

定理 1 设 m 和 n 是正整数满足 $m \geq n$ 。那么存在

$$n^m - C(n, 1)(n-1)^m + C(n, 2)(n-2)^m - \cdots + (-1)^{n-1} C(n, n-1) \cdot 1^m$$

个从 m 元素集到 n 元素集的映上函数。

从 m 个元素的集到 n 个元素的集的映上函数是这样一种对应方式：它把定义域中的 m 个元素分配到 n 个不可辨别的盒子中，使得每个盒子都不是空的，然后将陪域中的 n 个元素中的每一个元素都与一个盒子相对应。这意味着从具有 m 个元素的集到具有 n 个元素的集的映上函数的个数，等于把 m 个可辨别的物体分配到 n 个不可辨别的盒子中，使得每个盒子都不空时的方法数乘以具有 n 个元素的集的排列数。因此，从 m 个元素的集到 n 个元素的集的映上函数的个数为 $n! S(m, n)$ ，其中 $S(m, n)$ 是 3.5 节中定义的第二类斯特林数。这意味着我们可以用定理 1 来推导 3.5 节所给的关于 $S(m, n)$ 的公式。（关于第二类斯特林数更详细的信息，可参见 [MiRo91] 中的第 6 章。）

下面给出定理 1 的另一个应用的实例。

例 3 把 5 项工作分给 4 个不同的雇员，如果每个雇员至少分配 1 项工作，问有多少种方式？

解 把工作分配看作从 5 个工作集到 4 个雇员集的函数。每个雇员至少得到 1 项工作的分配对应于从工作集到雇员集的映上函数。因此，由定理 1 存在

$$4^5 - C(4, 1)3^5 + C(4, 2)2^5 - C(4, 3)1^5 = 1024 - 972 + 192 - 4 = 240$$

种方式来分配工作并使得每个雇员至少得到 1 项工作。 ■

4.6.5 错位排列

下面将用容斥原理计数排列 n 个物体并使得没有一个物体在它的初始位置上的方式数。考虑下面的例子。

例 4 帽子认领问题 在一个餐厅里，一个新的雇员寄存 n 个人的帽子时忘记把寄存号放在帽子上。当顾客取回他们的帽子时，这个雇员从剩下的帽子中随机选择发给他们。问没有一个人收到自己的帽子的概率是多少？ ■

注意 答案就是重新排列帽子使得没有帽子在它的初始位置上的方式数除以 n 个帽子的排列数 $n!$ 。在我们找出排列 n 个物体并使得没有一个物体在它的初始位置上的方式数后再考虑这个例子。



一个错位排列是使得没有一个物体在它的初始位置上的排列。为求解例 4 中的问题我们需要确定 n 个物体的错位排列数。

例 5 排列 21453 是 12345 的一个错位排列，因为没有数在它的初始位置上。但是，21543 不是 12345 的错位排列，因为 4 留在它的初始位置上。 ■

设 D_n 表示 n 个物体的错位排列数。例如， $D_3 = 2$ ，因为 123 的错位排列是 231 和 312。我们将使用容斥原理对所有的正整数 n 求 D_n 。

定理 2 n 元素集的错位排列数是

$$D_n = n! \left[1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \cdots + (-1)^n \frac{1}{n!} \right]$$

证 如果排列保持元素 i 不变，就设排列有性质 P_i 。错位排列的个数就是对 $i=1, 2, \dots, n$ ，没有性质 P_i 的排列数，或

$$D_n = N(P'_1 P'_2 \cdots P'_n)$$

使用容斥原理得到

$$D_n = N - \sum_i N(P_i) + \sum_{i < j} N(P_i P_j) - \sum_{i < j < k} N(P_i P_j P_k) + \cdots + (-1)^n N(P_1 P_2 \cdots P_n)$$

其中 N 是 n 个元素的排列数。这个等式说明，所有的元素都发生变化的排列数，等于排列的总数减去至少保持 1 个元素不变的排列数，加上至少保持 2 元素不变的排列数，减去至少保持 3 个元素不变的排列数，等等。现在找出在等式右边出现的所有的量。

首先注意到 $N=n!$, 因为 N 仅仅就是 n 个元素排列的总数。还有, $N(P_i)=(n-1)!$, 这是由乘积法则得到的, 因为 $N(P_i)$ 是保持元素 i 不变的排列数, 因而第 i 个位置是确定的, 但是其余的每个位置可以放任意元素。类似地,

$$N(P_i P_j) = (n-2)!$$

因为这是保持元素 i 和 j 不变的排列数, 但是其余 $(n-2)$ 个元素的位置可以被任意地安排。一般说来有

$$N(P_{i_1} P_{i_2} \cdots P_{i_m}) = (n-m)!$$

因为这是保持元素 $i_1, i_2, \dots, i_m$ 不变的排列数, 但是其他 $(n-m)$ 个元素的位置可以被任意安排。由于存在 $C(n, m)$ 种方式从 n 个元素中选择 m 个, 从而有

$$\begin{aligned} \sum_{1 \leq i \leq n} N(P_i) &= C(n, 1)(n-1)! \\ \sum_{1 \leq i < j \leq n} N(P_i P_j) &= C(n, 2)(n-2)! \end{aligned}$$

一般地, 有

$$\sum_{1 \leq i_1 < i_2 < \cdots < i_m \leq n} N(P_{i_1} P_{i_2} \cdots P_{i_m}) = C(n, m)(n-m)!$$

所以, 把这些等式代入关于 D_n 的公式得

$$\begin{aligned} D_n &= n! - C(n, 1)(n-1)! + C(n, 2)(n-2)! - \cdots + (-1)^n C(n, n)(n-n)! \\ &= n! - \frac{n!}{1!(n-1)!}(n-1)! + \frac{n!}{2!(n-2)!}(n-2)! - \cdots + (-1)^n \frac{n!}{n!0!}0! \end{aligned}$$

化简这个表达式得

$$D_n = n! \left[1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + (-1)^n \frac{1}{n!} \right]$$

□

现在对于给定的正整数 n 求 D_n 就简单了。例如, 使用定理 2 得

$$D_3 = 3! \left[1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} \right] = 6 \left(1 - 1 + \frac{1}{2} - \frac{1}{6} \right) = 2$$

正如我们前面所看到的。

现在可以给出例 4 中的问题的解。

解 没有一个人收到自己的帽子的概率是 $D_n/n!$ 。由定理 2, 这个概率是

$$\frac{D_n}{n!} = 1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + (-1)^n \frac{1}{n!}$$

对于 $2 \leq n \leq 7$, 这个概率的值在表 4-3 中给出。

表 4-3 错位排列的概率

n	2	3	4	5	6	7
$D_n/n!$	0.500 00	0.333 33	0.375 00	0.366 67	0.368 06	0.367 86

使用微积分的方法可以证明

$$e^{-1} = 1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + (-1)^n \frac{1}{n!} + \cdots \sim 0.368$$

因为这是一个项趋向于 0 的交错级数, 当 n 无限增长时, 没有一个人取回自己的帽子的概率趋于 $e^{-1} \sim 0.368$ 。事实上, 可以证明这个概率与 e^{-1} 的差在 $1/(n+1)!$ 之内。 ■

练习

- 假设 1 蒲式耳 100 个苹果中 20 个有虫, 15 个有擦伤。只有没虫也没擦伤的苹果才可以卖。如果 10 个擦伤的苹果有虫, 那么 100 个苹果中有多少个可以卖?
- 方程 $x_1 + x_2 + x_3 = 13$ 有多少个解? 其中 x_1, x_2, x_3 是小于 6 的非负整数。
- 使用容斥原理求小于 200 的素数的个数。
- 有多少小于 10 000 的正整数不是一个整数的 2 次或更高次幂?

5. 有多少种方式把 6 个不同的玩具分给 3 个不同的孩子并使得每个孩子至少得到 1 个玩具?
6. 有多少种方式把 7 项不同的工作分给 4 个不同的雇员, 使得每个雇员至少得到 1 项工作, 并且把最困难的工作分给最好的雇员?
7. 一个 7 元素集合有多少个错位排列?
8. 一个把信放入信袋的机器发生了故障并且随机把信放入信袋中。在一组 100 封信中发生下面事件的概率是多少?
 - a) 没有信放对了信袋。
 - b) 恰好 1 封信放对了信袋。
 - c) 恰好 98 封信放对了信袋。
 - d) 恰好 99 封信放对了信袋。
 - e) 所有的信都放对了信袋。
- *9. 有多少种方式安排数字 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 使得没有偶数在它的初始位置上?
10. 对哪些正整数 n , 错位排列数 D_n 是偶数?
- *11. 当 n 的素因子分解式是

$$n = p_1^{a_1} p_2^{a_2} \cdots p_m^{a_m}$$

时, 使用容斥原理推导一个关于 $\phi(n)$ 的公式。

12. 有多少个 $\{1, 2, 3, 4, 5, 6\}$ 的错位排列的前 3 位是取某种次序的整数 1, 2, 3?
13. 证明定理 1。

关键术语和结果

术语

递推关系: 一个公式, 它把序列除了某些初始项以外的项, 表示成这个序列前面的一个或若干个项的函数。

递推关系的初始条件: 满足递推关系的序列在该关系起作用之前的某些项的值。

常系数线性齐次递推关系: 一个递推关系, 除了初始项之外, 它把序列的项表示成前面项的线性组合。

常系数线性齐次递推关系的特征根: 与常系数线性齐次递推关系相关的多项式的根。

常系数线性非齐次递推关系: 一个递推关系, 除了初始项之外, 它把序列的项表示成前面项的线性组合加上一个仅仅依赖于序标的不恒为 0 的函数。

分治算法: 求解问题的一种算法, 求解中递归地把问题划分成固定数目的较小的同种类型的问题。

序列的生成函数: 用序列的第 n 项作为 x^n 的系数的形式幂级数。

埃拉托色尼筛: 找出小于一个给定正整数的素数的过程。

错位排列: 使得没有物体在它的初始位置上的排列。

结果

两个有穷集合并集的元素数公式:

$$|A \cup B| = |A| + |B| - |A \cap B|$$

三个有穷集合并集的元素数公式:

