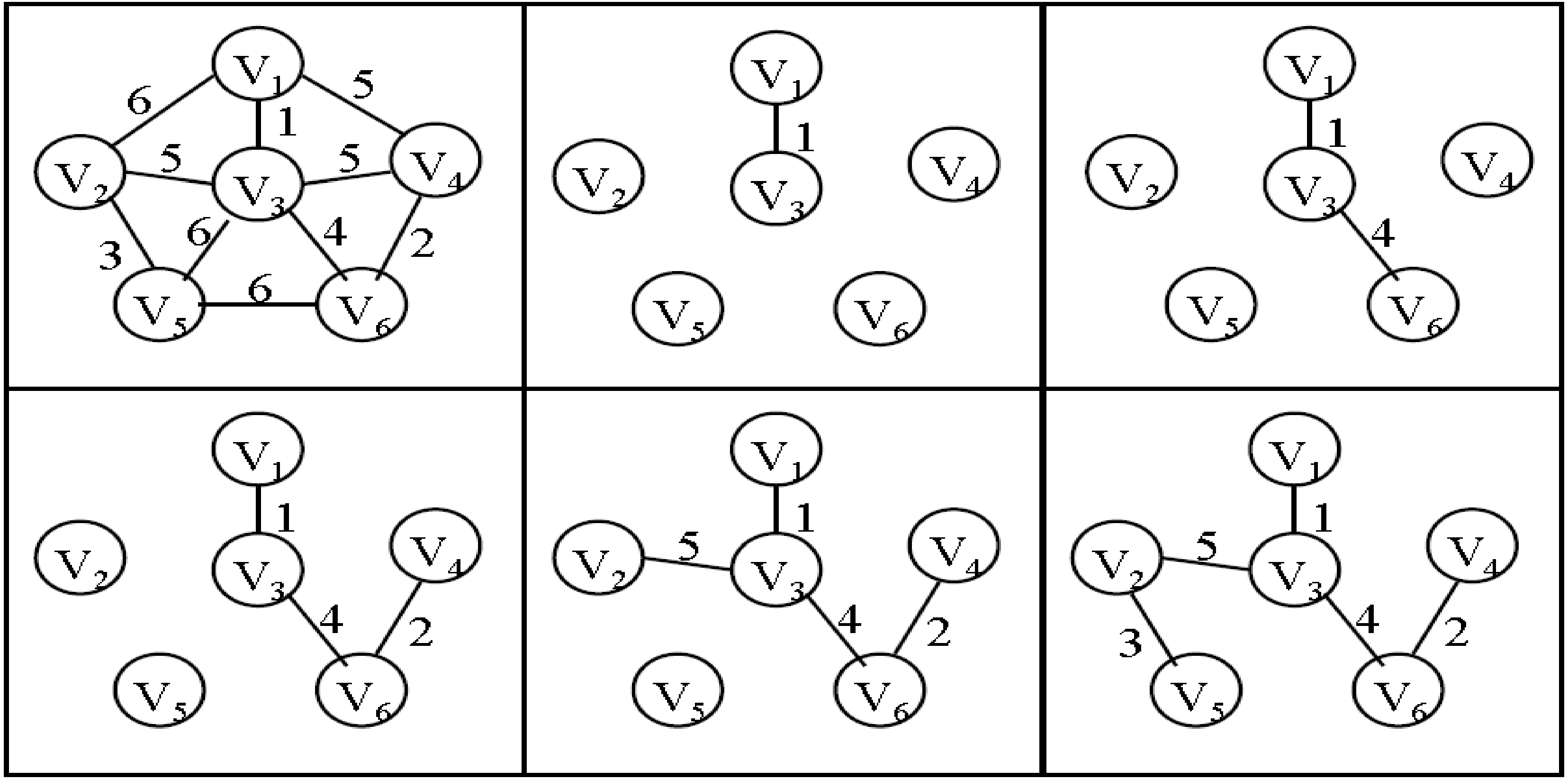
- Prim算法过程

- 同样是采用贪心策略
- 第一步任意选取一个结点加入集合A，选择边的时候，选择连接A和其余不在A中点的最小权重的边（和Dijkstra单源最短路算法类似）。

Kruskal算法示意图：



Prim算法示意图：



Kruskal算法和Prim算法

- **Kruskal算法是对边集的操作**
  - Kruskal算法中，集合A是一个森林，其结点就是给定图的结点。
  - 每次加入到集合A中的安全边永远是权重最小的连接两个不同分量的边。
- **Prim算法是对点集的操作**
  - Prim算法中，集合A是一棵树。
  - 每次加入到A中的安全边永远是连接A和A之外的某个结点的边中权重最小的边。