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

容斥原理:

$$\begin{aligned} |A_1 \cup A_2 \cup \cdots \cup A_n| &= \sum_{1 \leq i \leq n} |A_i| - \sum_{1 \leq i < j \leq n} |A_i \cap A_j| \\ &\quad + \sum_{1 \leq i < j < k \leq n} |A_i \cap A_j \cap A_k| \\ &\quad - \cdots + (-1)^{n+1} |A_1 \cap A_2 \cap \cdots \cap A_n| \end{aligned}$$

从 m 元素集到 n 元素集的映上函数个数:

$$n^m - C(n, 1)(n-1)^m + C(n, 2)(n-2)^m - \cdots + (-1)^{n-1} C(n, n-1) \cdot 1^m$$

n 个物体的错位排列数:

$$D_n = n! \left[1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + (-1)^n \frac{1}{n!} \right]$$

复习题

1. a) 什么是递推关系?
b) 如果在一个获利 9% 的账上储蓄 1 000 000 美元, 求与 n 年后账上钱数有关的递推关系。
2. 解释怎样用斐波那契数求解关于兔子的斐波那契问题。
3. a) 找出与求解汉诺塔难题的所需步数有关的递推关系。
b) 显示怎样使用迭代来求解这个递推关系。
4. a) 解释怎样找一个与不包含两个连续的 1 的 n 位二进位串个数有关的递推关系。
b) 描述另一个计数问题使得它的解满足同一个递推关系。
5. 定义一个 k 阶的线性齐次递推关系。
6. a) 解释怎样求解二阶线性齐次递推关系。
b) 如果 $a_0 = 3$, $a_1 = 15$, 对于 $n \geq 2$, 求解递推关系 $a_n = 13a_{n-1} - 22a_{n-2}$ 。
c) 如果 $a_0 = 3$, $a_1 = 35$, 对于 $n \geq 2$, 求解递推关系 $a_n = 14a_{n-1} - 49a_{n-2}$ 。
7. a) 如果 $f(n)$ 满足分治递推关系 $f(n) = af(n/b) + g(n)$, 这里 b 整除正整数 n , 解释怎样求 $f(b^k)$, 其中 k 是正整数。
b) 如果 $f(n) = 3f(n/4) + 5n/4$ 且 $f(1) = 7$, 求 $f(256)$ 。
8. a) 对于用二分搜索在表中找一个数所用的比较次数推导一个分治的递推关系。
b) 从你在(a)给出的分治递推关系使用 4.3 节中的定理 1, 对于二分搜索所用的比较次数给出一个大 O 估计。
9. a) 给出一个关于 3 个集合并集元素数的公式。
b) 解释为什么这个公式是有效的。
c) 解释怎样使用(a)的公式求不超过 1000 且能被 6, 10 或 15 整除的正整数个数。
d) 解释怎样使用(a)的公式求方程 $x_1 + x_2 + x_3 + x_4 = 22$ 的非负整数解的个数, 其中 $x_1 < 8$, $x_2 < 6$, $x_3 < 5$ 。
10. a) 给出一个关于 4 个集合并集元素数的公式。解释为什么它是有效的。
b) 假设 A_1, A_2, A_3 和 A_4 每个集合含 25 个元素, 其中任意 2 个集合的交含 5 个元素, 任 3 个集合的交含 2 个元素, 所有 4 个集合含 1 个公共元素。问在这 4 个集合的并集中有多少个元素?
11. a) 叙述容斥原理。
b) 概述这个原理的证明。
12. 解释怎样使用容斥原理计数从 m 元素集合到 n 元素集合的映上函数的个数。
13. a) 怎样计数把 m 项工作分给 n 个雇员并使得每个雇员至少得到一项工作的方案数?
b) 把 7 项工作分给 3 个雇员并使得每个雇员至少得到一项工作有多少种方案?
14. 解释怎样使用容斥原理计数不超过正整数 n 的素数个数。
15. a) 定义一个错位排列。
b) 一个寄存帽子的人给 n 个人发还帽子, 使得没有人得到自己帽子的方式的计数为什么和 n 个物体的错位排列数一样?
c) 解释怎样计数 n 个物体的错位排列数。

补充练习

1. 一个 10 人小组开始一系列的通信活动, 每个人把这封信寄给另外 4 个人。每个收到信的人再把这封信寄给另外的 4 个人。
a) 如果没有人收到的信多于 1 封, 求与这个通信活动的第 n 步寄出信数有关的递推关系。
b) 在(a)中递推关系的初始条件是什么?
c) 在这个通信活动的第 n 步寄出了多少封信?
2. 一个核反应堆产生了 18 克放射性同位素。每小时放射性同位素衰变 1%。
a) 对 n 小时后留下的同位素量建立一个递推关系。
b) 对于(a)的递推关系, 初始条件是什么?

c) 求解这个递推关系。

3. 美国政府每小时印1美元纸币超过10 000张, 5美元纸币超过4000张, 10美元纸币超过3000张, 20美元纸币超过2500张, 50美元纸币超过1000张, 100美元纸币与前一小时的张数一样。在初始时刻每种钱币有1000张。

a) 建立一个关于第 n 小时总钱数的递推关系。

b) (a)中递推关系的初始条件是什么?

c) 求解这个第 n 小时总钱数的递推关系。

d) 建立一个关于前 n 小时总钱数的递推关系。

e) 求解这个关于前 n 小时总钱数的递推关系。

4. 每个前一小时已经存在的细菌在每小时都分裂出2个新的细菌, 并且所有的细菌只有2小时的生命。假设这群细菌开始时有100个新细菌。

a) 建立关于 n 小时后存在细菌数目的递推关系。

b) 这个递推关系的解是什么?

c) 什么时候这群细菌的个数将超过100万个?

5. 使用两个不同的信号在通信信道发送信息。传送一个信号需要2微秒, 传送另一个信号要3微秒。一个信息的每个信号后紧跟着下一个信号。

a) 求与在 n 微秒中可以发送的不同信号数有关的递推关系。

b) 对于(a)的递推关系, 初始条件是什么?

c) 在12微秒内可以发送多少个不同的信息?

6. 一个小邮局只有4分、6分和10分邮票。如果考虑邮票使用的次序, 求与这些邮票构成 n 分邮资的方式数有关的递推关系。这个递推关系的初始条件是什么?

7. 使用在练习6描述的规则, 有多少种方式构成下述邮资?

a) 12分

b) 14分

c) 18分

d) 22分

8. 求具有 $a_0=1$ 和 $b_0=2$ 的联立方程组

$$a_n = a_{n-1} + b_{n-1}$$

$$b_n = a_{n-1} - b_{n-1}$$

9. 如果 $a_0=1$ 且 $a_1=2$, 求解递推关系 $a_n = a_{n-1}^2 / a_{n-2}$ 。[提示: 两边取对数得到关于序列 $\log a_n$ 的递推关系, $n=0, 1, 2, \dots$]

- *10. 如果 $a_0=2$ 和 $a_1=2$, 求解递推关系 $a_n = a_{n-1}^3 a_{n-2}^2$ 。(见练习9的提示。)

11. 如果 $a_0=2$, $a_1=4$ 和 $a_2=8$, 求解递推关系 $a_n = 3a_{n-1} - 3a_{n-2} + a_{n-3} + 1$ 。

12. 如果 $a_0=2$, $a_1=2$ 和 $a_2=4$, 求解递推关系 $a_n = 3a_{n-1} - 3a_{n-2} + a_{n-3}$ 。

- *13. 假设在4.1节的例4中, 一对兔子在繁殖2次以后就离开这个岛。求与第 n 个月中的岛上兔子对数有关的递推关系。

14. 当 n 可被5整除时求解递推关系 $f(n) = 3f(n/5) + 2n^k$, 其中 $n=5^k$, k 是正整数, $f(1)=1$ 。

15. 如果 f 是增函数, 估计练习14中 f 的大小。

16. 找出与下述算法所使用比较次数有关的递推关系: 通过把 n 个数的序列递归地划分成两个子序列找出最大和第二大的元素, 在每一步划分时要求这两个子序列项数相等或一个子序列比另一个子序列多一项。当子序列达到2项时停止。

17. 估计练习16描述的算法所使用的比较次数。

设 $\{a_n\}$ 是实数序列。这个序列的前向差分定义如下: 第一个前向差分是 $\Delta a_n = a_{n+1} - a_n$; 第 $k+1$ 个前向差分 $\Delta^{k+1} a_n$ 是通过 $\Delta^{k+1} a_n = \Delta^k a_{n+1} - \Delta^k a_n$ 由 $\Delta^k a_n$ 得到的。

18. 求 $\Delta^k a_n$, 其中

a) $a_n = 3$

b) $a_n = 4n + 7$

c) $a_n = n^2 + n + 1$

19. 设 $a_n = 3n^3 + n + 2$, 求 $\Delta^k a_n$, 其中 k 等于

a) 2

b) 3

c) 4

- *20. 假设 $a_n = P(n)$, 其中 P 是 d 次多项式。证明: 对所有的非负整数 n , $\Delta^{d+1} a_n = 0$ 。

21. 令 $\{a_n\}$ 和 $\{b_n\}$ 是实数序列。证明

$$\Delta(a_n b_n) = a_{n+1}(\Delta b_n) + b_n(\Delta a_n)$$

22. 证明: 如果 $F(x)$ 和 $G(x)$ 分别是序列 $\{a_k\}$ 和 $\{b_k\}$ 的生成函数, 且 c 和 d 是实数, 那么 $(cF + dG)(x)$ 是 $\{ca_k + db_k\}$ 的生成函数。

23. (需要微积分)这个练习说明了怎样使用生成函数求解递推关系 $(n+1)a_{n+1} = a_n + (1/n!)$, $n \geq 0$, 初始条件 $a_0 = 1$.
- a) 设 $G(x)$ 是关于 $\{a_n\}$ 的生成函数。证明 $G'(x) = G(x) + e^x$ 且 $G(0) = 1$ 。
- b) 由(a)证明 $(e^{-x}G(x))' = 1$, 且断定 $G(x) = xe^x + e^x$ 。
- c) 使用(b)找出关于 a_n 的封闭公式。
24. 假设在离散数学班的第一次考试中 14 个学生得 A, 第二次考试中 18 个得 A。如果 22 个学生在第一或第二次考试得 A, 有多少学生两次考试都得 A?
25. 在蒙默思郡(英国威尔士郡原郡名)323 个农场至少有马、牛或羊其中的 1 种。如果 224 个农场有马, 85 个有牛, 57 个有羊, 18 个农场 3 种家畜全有, 那么有多少个农场恰好有这 3 种家畜中的 2 种?
26. 查询某学院关于学生记录的数据库得到下述数据: 学院有 2175 个学生, 其中 1675 个不是一年级学生, 1074 个学生选了微积分, 444 个学生选了离散数学, 607 个不是一年级学生且选了微积分, 350 个学生选了微积分和离散数学, 201 个不是一年级学生且选了离散数学, 143 个不是一年级学生并且选了微积分和离散数学。所有这些对查询的回答都是正确的吗?
27. 某大学数学学院的学生可以选择下述一个或多个方向作为主修方向: 应用数学(AM)、纯粹数学(PM)、运筹学(OR)及计算机科学(CS)。如果包括同时主修在内, 主修 AM 的有 23 个学生; 主修 PM 的有 17 个学生; 主修 OR 的 44 个; 主修 CS 的 63 个; 主修 AM 与 PM 的 5 个; 主修 AM 与 CS 的 8 个; 主修 AM 与 OR 的 4 个; 主修 PM 与 CS 的 6 个; 主修 PM 与 OR 的 5 个; 主修 OR 与 CS 的 14 个; 主修 PM, OR 与 CS 的 2 个; 主修 AM, OR 与 CS 的 2 个; 主修 PM, AM 与 OR 的 1 个; 主修 PM, AM 与 CS 的 1 个; 还有 1 个主修所有 4 个方向。问这个学院有多少学生?
28. 当使用容斥原理表示 7 个集合的并集中的元素数时, 如果其中没有 6 个或更多的集合含有公共元素, 那么需要多少项?
29. 方程 $x_1 + x_2 + x_3 = 20$, $2 < x_1 < 6$, $6 < x_2 < 10$, $0 < x_3 < 5$ 有多少个正整数解?
30. 有多少小于 1 000 000 的正整数
- a) 能够被 2, 3 或 5 整除? b) 不被 7, 11 或 13 整除? c) 被 3 整除但不被 7 整除?
31. 有多少小于 200 的正整数是
- a) 整数的 2 次或更高次幂? b) 整数的 2 次或更高次幂, 或是素数?
- c) 不被一个大于 1 的整数的平方整除? d) 不被一个大于 1 的整数的立方整除?
- e) 不被 3 个或更多的素数整除?
- *32. 把 6 个不同的工作分给 3 个不同的雇员, 如果最难的工作分给最有经验的雇员并且最容易的工作分给最缺乏经验的雇员, 那么有多少种分法?
33. 由寄存帽子的人随机发还给 n 个人帽子, 那么恰好一个人拿到自己帽子的概率是多少?
34. 有多少个 6 位二进制串不包含 4 个连续的 1?
35. 一个 6 位二进制串包含至少 4 个 1 的概率是多少?

计算机题目

按下述给定的输入和输出写出程序。

- 给定正整数 n , 列出汉诺塔难题从一根柱子到另一根柱子依照游戏规则移动 n 个盘子需要的所有移动。
- 给定正整数 n 和整数 k , $1 \leq k \leq n$, 列出富雷姆-斯图尔特算法(见 4.1 节练习 27 前面的说明)依照游戏规则用 4 根柱子从一根柱子到另一根柱子移动 n 个盘子需要的所有移动。
- 给定正整数 n , 列出不包含连续 2 个 0 的所有的 n 位二进制序列。
- 给定正整数 n , 写出在 $n+1$ 个变量的乘积中加括号的所有方式。
- 给定递推关系 $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ 和正整数 k , 其中 c_1 和 c_2 是实数, 初始条件为 $a_0 = C_0$ 及 $a_1 = C_1$, 使用迭代求 a_k 。
- 给定递推关系 $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ 和初始条件 $a_0 = C_0$ 及 $a_1 = C_1$, 确定唯一的解。
- 给定形如 $f(n) = af(n/b) + c$ 的递推关系, 其中 a 是实数, b 是正整数, c 是实数, k 是正整数, 使用迭代求 $f(b^k)$ 。
- 给定 3 个集合的交集的元素个数, 每两个集合的交集的元素个数, 和每个集合中的元素个数, 求其并集中的元素个数。

9. 给定正整数 n , 求在 n 个集合的并集中元素个数的公式。
10. 给定正整数 m 和 n , 求从 m 元素集到 n 元素集的映上函数个数。
11. 给定正整数 n , 列出集合 $\{1, 2, 3, \dots, n\}$ 的所有错位排列。

计算和研究

使用一个计算程序或你已完成的程序做下面的练习。

1. 求 f_{100} , f_{500} 和 f_{1000} 的精确值, 其中 f_n 是斐波那契数。
2. 求比 1 000 000 大、比 1 000 000 000 大和比 1 000 000 000 000 大的最小的斐波那契数。
3. 求尽可能多的同为素数的斐波那契数, 目前还不知道是否存在无限多个这样的数。
4. 写出求解 10 个盘子的汉诺塔难题所需要的所有的移动。
5. 按照雷夫难题的规则, 用 4 根柱子从一根柱到另一根柱移动 20 个盘子, 写出使用弗雷姆-斯图尔特算法需要的所有移动。
6. 通过下面的方法验证求解 n 个盘子的雷夫难题的弗雷姆猜想: 对于尽可能多的整数 n , 证明这个难题不可能使用比具有最优选择 k 的弗雷姆-斯图尔特算法还要少的移动来求解。
7. 计算对于各种整数 n , 包括 16, 64, 256 和 1024, 使用在 4.3 节描述的快速乘法和整数相乘的标准算法做两个 n 位整数相乘所需要的运算次数。
8. 计算对于各种整数 n , 包括 4, 16, 64 和 128, 使用在 4.3 节描述的快速矩阵乘法和矩阵相乘的标准算法做两个 $n \times n$ 矩阵相乘所需要的运算次数。
9. 使用埃拉托色尼筛求不超过 1000 的所有素数。
10. 使用在 4.6 节描述的求不超过 100 的素数个数的方法求不超过 10 000 的素数个数。
11. 列出 $\{1, 2, 3, 4, 5, 6, 7, 8\}$ 的所有的错位排列。
12. 对所有不超过 20 的正整数 n 计算 n 个物体的一个排列是错位排列的概率, 并确定这些概率逼近 e 的速度。

写作题目

用本书以外的资料, 按下列要求写成短文。

1. 找出斐波那契发表的关于兔子数模型难题的原始材料。讨论斐波那契提出的这个问题和其他问题, 并且给出关于斐波那契本人的某些信息。
2. 解释斐波那契数怎样在其他应用中出现, 如叶序、植物叶片排列的研究、镜子反射的研究等。
3. 描述汉诺塔难题的各种不同的变形问题, 包括多于 3 个柱子的(包括课本和练习中讨论的雷夫难题在内), 盘子移动受限制的, 以及允许有同样大小盘子的。关于求解每种变形问题所要求的移动次数有什么已知的结论?
4. 尽可能多地讨论出现卡塔兰数的不同的问题。
5. 描述派尔克发现的与一次谎话搜索相关的乌拉姆问题的解。
6. 讨论派尔克发现的与多次谎话搜索相关的乌拉姆问题的变种, 关于这个问题你还知道什么?
7. 讨论求平面上点集的凸包的分治算法。
8. 查询幸运数(lucky number)的定义。解释怎样使用类似于埃拉托色尼筛的技术找出幸运数。求所有小于 1000 的幸运数。
9. 描述在数论中使用的筛法。使用这种方法已经得到了哪些结果?
10. 查询古代法国纸牌相遇游戏的规则。描述这些规则并且描述皮埃尔·雷蒙德·蒙特莫特关于“相遇问题”的论文。
11. 描述怎样使用指数生成函数求解各种计数问题。
12. 描述计数的 Polyá 理论和可使用这个理论求解的计数问题的种类。
13. 管家问题是求安排 n 对夫妇围圆桌就座的方法数, 使得就座时男女相间并且没有丈夫和妻子相邻。解释怎样用卢卡斯(E. Lucas)方法求解这个问题。
14. 解释怎样使用棋盘多项式(rook polynomial)求解计数问题。

关 系

在许多情况下集合的元素之间都存在某种关系。每天我们都要涉及各种关系，例如在一个商行和它的电话号码之间的关系，雇员与其工资之间的关系，一个人与一个亲属之间的关系，等等。在数学中我们研究的关系，如一个正整数与被它除的一个正整数、一个整数与和它模5同余的一个整数、一个实数与一个比它大的实数之间的关系，以及一个实数 x 和它的函数值 $f(x)$ ，等等。在计算机科学中常常出现的关系，如一个程序与它所使用的一个变量、一种计算机语言与这个语言的一个有效语句之间的关系等。

集合的元素之间的关系被表示成一种结构，这种结构叫做关系。它其实是集合间的笛卡儿积的一个子集。可以用关系来求解问题，例如，确定在一个网络中的哪两个城市之间开通航线，为一个复杂课题的不同阶段的工作找一个可行的执行次序，或者产生一个有用的方式以便在计算机数据库中存储信息。