贪心策略

小结

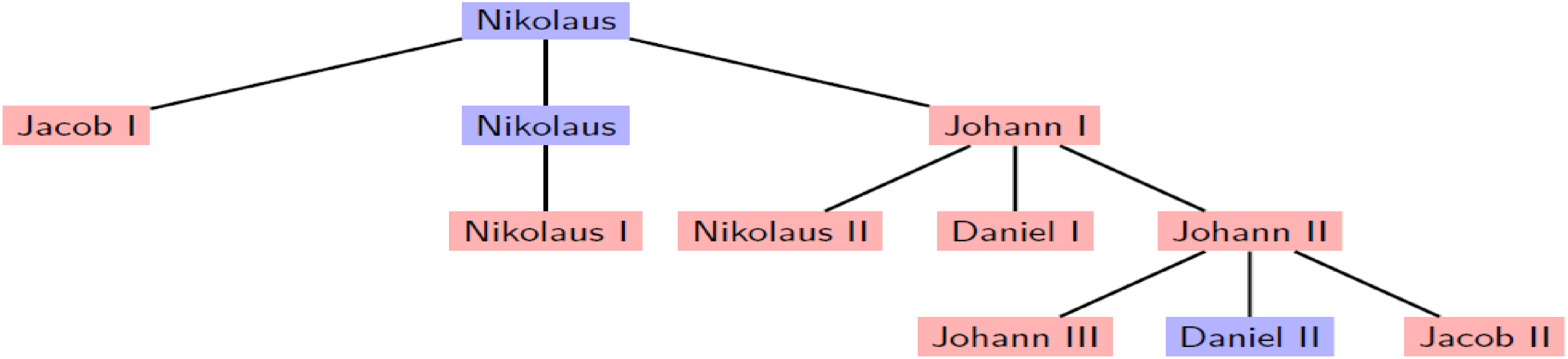
- 树
  - 无向树
    - 生成树
      - 最小生成树：Kruskal Prim

作业

- 习题16
  - 第25题 （a）求最小生成树

复习引入

- 无向树
- 有向树?


Nikolaus
Jacob I
Nikolaus
Johann I
Nikolaus I
Nikolaus II
Daniel I
Johann II
Johann III
Daniel II
Jacob II

根树

16.3 根树及其应用

定义16.4 T是有向树（基图为无向树）

(1) T 为根树——T 中一个顶点入度为0，其余的入度均为1.