在某些计算机语言中，一个变量名仅是它的前31个字母有效。此关系由前31个字母相同的字符串有序对组成，这是一种被称为等价关系的特殊关系。等价关系在数学和计算机科学中均有体现。本章将要学习等价关系和其他几种特殊关系。

5.1 关系及其性质

5.1.1 引言

 可以用两个相关元素构成的有序对来表达两个集合的元素之间的关系，这是一种最直接的方式。为此，有序对的集合就叫做二元关系。在这一节，我们引入用于描述二元关系的基本术语。在这一章的后面，我们将使用关系来求解涉及通信网络、项目调度以及识别集合中具有共同性质的元素等问题。

定义1 设 A 和 B 是集合，一个从 A 到 B 的二元关系是 $A \times B$ 的子集。

换句话说，一个从 A 到 B 的二元关系是有序对的集合 R ，其中每个有序对的第一个元素取自 A 并且第二个元素取自 B 。我们使用记号 aRb 表示 $(a, b) \in R$ ， $a \not R b$ 表示 $(a, b) \notin R$ 。当 (a, b) 属于 R 时称 a 与 b 有关系 R 。

二元关系表示了两个集合的元素之间的关系。在本章的后面我们将引入 n 元关系，它表示了三个以上集合中元素之间的关系。当不发生混淆时我们将省去二元这个词。

例1~例3说明了关系的概念。

例1 设 A 是你们学校的学生的集合， B 是课程的集合。令 R 是由 (a, b) 对构成的关系，其中 a 是选修课程 b 的学生。例如，如果Jason Goodfriend和Deborah Sherman选修CS518，即离散数学，有序对 $(\text{Jason Goodfriend}, \text{CS518})$ 和 $(\text{Deborah Sherman}, \text{CS518})$ 属于 R 。如果Jason Goodfriend也选修CS510，即数据结构，那么有序对 $(\text{Jason Goodfriend}, \text{CS510})$ 也属于 R 。但是，如果Deborah Sherman没有选修CS510，那么有序对 $(\text{Deborah Sherman}, \text{CS510})$ 不在 R 中。

注意如果一个学生目前没有选修任何课程，那么在 R 中没有以这个学生为第一元素的对。类似地，如果一门课程目前没有开设，那么在 R 中也没有以这门课程作为第二个元素的对。 ■

例2 设 A 是所有城市的集合， B 是美国的50个州的集合。如下定义关系 R ：如果城市 a 是在州 b ，则 (a, b) 属于 R 。例如， $(\text{Boulder}, \text{科罗拉多州})$ ， $(\text{Bangor}, \text{缅因州})$ ， $(\text{Ann Arbor}, \text{密歇根州})$ ， $(\text{Cupertino}, \text{加利福尼亚州})$ 和 $(\text{Red Bank}, \text{新泽西州})$ 是在 R 中。 ■

例3 设 $A=\{0, 1, 2\}$, $B=\{a, b\}$, 那么 $\{(0, a), (0, b), (1, a), (2, b)\}$ 是从 A 到 B 的关系。这意味着, 比如说有 $0Ra$, 但没有 $1Rb$, 关系可以用图来表示, 如图 5-1 所示, 用箭头来表示有序对。另一种表示关系的方式就是用一张表, 这也在图 5-1 给出。在 5.3 节我们将更详细地讨论关系的表示。

5.1.2 函数作为关系

一个从集合 A 到集合 B 的函数 f (如 2.3 节的定义) 对于 A 中的每个元素都指定 B 中一个唯一的元素。 f 的图示是使得 $b=f(a)$ 的有序对 (a, b) 的集合。由于 f 的图示是 $A \times B$ 的子集, 它就是一个从 A 到 B 的关系。此外, 函数的图示有下述性质: A 的每个元素恰好是图中一个有序对的第一元素。

相反, 如果 R 是从 A 到 B 的关系, 并且使得 A 中的每个元素是 R 中恰好一个有序对的第一元素, 那么 R 的图示就可以定义一个函数。只要对 A 的每个元素指定唯一的元素 $b \in B$ 使得 $(a, b) \in R$ 就可以做到。

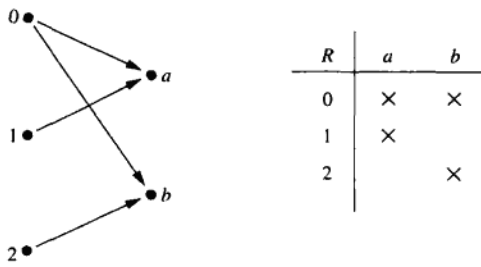


图 5-1 在例 3 的关系 R 中的有序对

可以用关系表达在集合 A 和集合 B 之间的一对多的相关性, 其中 A 的一个元素可以与 B 中多个元素有关。函数表示了这样一种关系, 对于 A 中的每个元素恰好只有一个 B 中的元素与之相关。

关系是函数的推广, 可以用关系表示集合之间的更为广泛的联系。

5.1.3 集合的关系

集合 A 到它自身的关系是特别令人感兴趣的。

定义 2 集合 A 的关系是从 A 到 A 的关系。

换句话说, 集合 A 的关系是 $A \times A$ 的子集。

例 4 设 A 是集合 $\{1, 2, 3, 4\}$, A 上的关系 $R=\{(a, b) \mid a \text{ 整除 } b\}$ 中有哪些有序对?

解 (a, b) 在 R 中, 当且仅当 a 和 b 是不超过 4 的正整数且 a 整除 b , 我们看到

$$R = \{(1,1), (1,2), (1,3), (1,4), (2,2), (2,4), (3,3), (4,4)\}$$

在这个关系中的有序对的图和表的表示都在图 5-2 中给出。

例 5 给出了某些整数集合上的关系的实例。

例 5 考虑下面这些整数集合上的关系:

$$R_1 = \{(a, b) \mid a \leq b\}$$

$$R_2 = \{(a, b) \mid a > b\}$$

$$R_3 = \{(a, b) \mid a = b \text{ 或 } a = -b\}$$

$$R_4 = \{(a, b) \mid a = b\}$$

$$R_5 = \{(a, b) \mid a = b + 1\}$$

$$R_6 = \{(a, b) \mid a + b \leq 3\}$$

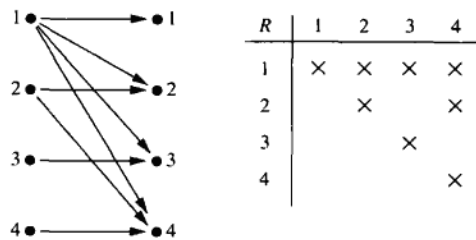


图 5-2 例 4 中关系 R 的有序对

其中, 哪些关系包含了有序对 $(1, 1)$, $(1, 2)$, $(2, 1)$, $(1, -1)$ 以及 $(2, 2)$?

注意 和例 1~例 4 的关系不同, 这些是无穷集合上的关系。

解 有序对 $(1, 1)$ 在 R_1 , R_3 , R_4 和 R_6 中; 有序对 $(1, 2)$ 在 R_1 和 R_6 中; 有序对 $(2, 1)$ 在 R_2 , R_5 和 R_6 中; $(1, -1)$ 在 R_2 , R_3 和 R_6 中; 最后, 有序对 $(2, 2)$ 在 R_1 , R_3 和 R_4 中。

不难确定有穷集上的关系个数, 因为集合 A 上的关系仅仅是 $A \times A$ 的子集。

例 6 n 元素集合有多少个关系?

解 集合 A 上的关系是 $A \times A$ 的子集。因为当 A 是 n 元素集合时 $A \times A$ 有 n^2 个元素, 并且 m

个元素的集合有 2^m 个子集, 故 $A \times A$ 的子集有 2^n 个。于是 n 元素集合有 2^n 个关系。例如, 在集合 $\{a, b, c\}$ 上存在 $2^{3^2} = 2^9 = 512$ 个关系。

5.1.4 关系的性质

有若干个用于把集合上的关系分类的性质。这里我们只介绍其中最重要的性质。

在某些关系中一个元素总是与自己相关。例如, 设 R 是所有的人的集合上的关系, 若 x 和 y 有相同的母亲和相同的父亲, 那么 (x, y) 属于 R 。于是对于每个人 x , 有 xRx 。

定义 3 如果对每个元素 $a \in A$ 有 $(a, a) \in R$, 那么集合 A 上的关系 R 叫做自反的。

注意 我们看到如果 $\forall a((a, a) \in R)$, 那么集合上的关系是自反的, 这里的论域是 A 中所有元素的集合。

我们看到如果 A 的每个元素都关系到它自己, A 上的关系就是自反的。例 7~9 说明了自反关系的概念。

例 7 考虑下面 $\{1, 2, 3, 4\}$ 上的关系

$$R_1 = \{(1, 1), (1, 2), (2, 1), (2, 2), (3, 4), (4, 1), (4, 4)\}$$

$$R_2 = \{(1, 1), (1, 2), (2, 1)\}$$

$$R_3 = \{(1, 1), (1, 2), (1, 4), (2, 1), (2, 2), (3, 3), (4, 1), (4, 4)\}$$

$$R_4 = \{(2, 1), (3, 1), (3, 2), (4, 1), (4, 2), (4, 3)\}$$

$$R_5 = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4, 4)\}$$

$$R_6 = \{(3, 4)\}$$

其中哪些是自反的?

解 关系 R_1 和 R_3 是自反的, 因为它们都包含了所有形如 (a, a) 的对, 即 $(1, 1)$, $(2, 2)$, $(3, 3)$ 和 $(4, 4)$ 。其他的不是自反的, 因为它们不包含所有这些有序对。特别地, R_2 , R_4 和 R_6 不是自反的, 因为 $(3, 3)$ 都不在这些关系里。

例 8 例 5 中哪些关系是自反的?

解 这个例子中的自反关系是 R_1 (因为对每个整数 a 有, $a \leq a$), R_3 和 R_4 。对于这个例子中其他的关系容易找到不在关系中的形如 (a, a) 的对 (留给读者作为练习)。

例 9 正整数集合上的“整除”关系是自反的吗?

解 因为只要 a 是正整数就有 $a \mid a$, “整除”关系是自反的。(注意, 如果我们将正整数集替换为所有整数集, 则“整除”关系不是自反的, 因为 0 不能整除 0。)

在某些关系中一个元素与第二个元素相关, 当且仅当第二个元素也与第一个元素相关。比如一个关系由 (x, y) 对构成, 其中 x 和 y 是你校的学生并且他们至少学一门公共课程。这个关系就有这种性质。而某些关系有另一种性质, 即如果一个元素与第二个元素相关, 那么第二个元素就不与第一个元素相关。比如一个关系由 (x, y) 对构成, 其中 x 和 y 是你校的学生, 且 x 比 y 的平均成绩高。这个关系就有后一种性质。

定义 4 对于 $a, b \in A$, 如果只要 $(a, b) \in R$ 就有 $(b, a) \in R$, 则集合 A 上的关系 R 叫做对称的。如果对 $a, b \in A$, 仅当 $a = b$ 时 $(a, b) \in R$ 和 $(b, a) \in R$, 则集合 A 上的关系 R 叫做反对称的。

注意 使用量词, 可以看到如果 $\forall a \forall b((a, b) \in R \rightarrow (b, a) \in R)$, 则 A 上的关系 R 是对称的。类似地, 如果 $\forall a \forall b((a, b) \in R \wedge (b, a) \in R \rightarrow (a = b))$, 则 A 上的关系 R 是反对称的。

就是说, 关系 R 是对称的, 当且仅当如果 a 与 b 相关则 b 与 a 就相关。关系 R 是反对称的, 当且仅当不存在由不同元素 a 和 b 构成的有序对使得 a 与 b 相关并且 b 与 a 也相关。对称与反对称的概念不是对立的, 因为一个关系可以同时有这两种性质或者两种性质都没有。一个关系如果包含了某些形如 (a, b) 的对, 其中 $a \neq b$, 这个关系就不可能同时是对称和反对称的。

注意 尽管从统计数据中可以看出, 对称的或反对称的关系相对不多, 但许多重要的关系都具有两种性质其中之一(见练习 21)。

例 10 例 7 中的哪些关系是对称的? 哪些是反对称的?

解 关系 R_2 和 R_3 是对称的, 因为在这些关系中, 只要 (a, b) 属于关系就有 (b, a) 属于关系。对于关系 R_2 , 唯一需要检查的就是 $(1, 2)$ 和 $(2, 1)$ 属于这个关系。对于 R_3 , 必须要检查 $(1, 2)$ 和 $(2, 1)$ 属于这个关系, 还有 $(1, 4)$ 和 $(4, 1)$ 也属于这个关系。读者应该能够验证其他的系统中没有一个是反对称的, 只需找到一个有序对 (a, b) , 使得它在关系中但 (b, a) 不在关系中。

R_1 、 R_5 和 R_6 都是反对称的。其中每一个关系都不存在这样的有序对, 即它由元素 a 和 b 构成, 使得 $a \neq b$, 但 (a, b) 和 (b, a) 都属于这个关系。读者只需找到有序对 (a, b) 满足 $a \neq b$, 但 (a, b) 和 (b, a) 都属于这个关系, 就能够验证其他系统中没有一个是反对称的。 ■

例 11 例 5 中的哪些关系是对称的? 哪些是反对称的?

解 关系 R_3 、 R_4 和 R_6 是对称的。 R_3 是对称的, 因为如果 $a=b$ 或 $a=-b$, 那么就有 $b=a$ 或 $b=-a$ 。 R_4 是对称的, 因为 $a=b$ 推出 $b=a$ 。 R_6 是对称的, 因为 $a+b \leq 3$ 推出 $b+a \leq 3$ 。读者应该能验证其他关系没有一个是反对称的。

关系 R_1 、 R_2 、 R_5 和 R_7 是反对称的。 R_1 是反对称的, 因为不等式 $a \leq b$ 和 $b \leq a$ 推出 $a=b$ 。 R_2 是反对称的, 因为 $a > b$ 和 $b > a$ 是不可能的。 R_5 是反对称的, 因为两个元素相对于 R_5 相关, 当且仅当它们是相等的。 R_7 是反对称的, 因为 $a=b+1$ 和 $b=a+1$ 是不可能的。读者应该能验证其他关系没有一个是反对称的。 ■

例 12 正整数集合上的整除关系是对称的吗? 是反对称的吗?

解 这个关系不是对称的, 因为 $1 \mid 2$, 但 $2 \nmid 1$ 。它是反对称的, 因为如果 a 和 b 是正整数且 $a \mid b$ 和 $b \mid a$, 那么 $a=b$ (这个验证留给读者作为练习)。 ■

设 R 是有序对 (x, y) 构成的关系, 其中 x 与 y 是你校的学生, 且 x 比 y 得到更多的学分。假设 x 与 y 相关并且 y 与 z 相关, 这意味着 x 比 y 得到更多的学分并且 y 比 z 得到更多的学分。可以断言 x 比 z 得到更多的学分, 因此 x 与 z 相关。我们证明了 R 有传递性, 这个性质定义如下。

定义 5 如果对于 $a, b, c \in A$, $(a, b) \in R$ 并且 $(b, c) \in R$ 则 $(a, c) \in R$, 那么集合 A 上的关系 R 叫做传递的。

注意 使用量词, 如果 $\forall a \forall b \forall c ((a, b) \in R \wedge (b, c) \in R) \rightarrow (a, c) \in R$, 集合 A 上的关系就叫做传递的。

例 13 例 7 中的关系哪些是传递的?

解 R_4 、 R_5 和 R_6 是传递的。对于这些关系, 我们可以通过验证若 (a, b) 和 (b, c) 属于这个关系则 (a, c) 也属于这个关系来证明每个关系都是传递的。例如, R_4 是传递的, 因为只有 $(3, 2)$ 和 $(2, 1)$, $(4, 2)$ 和 $(2, 1)$, $(4, 3)$ 和 $(3, 1)$, 以及 $(4, 3)$ 和 $(3, 2)$ 是这种有序对, 而 $(3, 1)$, $(4, 1)$ 和 $(4, 2)$ 属于 R_4 。读者应该能验证 R_5 和 R_6 也是传递的。

R_1 不是传递的, 因为 $(3, 4)$ 和 $(4, 1)$ 属于 R_1 , 但 $(3, 1)$ 不属于 R_1 。 R_2 不是传递的, 因为 $(2, 1)$ 和 $(1, 2)$ 属于 R_2 , 但 $(2, 2)$ 不属于 R_2 。 R_3 不是传递的, 因为 $(4, 1)$ 和 $(1, 2)$ 属于 R_3 , 但 $(4, 2)$ 不属于 R_3 。 ■

例 14 例 5 中的哪些关系是传递的?

解 关系 R_1 、 R_2 、 R_3 和 R_4 是传递的。 R_1 是传递的, 因为 $a \leq b$ 和 $b \leq c$ 推出 $a \leq c$ 。 R_2 是传递的, 因为 $a > b$ 和 $b > c$ 推出 $a > c$ 。 R_3 是传递的, 因为 $a = \pm b$ 和 $b = \pm c$ 推出 $a = \pm c$ 。正如读者应该验证的, 显然 R_4 也是传递的。 R_5 不是传递的, 因为 $(2, 1)$ 和 $(1, 0)$ 属于 R_5 , 但 $(2, 0)$ 不属于 R_5 。 R_6 不是传递的, 因为 $(2, 1)$ 和 $(1, 2)$ 属于 R_6 , 但 $(2, 2)$ 不属于 R_6 。 ■

例 15 正整数集合上的“整除”关系是传递的吗?

解 假设 a 整除 b 且 b 整除 c , 那么存在正整数 k 和 l 使得 $b = ak$ 和 $c = bl$, 因此 $c = a(kl)$, 即 a 整除 c 。从而证明了这个关系是传递的。 ■

可以使用计数技术确定具有特殊性质的关系的个数。由此可以得知：这个性质在 n 元素集合的所有关系的集合上有多普遍。

例 16 n 元素集合上有多少个自反的关系？

解 A 上的关系 R 是 $A \times A$ 的子集。因此，要通过指定 $A \times A$ 中 n^2 个有序对中的每一个是否在 R 中来确定关系。因而，如果 R 是自反的，对于 $a \in A$ ， n 个有序对 (a, a) 中的每一个都必须在 R 中。其他 $n(n-1)$ 个形如 (a, b) 的有序对， $a \neq b$ ，可能在也可能不在 R 中。因此，由计数的乘积法则，存在 $2^{n(n-1)}$ 个自反的关系。[这就是选择具有 $a \neq b$ 的每个元素 (a, b) 是否属于 R 的方式数。]

n 元素集合上的对称关系和反对称关系数可以用和例 16 类似的推理得出(见本节末的练习 21)。计数 n 元素集合上的传递关系数的问题已超出本书的范围。

5.1.5 关系的组合

因为从 A 到 B 的关系是 $A \times B$ 的子集，可以按照两个集合组合的任何方式来组合两个从 A 到 B 的关系。考虑例 17 至例 19。

例 17 设 $A = \{1, 2, 3\}$ 和 $B = \{1, 2, 3, 4\}$ 。组合关系 $R_1 = \{(1, 1), (2, 2), (3, 3)\}$ 和 $R_2 = \{(1, 1), (1, 2), (1, 3), (1, 4)\}$ 可以得到

$$R_1 \cup R_2 = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (3, 3)\}$$

$$R_1 \cap R_2 = \{(1, 1)\}$$

$$R_1 - R_2 = \{(2, 2), (3, 3)\}$$

$$R_2 - R_1 = \{(1, 2), (1, 3), (1, 4)\}$$

例 18 设 A 和 B 分别是学校的所有学生和所有课程的集合。假设 R_1 由所有有序对 (a, b) 组成，其中 a 是选修课程 b 的学生。 R_2 由所有的有序对 (a, b) 构成，其中课程 b 是 a 的必修课。关系 $R_1 \cup R_2$ ， $R_1 \cap R_2$ ， $R_1 \oplus R_2$ ， $R_1 - R_2$ 和 $R_2 - R_1$ 是什么？