(2) 树根——入度为0的顶点

(3) 树叶——入度为1，出度为0的顶点

(4) 内点——入度为1，出度不为0的顶点

(5) 分支点——树根与内点的总称

(6) 顶点v的层数——从树根到v的通路长度

(7) 树高——T 中层数最大顶点的层数

(8) 平凡根树——平凡图

小贴士-有向树的判断

（1）将所有有向边都略去方向变为无向边，得到一个无向图。

（2）判断该无向图是否是树

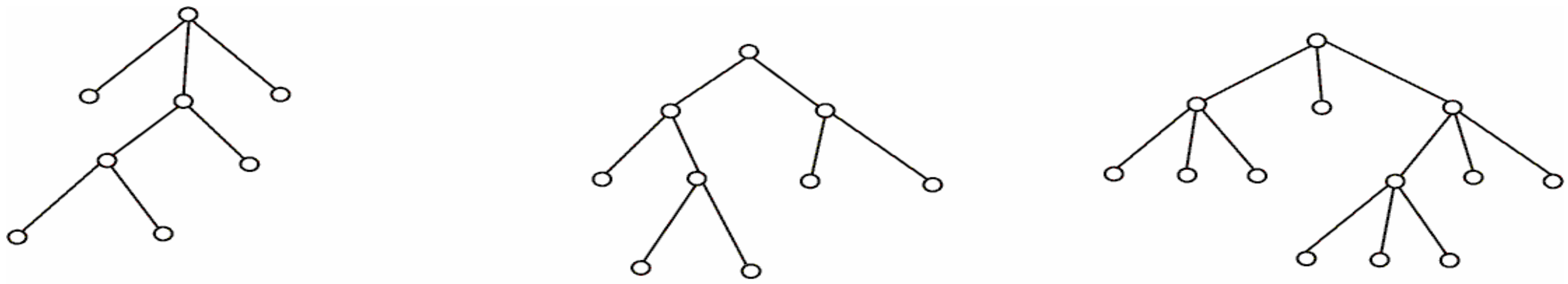
小贴士-根树的判断

（1）判断是否为有向树。

（2）计算所有结点的度数，看是否恰有一个结点的入度为0，其余所有结点的入度均为1。

根树实例

- 根树的画法——树根放上方，省去所有有向边上的箭头



家族树与根子树

定义16.5 T 为非平凡根树

(1) 祖先与后代

(2) 父亲与儿子

(3) 兄弟

定义16.6 设v为根树T中任意一顶点，称v及其后代的导出子图为以v为根的根子树.

最优二叉树

引入

等长编码

- 在计算机及通讯业中，常用二进制编码来表示符号。
- 例如，可用00、01、10、11 分别表示字母*A*、*B*、*C*、*D*，这称作等长编码。这在四个字母出现频率基本相等的情况下是非常合理的。

不等长编码

- 字母出现的频率很不一样，如*A*出现的频率为50%，*B*出现的频率为25%，*C*出现的频率为20%，*D*出现的频率为5%时，使用等长编码就不是最优的方式了。
- 可以用不等长编码。

- 在同样传输100个字母的情况下，等长编码需2*100=**200** 个二进制位
- 而不等长编码（ 如用000表示字母*D*，用001表示字母*C*，01表示*B*，1表示*A*）仅需3*5+3*20+2*25+1*50=**175** 个二进制位。

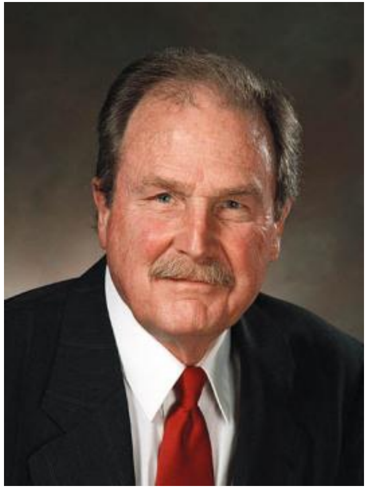
- 但不等长编码不能随意定义，否则会引起问题，如当我们用1表示A，用00表示B，用001表示C，用000表示D 时，如果接收到的信息为001000，则无法辨别它是CD 还是BAD。

前缀编码原则

最优二叉树

定义16.7 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 $W(t)=\sum_{i=1}^{t} w_i l(v_i)$为$T$的权，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叉树中，权最小的2叉树称为最优二叉树.

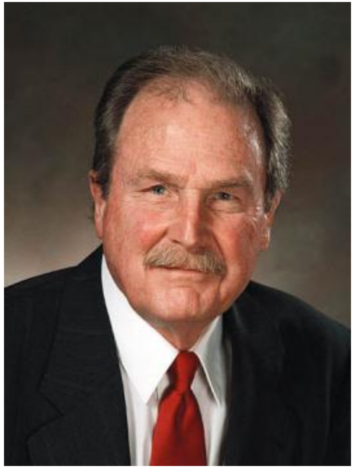
求最优树的算法—— Huffman算法



戴维·哈夫曼

美国计算机科学家

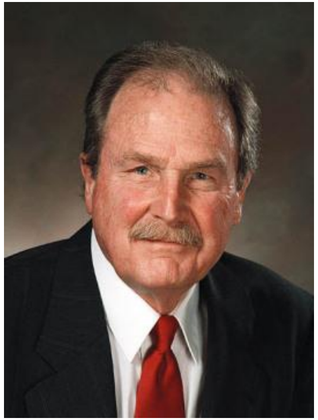
- 他从小聪慧好学，在俄亥俄州立大学毕业时只有17岁。
- 然后他进入麻省理工学院（MIT）一边工作，一边深造，哈夫曼编码就是他在1952年做博士论文时发明的。这是一种根据字母的使用频率而设计的变长码，能大大提高信息的传输效率，至今仍有广泛的应用。
- 在MIT一直工作到1967年。
- 之后转入加州大学的圣克鲁兹分校，成为该校计算机科学系的创始人，并于1970年至1973年任系主任，为此系的学术研究做出了许多杰出贡献。