解 关系 $R_1 \cup R_2$ 由所有的有序对 (a, b) 组成，其中 a 是一个学生，课程 b 要么是他的选修课，要么是他的必修课。 $R_1 \cap R_2$ 是有序对 (a, b) 的集合，其中 a 是一个学生，他选修了课程 b 并且课程 b 也是 a 的必修课。 $R_1 \oplus R_2$ 由所有的有序对 (a, b) 组成，其中学生 a 已经选修了课程 b 但课程 b 不是 a 的必修课，或者课程 b 是 a 的必修课，但是 a 没有选修它。 $R_1 - R_2$ 是所有对 (a, b) 的集合，其中 a 已经选修了课程 b ，但 b 不是 a 的必修课，即 b 是 a 的选修课。 $R_2 - R_1$ 是所有对 (a, b) 的集合，其中 b 是 a 的必修课，但 a 没有选修它。

例 19 设 R_1 是实数集合上的“小于”关系， R_2 是实数集合上的“大于”关系，即 $R_1 = \{(x, y) \mid x < y\}$ 和 $R_2 = \{(x, y) \mid x > y\}$ 。 $R_1 \cup R_2$ ， $R_1 \cap R_2$ ， $R_1 - R_2$ ， $R_2 - R_1$ ， $R_1 \oplus R_2$ 是什么？

解 注意到 $(x, y) \in R_1 \cup R_2$ 当且仅当 $(x, y) \in R_1$ 或 $(x, y) \in R_2$ ，因此 $(x, y) \in R_1 \cup R_2$ 当且仅当 $x < y$ 或 $x > y$ 。由于条件 $x < y$ 或 $x > y$ 与条件 $x \neq y$ 一样，因此 $R_1 \cup R_2 = \{(x, y) \mid x \neq y\}$ 。换句话说，“小于”关系与“大于”关系的并是“不相等”关系。

其次，注意到一个对 (x, y) 同时属于 R_1 和 R_2 是不可能的，因为 $x < y$ 并且 $x > y$ 是不可能的。从而得到 $R_1 \cap R_2 = \emptyset$ 。也可以看到 $R_1 - R_2 = R_1$ ， $R_2 - R_1 = R_2$ ， $R_1 \oplus R_2 = R_1 \cup R_2 - R_1 \cap R_2 = \{(x, y) \mid x \neq y\}$ 。

组合关系有另一种方式，这种方式和函数的复合相似。

定义 6 设 R 是从集合 A 到集合 B 的关系， S 是从集合 B 到集合 C 的关系。 R 和 S 的合成是由有序对 (a, c) 构成的关系，其中 $a \in A$ ， $c \in C$ ，并且对于它们存在一个元素 $b \in B$ 使得 $(a, b) \in R$ 和 $(b, c) \in S$ 。我们用 $S \circ R$ 表示 R 与 S 的合成。

计算两个关系的合成需要我们找出这些元素，它们既是第一个关系中的有序对的第二个元素，也是第二个关系中的有序对的第一个元素。正如例 20 和例 21 所示。

⊖ 原书此处为 $R_1 \cup R_2 - R_1 \cap R_2$ ，译文根据本书第 1 章的定义加了括号。——译者注

例 20 R 是从 $\{1, 2, 3\}$ 到 $\{1, 2, 3, 4\}$ 的关系且 $R = \{(1, 1), (1, 4), (2, 3), (3, 1), (3, 4)\}$, S 是从 $\{1, 2, 3, 4\}$ 到 $\{0, 1, 2\}$ 的关系且 $S = \{(1, 0), (2, 0), (3, 1), (3, 2), (4, 1)\}$, R 与 S 的合成是什么?

解 $S \circ R$ 是由所有 R 中的有序对和 S 中的有序对构成的, 其中在 R 中有序对的第二元素与 S 中有序对的第一元素相同。例如, R 中的有序对 $(2, 3)$ 和 S 中的有序对 $(3, 1)$ 产生了 $S \circ R$ 中的有序对 $(2, 1)$ 。计算所有在 $R \circ S$ 中的有序对, 我们得到

$$S \circ R = \{(1, 0), (1, 1), (2, 1), (2, 2), (3, 0), (3, 1)\}$$

例 21 双亲关系与自身的合成 设 R 是所有人集合上的双亲关系, 即如果 a 是 b 的父母, 则 $(a, b) \in R$ 。当且仅当存在一个人 b 使得 a 是 b 的父母且 b 是 c 的父母。换句话说, $(a, c) \in R \circ R$ 当且仅当 a 是 c 的祖父母或外祖父母。

从两个关系合成的定义可以递归地来定义关系 R 的幂。

定义 7 设 R 是集合 A 上的关系。幂 R^n , $n=1, 2, 3, \dots$, 递归地定义为

$$R^1 = R \quad \text{和} \quad R^{n+1} = R^n \circ R$$

这个定义证明了 $R^2 = R \circ R$, $R^3 = R^2 \circ R = (R \circ R) \circ R$, 等等。

例 22 设 $R = \{(1, 1), (2, 1), (3, 2), (4, 3)\}$, 求幂 R^n , $n=2, 3, 4, \dots$ 。

解 因为 $R^2 = R \circ R$, 我们得到 $R^2 = \{(1, 1), (2, 1), (3, 1), (4, 2)\}$ 。进一步, 因为 $R^3 = R^2 \circ R$, $R^3 = \{(1, 1), (2, 1), (3, 1), (4, 1)\}$ 。其他的计算证明了 R^4 和 R^3 一样, 因此 $R^4 = \{(1, 1), (2, 1), (3, 1), (4, 1)\}$ 。从而对 $n=5, 6, 7, \dots$, 有 $R^n = R^3$ 。读者应该能够验证这个结果。

下面的定理证明一个传递关系的幂是该关系的子集。5.4 节将要用到这一结果。

定理 1 集合 A 上的关系 R 是传递的, 当且仅当对 $n=1, 2, 3$, 有 $R^n \subseteq R$ 。

证 首先证明定理的充分条件。假设对 $n=1, 2, 3$, 有 $R^n \subseteq R$ 。特别地, 有 $R^2 \subseteq R$ 。为证明 R 的传递性, 注意到如果 $(a, b) \in R$ 并且 $(b, c) \in R$, 根据合成定义就有 $(a, c) \in R^2$ 。因为 $R^2 \subseteq R$, 这就意味着 $(a, c) \in R$ 。因此 R 是传递的。

我们将使用数学归纳法证明定理的必要条件。对于 $n=1$, 定理的这个结果是显而易见的。

假定 $R^n \subseteq R$, 其中 n 是一个正整数, 这是归纳假设。为完成归纳步骤, 必须证明这将推出 R^{n+1} 也是 R 的子集。为证明这一点, 假设 $(a, b) \in R^{n+1}$, 那么因为 $R^{n+1} = R^n \circ R$, 存在元素 $x \in A$ 使得 $(a, x) \in R$ 并且 $(x, b) \in R^n$ 。由归纳假设, 即 $R^n \subseteq R$, 推出 $(x, b) \in R$ 。下一步, 因为 R 是传递的, 以及 $(a, x) \in R$ 和 $(x, b) \in R$, 得到 $(a, b) \in R$ 。这就证明了 $R^{n+1} \subseteq R$, 从而完成了证明。 $\square$

练习

- 列出从 $A = \{0, 1, 2, 3, 4\}$ 到 $B = \{0, 1, 2, 3\}$ 的关系 R 中的有序对, 其中 $(a, b) \in R$, 当且仅当
 - $a = b$
 - $a + b = 4$
 - $a > b$
 - $a \mid b$
 - $\gcd(a, b) = 1$
 - $\text{lcm}(a, b) = 2$
- 对集合 $\{1, 2, 3, 4\}$ 上的如下每一个关系, 确定它是否是自反的, 是否是对称的, 是否是反对称的, 是否是传递的。
 - $\{(2, 2), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4)\}$
 - $\{(1, 1), (1, 2), (2, 1), (2, 2), (3, 3), (4, 4)\}$
 - $\{(2, 4), (4, 2)\}$
 - $\{(1, 2), (2, 3), (3, 4)\}$
 - $\{(1, 1), (2, 2), (3, 3), (4, 4)\}$
 - $\{(1, 3), (1, 4), (2, 3), (2, 4), (3, 1), (3, 4)\}$
- 确定所有 Web 页上的关系 R 是否为自反的、对称的、反对称的和传递的, 其中 $(a, b) \in R$, 当且仅当
 - 每个访问 Web 页 a 的人也访问了 Web 页 b 。
 - 在 Web 页 a 和 b 上没有公共链接。
 - 在 Web 页 a 和 b 上至少有一条公共链接。
 - 存在一个 Web 页包含了到 Web 页 a 和 Web 页 b 的链接。

4. 确定所有整数集合上的关系 R 是否是自反的、对称的、反对称的和传递的, 其中 $(x, y) \in R$, 当且仅当
- a) $x \neq y$ b) $xy \geq 1$ c) $x = y + 1$ 或 $x = y - 1$ d) $x \equiv y \pmod{7}$
 e) x 是 y 的倍数 f) x 与 y 都是负的或都是非负的 g) $x = y^2$ h) $x \geq y^2$

如果对于每个 $a \in A$, 有 $(a, a) \notin R$, 那么集合 A 上的关系 R 是反自反的, 即如果没有 A 中的元素与自己有关系, 关系 R 就是反自反的。

5. 练习 2 的哪些关系是反自反的?
 6. 练习 3 的哪些关系是反自反的?
 7. 集合上的关系可能既不是自反的也不是反自反的吗?
 8. 给出在所有人的集合上一个反自反关系的例子。

一个关系 R 叫做非对称的, 如果 $(a, b) \in R$ 推出 $(b, a) \notin R$ 。

9. 使用量词表达一个关系是非对称的含义。
 10. 从 m 元素集合到 n 元素集合有多少个不同的关系?