戴维•哈夫曼

美国计算机科学家

- 除了哈夫曼编码外，哈夫曼在其他方面也有不少创造，他设计的二叉最优搜索树算法就被认为是同类算法中效率最高的，因而被命名为**哈夫曼算法**，是**动态规划**（Dynamic Programming）的一个范例。哈夫曼算法也广泛应用于**传真机**、**图像压缩**和**计算机安全**领域。但他却从未为此算法申请过专利或其他能够为他带来经济利益的东西，而是将全部的精力放在教学上，用他自己的话来说，“我所要给世界带来的就是我的学生。”



戴维•哈夫曼

美国计算机科学家

- 1982年获得美国电气与电子工程师学会（IEEE）计算机先驱奖。
- 1973年获得IEEE的麦克道尔（McDowell）奖。
- 1998年IEEE下属的信息论分会为纪念信息论创立50周年，授予他金禧（Golden Jubilee）奖。
- 1999年6月，荣获了以哈明码发明人命名的哈明奖章（Hamming Medal）。

最优二叉树

求最优树的算法—— Huffman算法

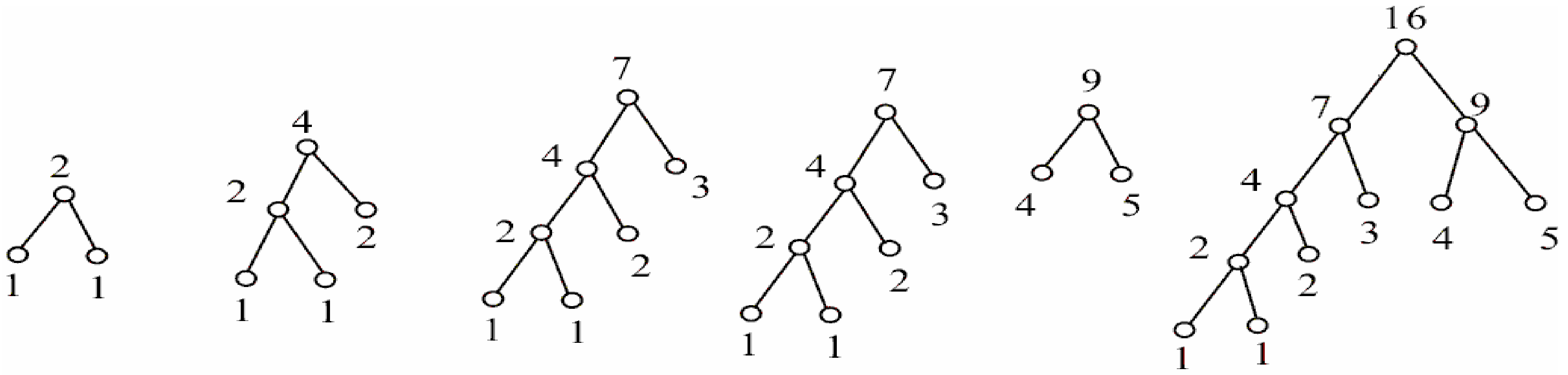
给定实数$w_1, w_2, \ldots, w_t$，且$w_1 \le w_2 \le \ldots \le w_t$.

(1) 连接权为w_1, w_2的两片树叶，得一个分支点，其权为w_1+w_2.

(2) 在$w_1+w_2, w_3, \ldots, w_t$中选出两个最小的权，连接它们对应的顶点(不一定是树叶)，得新分支点及所带的权.

(3) 重复(2)，直到形成 $t-1$个分支点，t片树叶为止.

例 4　求带权为1, 1, 2, 3, 4, 5的最优树.
解题过程由图给出，$W(T)=38$



最佳前缀码

定义16.8 设$\alpha_1, \alpha_2, \ldots, \alpha_{n-1}, \alpha_n$是长度为 n 的符号串

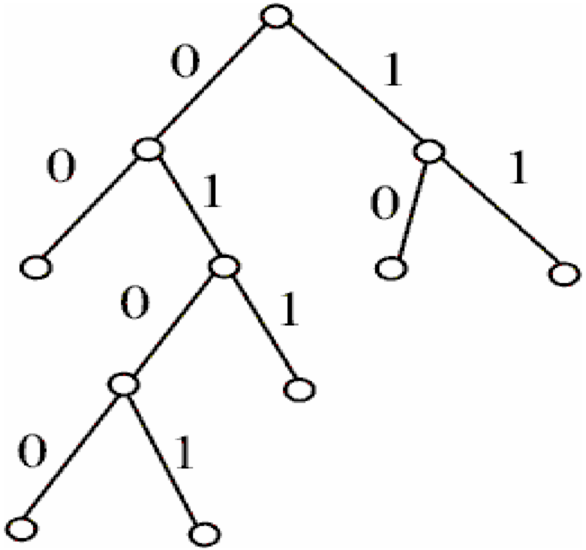
(1) 前缀——$\alpha_1, \alpha_1\alpha_2, \ldots, \alpha_1\alpha_2\ldots\alpha_{n-1}$

(2) 前缀码——$\{\beta_1, \beta_2, \ldots, \beta_m\}$中任何两个元素互不为前缀

(3) 二元前缀码——$\beta_i\,(i=1, 2, \ldots, m)$ 中只出现两个符号，如0与1.

图所示二叉树产生的前缀码为

{ 00, 10, 11, 011, 0100, 0101 }



用Huffman算法产生最佳前缀码

例5 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%

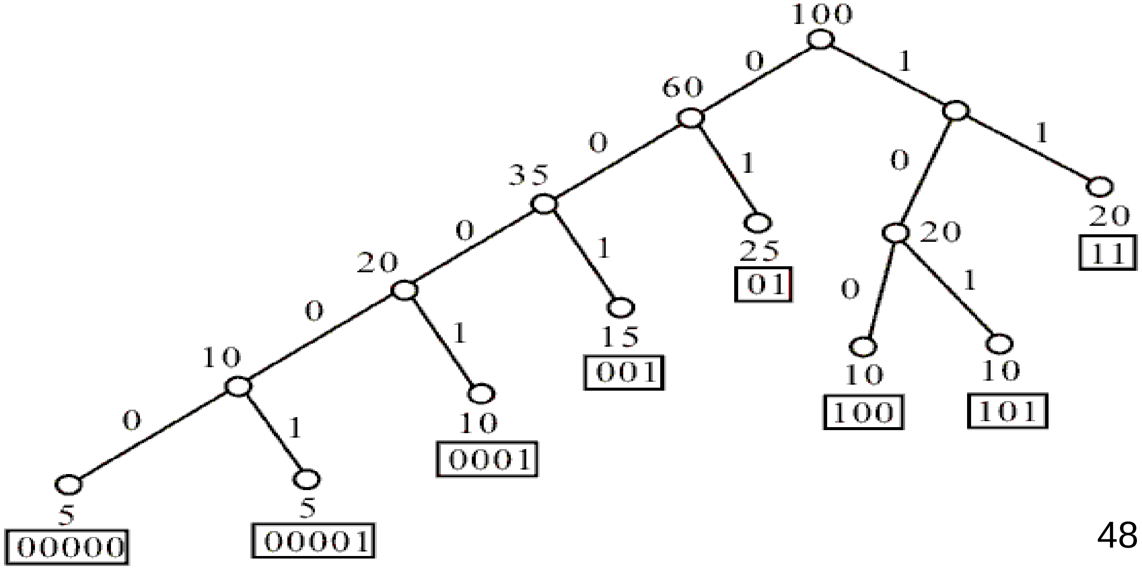
求传输它们的最佳前缀码，并求传输10^n（$n\geq 2$）个按上述比例出现的八进制数字需要多少个二进制数字？若用等长的（长为3）的码字传输需要多少个二进制数字？

求最佳前缀码

解 用100个八进制数字中各数字出现的个数，即以100乘各频率为权，并将各权由小到大排列，得w_1=5, w_2=5, w_3=10, w_4=10, w_5=10, w_6=15, w_7=20, w_8=25. 用此权产生的最优树如图所示.

01-----0	11-----1
001-----2	100-----3
101-----4	0001-----5
00000-----6	00001-----7

$W(T)$=285，传$10^n(n\geq 2)$个用二进制数字需2.85×10^n个，用等长码需3×10^n个数字.



应用于决策问题

- 用机器分辨一些币值为1 分、2 分、5 分的硬币，假设各种硬币出现的概率分别为0.5、0.4、0.1。
- 问如何设计一个分辨硬币的算法，使所需的时间最少？(假设每作一次判别所用的时间相同，以此为一个时间单位)

所需时间：2*0.1 + 2*0.4+ 1*0.5=1.5(时间单位)。

小结

- 树
  - 无向树---生成树（最小生成树）
  - 有向树（根树）---最优二叉树

作业

- 习题16
  - 第41题 最优二叉树的应用


THANK
YOU