■ 设 R 是从集合 A 到集合 B 的关系。从 B 到 A 的逆关系是有序对的集合 $\{(b, a) \mid (a, b) \in R\}$, 记作 R^{-1} , 补关系 $\bar{R}$ 是有序对的集合 $\{(a, b) \mid (a, b) \notin R\}$ 。

11. 设 R 是正整数集合上的关系 $R = \{(a, b) \mid a \text{ 整除 } b\}$, 求
 a) R^{-1} b) $\bar{R}$
 12. 设从 A 到 B 的函数 f 是一一对应的。令 R 是和 f 的图相等的关系, 即 $R = \{(a, f(a)) \mid a \in A\}$ 。逆关系 R^{-1} 是什么?
 13. 设 A 是你校的学生集合, B 是学校图书馆的书的集合。设 R_1 和 R_2 都是有序对 (a, b) 构成的关系, 在 R_1 中, 学生 a 修一门课程需要读书 b , 在 R_2 中, 学生 a 已经读过书 b 。描述在下面每个关系中的有序对。
 a) $R_1 \cup R_2$ b) $R_1 \cap R_2$ c) $R_1 \oplus R_2$ d) $R_1 - R_2$ e) $R_2 - R_1$
 14. 设 R 是由人的集合上的有序对 (a, b) 组成的关系, 其中 a 是 b 的父母。设 S 是由人的集合上的有序对 (a, b) 组成的关系, 其中 a 是 b 的兄弟姐妹。 $S \circ R$ 和 $R \circ S$ 是什么关系?

练习 15~16 涉及的都是实数集合上的关系:

$R_1 = \{(a, b) \in \mathbb{R}^2 \mid a > b\}$, “大于”关系
 $R_2 = \{(a, b) \in \mathbb{R}^2 \mid a \geq b\}$, “大于或等于”关系
 $R_3 = \{(a, b) \in \mathbb{R}^2 \mid a < b\}$, “小于”关系
 $R_4 = \{(a, b) \in \mathbb{R}^2 \mid a \leq b\}$, “小于或等于”关系
 $R_5 = \{(a, b) \in \mathbb{R}^2 \mid a = b\}$, “等于”关系
 $R_6 = \{(a, b) \in \mathbb{R}^2 \mid a \neq b\}$, “不等于”关系

15. 求
 a) $R_2 \cup R_4$ b) $R_3 \cup R_5$ c) $R_3 \cap R_5$ d) $R_4 \cap R_6$
 e) $R_3 - R_6$ f) $R_6 - R_3$ g) $R_2 \oplus R_5$ h) $R_3 \oplus R_5$
 16. 求
 a) $R_2 \circ R_1$ b) $R_2 \circ R_2$ c) $R_3 \circ R_5$ d) $R_4 \circ R_1$
 e) $R_5 \circ R_3$ f) $R_3 \circ R_6$ g) $R_4 \circ R_6$ h) $R_6 \circ R_6$
 17. 设 R 是具有博士学位的人的集合上的关系, 要求 $(a, b) \in R$, 当且仅当 a 是 b 的导师。什么情况下一个有序对 (a, b) 在 R^2 中? 什么情况下一个有序对 (a, b) 在 R^n 中? 这里 n 是正整数。(注意每个具有博士学位的人有一个导师。)
 18. 设 R_1 和 R_2 分别是整数集合上的“模 3 同余”和“模 4 同余”关系, 即 $R_1 = \{(a, b) \mid a \equiv b \pmod{3}\}$ 和 $R_2 = \{(a, b) \mid a \equiv b \pmod{4}\}$ 。求
 a) $R_1 \cup R_2$ b) $R_1 \cap R_2$ c) $R_1 - R_2$ d) $R_2 - R_1$ e) $R_1 \oplus R_2$
 19. $\{0, 1\}$ 上的 16 个不同的关系中有多少个包含了对 $(0, 1)$?
 20. a) 在集合 $\{a, b, c, d\}$ 上有多少个关系?
 b) 在集合 $\{a, b, c, d\}$ 上有多少个关系包含对 (a, a) ?
 *21. n 元素集合上有多少个关系是
 a) 对称的? b) 反对称的? c) 非对称的? d) 反自反的? e) 自反的和对称的?

f) 既不是自反的也不是反自反的?

22. 找出在下面定理证明中的错误。

“定理”: 设 R 是集合 A 上的对称和传递的关系, 则 R 是自反的。

“证明”: 设 $a \in A$ 。取元素 $b \in A$ 使得 $(a, b) \in R$ 。由于 R 是对称的, 因而有 $(b, a) \in R$ 。现在使用传递性, 由 $(a, b) \in R$ 和 $(b, a) \in R$ 可以得出结论 $(a, a) \in R$ 。

23. 证明: 集合 A 上的关系 R 是对称的, 当且仅当 $R = R^{-1}$, 其中 R^{-1} 是 R 的逆关系。

24. 证明: 集合 A 上的关系 R 是自反的, 当且仅当逆关系 R^{-1} 是自反的。

25. 设 R 是自反的和传递的关系, 证明对所有的正整数 n , $R^n = R$ 。

26. 设 R 是集合 A 上的自反关系, 证明对所有的正整数 n , R^n 也是自反的。

27. 假设关系 R 是反自反的, R^2 一定是反自反的吗? 对你的答案给出理由。

5.2 n 元关系及其应用

5.2.1 引言

在两个以上集合的元素中常常会产生某种关系。例如, 学生的姓名、学生的专业以及学生的平均学分绩点之间的关系。类似地, 一个航班的航空公司、航班号、出发地、目的地、起飞时间和到达时间等也有一种关系。在数学中也有这种关系。例如, 有 3 个整数, 其中第一个整数比第二个整数大, 而第二个整数比第三个整数大。另一个例子是直线上的点之间的关系, 即当第二个点在第一和第三个点之间时, 这三个点有关系。

本节我们将研究两个以上集合的元素之间的关系。这种关系叫做 n 元关系。可以用这种关系表示计算机的数据库。这种表示有助于回答对数据库中所存信息的查询, 例如: 哪个航班在午夜 3 点到 4 点之间降落在 O'Hare 机场? 你们学校的二年级学生哪些是主修数学或计算机科学的, 并且平均学分绩点大于 3.0? 公司的哪些雇员为这个公司工作不到 5 年但所赚的钱已经超过 50 000 美元?

5.2.2 n 元关系

我们从建立关系数据库理论的基本定义开始。

定义 1 设 $A_1, A_2, \dots, A_n$ 是集合。在这些集合上的 n 元关系是 $A_1 \times A_2 \times \dots \times A_n$ 的子集。这些集合 $A_1, A_2, \dots, A_n$ 叫做关系的域, n 叫做它的阶。

例 1 R 是 $\mathbf{N} \times \mathbf{N} \times \mathbf{N}$ 上的三元组 (a, b, c) 构成的关系, 其中 a, b, c 是满足 $a < b < c$ 的整数。那么 $(1, 2, 3) \in R$, 但 $(2, 4, 3) \notin R$ 。这个关系的阶是 3。它的域都等于整数集合。

例 2 设 R 是 $\mathbf{Z} \times \mathbf{Z} \times \mathbf{Z}$ 上的三元组 (a, b, c) 构成的关系, 其中的 a, b, c 构成等差数列, 即 $(a, b, c) \in R$, 当且仅当存在一个整数 k , 使得 $b = a + k, c = a + 2k$, 或者 $b - a = k, c - b = k$ 。可知 $(1, 3, 5) \in R$, 因为 $3 = 1 + 2, 5 = 1 + 2 \cdot 2$, 但是 $(2, 5, 9) \notin R$, 因为 $5 - 2 = 3$, 而 $9 - 5 = 4$ 。这个关系的阶为 3, 且它的所有域均等于整数集。

例 3 设 R 是 $\mathbf{Z} \times \mathbf{Z} \times \mathbf{Z}^+$ 上的三元组 (a, b, m) 构成的关系, 其中的 a, b, m 都是整数, 且满足 $m \geq 1$, 和 $a \equiv b \pmod{m}$ 。则 $(8, 2, 3), (-1, 9, 5)$ 和 $(14, 0, 7)$ 都属于 R , 但 $(7, 2, 3), (-2, -8, 5)$ 和 $(11, 0, 6)$ 都不属于 R , 因为 $8 \not\equiv 2 \pmod{3}, -1 \not\equiv 9 \pmod{5}$ 和 $14 \not\equiv 0 \pmod{7}$, 而 $7 \not\equiv 2 \pmod{3}, -2 \not\equiv -8 \pmod{5}$ 和 $11 \not\equiv 0 \pmod{6}$ 。这个关系的阶为 3, 且它的前两个域是全部整数集而第三个域为正整数集。

例 4 设 R 是由 5 元组 (A, N, S, D, T) 构成的表示飞机航班的关系, 其中 A 是航空公司, N 是航班号, S 是出发地, D 是目的地, T 是起飞时间。例如, 如果 Nadir 直达航空公司在 15:00 有从 Newark 到 Bangor 的 963 航班, 那么 $(\text{Nadir}, 963, \text{Newark}, \text{Bangor}, 15:00)$ 属于 R 。这个关系的阶是 5, 并且它的域是所有航空公司的集合、航班号的集合、城市的集合、城市的集合以及时间的集合。

5.2.3 数据库和关系

操作数据库信息所需要的时间依赖于这些信息是怎样存储的。插入和删除记录, 更新记录, 检索记录以及从一个重叠的数据库中组合记录的操作, 在一个大型数据库中每

天要执行几百万次。由于这些操作的重要性，已经开发了数据库表示的各种方法。我们将讨论其中的一种基于关系概念的方法，叫做关系数据模型。

数据库由记录组成，这些记录是由字段构成的 n 元组。这些字段是 n 元组的数据项。例如，学生记录的数据库可以由包含学生的姓名、学号、专业、平均学分绩点(GPA)的字段构成。关系数据模型把一个记录的数据库表示成一个 n 元关系。于是，学生记录可以被表示成形如(学生姓名，学号，专业，GPA)的 4 元组。6 个记录的一个数据库样本是：

- (Ackermann, 231455, 计算机科学, 3.88)
- (Adams, 888323, 物理学, 3.45)
- (Chou, 102147, 计算机科学, 3.49)
- (Goodfriend, 453876, 数学, 3.45)
- (Rao, 678543, 数学, 3.90)
- (Stevens, 786576, 哲学, 2.99)

用于表示数据库的关系叫做表，因为这些关系常常用表来给出。这些表的每个列对应于数据库的一个属性。例如，同样的学生数据库显示在表 5-1。这个数据库的属性是学生姓名、学号、专业和平均学分绩点(GPA)。

表 5-1 学生

学生姓名	学号	专业	GPA
Ackermann	231 455	计算机科学	3.88
Adams	888 323	物理学	3.45
Chou	102 147	计算机科学	3.49
Goodfriend	453 876	数学	3.45
Rao	678 543	数学	3.90
Stevens	786 576	心理学	2.99

当 n 元组的某个域的值能够确定这个 n 元组时， n 元关系的这个域就叫做主关键字。这就是说当关系中没有两个 n 元组在这个域有相同的值时，这个域就是主关键字。

常常要从数据库增加或删除记录。由于这一点，一个域是主关键字的性质是随时间而改变的。所以，一个主关键字应该选择那种无论数据库怎样改变都能继续存在的字段。一个关系当前含有的所有 n 元组称为该关系的外延。数据库更持久的内容，包括它的名字和属性，则被称作数据库的内涵。选择主关键字的时候，应当选择那种能够为本数据库所有可能的外延充当主关键字的。要做到这一点，就必须认真考察数据库的内涵，以便预知可能在外延中出现的 n 元组集。

例 5 假设将来不再增加 n 元组，对于表 5-1 所示的 n 元关系，哪个域是主关键字？

解 因为在这个表中对应每个学生的姓名只有一个 4 元组，学生姓名的域是主关键字。类似地，在这个表中学号是唯一的，学号的域也是主关键字。但是，所学专业的域不是主关键字，因为包含同样的专业的 4 元组有多个。平均学分绩点的域也不是主关键字，因为有 2 个 4 元组包含了同样的 GPA。

在一个 n 元关系中域的组合作可以唯一地标识 n 元组。当一组域的值确定了一个关系中的 n 元组时，这些域的笛卡儿积就叫做复合关键字。

例 6 对于表 5-1 中的 n 元关系，假设不增加 n 元组，所学专业的域与 GPA 的域的笛卡儿积是复合关键字吗？

解 这个表中没有两个 4 元组有同样的专业和同样的 GPA，因此这个笛卡儿积是一个复合关键字。

因为主关键字和复合关键字用于唯一地标识数据库中的记录，当新的记录被加到这个数据库时关键字要保持有效是非常重要的。因此，应该做检测以保证在这个或这些相应的字段中每个新

记录与表中所有其他的记录不同。例如,使用学号作为学生记录的关键字是有意义的,因为没有两个学生有同样的学号。一个大学不应该使用姓名字段作为关键字,因为有可能两个学生有同样的姓名(如 John Smith)。

5.2.4 n 元关系的运算

n 元关系存在可以用于构成新的 n 元关系的不同的运算。综合使用这些运算能够回答对数据库的满足特定条件的所有元组的查询。

一个 n 元关系最基本的运算是在这个 n 元关系中确定满足特定条件的所有 n 元组。例如,我们可能想在学生记录的数据库找出主修计算机科学专业的所有学生的记录。可能我们想在同一个数据库找出所有平均学分绩点在 3.5 以上的学生。也可能我们想在这个数据库找出所有主修计算机科学专业的平均学分绩点在 3.5 以上的学生。为完成这些任务我们使用了选择运算。

定义 2 设 R 是 n 元关系, C 是 R 中元素可能满足的一个条件。那么选择运算 s_C 将 n 元关系 R 映射到 R 中满足条件 C 的所有 n 元组构成的 n 元关系。

例 7 为了找出表 5-1 所示的 n 元关系 R 中主修计算机科学专业的学生的记录,我们使用运算 s_{C_1} , 其中 C_1 是条件专业 = “计算机科学”。结果是两个 4 元组 (Ackermann, 231455, 计算机科学, 3.88) 和 (Chou, 102147, 计算机科学, 3.49)。类似地,为了在这个数据库找出平均学分绩点在 3.5 以上的学生记录,我们使用运算 s_{C_2} , 其中 C_2 是条件 $GPA > 3.5$ 。结果是两个 4 元组 (Ackermann, 231455, 计算机科学, 3.88) 和 (Rao, 678543, 数学, 3.90)。最后,为找出主修计算机科学专业的 GPA 在 3.5 以上的学生记录,我们使用运算 s_{C_3} , 其中 C_3 是条件 (专业 = “计算机科学” $\wedge$ $GPA > 3.5$)。结果由一个 4 元组 (Ackermann, 231455, 计算机科学, 3.88) 构成。

通过投影删去关系中每条记录相同的域可以构建一个新的 n 元关系。

定义 3 投影 $P_{i_1, i_2, \dots, i_m}$ 将 n 元组 $(a_1, a_2, \dots, a_n)$ 映到 m 元组 $(a_{i_1}, a_{i_2}, \dots, a_{i_m})$, 其中 $m \leq n$ 。

换句话说,投影 $P_{i_1, i_2, \dots, i_m}$ 删除了 n 元组的 $n - m$ 个分量,保留了第 $i_1, i_2, \dots, i_m$ 个分量。

例 8 当投影 $P_{1,3}$ 施用到 4 元组 (2, 3, 0, 4), (Jane Doe, 234111001, 地理学, 3.14) 以及 (a_1, a_2, a_3, a_4) 时结果是什么?

解 $P_{1,3}$ 把这些 4 元组分别映到 (2, 0), (Jane Doe, 地理学) 和 (a_1, a_3) 。

例 9 说明了怎样使用投影来产生新关系。

例 9 当投影 $P_{1,4}$ 施用于表 5-1 中的关系时结果是什么?

解 当施用投影 $P_{1,4}$ 时,表的第二列和第三列被删除,得到了表示学生姓名和平均学分绩点的对。表 5-2 给出了这个投影的结果。

当一个投影施用到一个关系的表中时,有可能使行变少。当关系中的某些 n 元组在投影的 m 个分量中每个分量的值都相同,只在被删除的分量有不同的值时,就会出现这种情况。例如,考虑下面的例子。

例 10 当投影 $P_{1,2}$ 施用到表 5-3 的关系时得到什么表?

解 表 5-4 给出了当投影 $P_{1,2}$ 施用到表 5-3 时得到的关系。注意在施用了这个投影后行数减少。

表 5-2 GPA

学生姓名	GPA
Ackermann	3.88
Adams	3.45
Chou	3.49
Goodfriend	3.45
Rao	3.90
Stevens	2.99

表 5-3 注册

学生	专业	课程
Glauser	生物学	BI 290
Glauser	生物学	MS 475
Glauser	生物学	PY 410
Marcus	数学	MS 511
Marcus	数学	MS 603
Marcus	数学	CS 322
Miller	计算机科学	MS 575
Miller	计算机科学	CS 455

当两个表分享着某些相同的字段时连接运算将这两个表组合成一个表。例如，一个表包含了航空公司、航班号和通道的字段，另一个表包含了航班号、通道和起飞时间的字段。可以将这两个表组合成一个包含航空公司、航班号、通道和起飞时间字段的表。

定义 4 设 R 是 m 元关系且 S 是 n 元关系，其中 $p \leq m$ 和 $p \leq n$ 。连接 $J_p(R, S)$ 是 $m+n-p$ 元关系，它包含了所有的 $(m+n-p)$ 元组 $(a_1, a_2, \dots, a_{m-p}, c_1, c_2, \dots, c_p, b_1, b_2, \dots, b_{n-p})$ ，其中 m 元组 $(a_1, a_2, \dots, a_{m-p}, c_1, c_2, \dots, c_p)$ 属于 R 且 n 元组 $(c_1, c_2, \dots, c_p, b_1, b_2, \dots, b_{n-p})$ 属于 S 。

换句话说，连接运算 J_p 从两个关系产生一个新的关系，它把第一个关系的所有 m 元组和第二个关系的所有 n 元组组合起来，其中 m 元组的后 p 个分量与 n 元组的前 p 个分量相同。

例 11 当用连接运算 J_2 组合表 5-5 和表 5-6 中的关系时所得结果关系是什么？

解 J_2 产生的关系在表 5-7 中。

表 5-4 专业

学生	专业
Glauser	生物学
Marcus	数学
Miller	计算机科学

表 5-5 教学课程

教授	系	课号
Cruz	动物学	335
Cruz	动物学	412
Farber	心理学	501
Farber	心理学	617
Grammer	物理学	544
Grammer	物理学	551
Rosen	计算机科学	518
Rosen	数学	575

表 5-6 教室安排

系	课号	教室	时间
计算机科学	518	N521	2:00P. M.
数学	575	N502	3:00P. M.
数学	611	N521	4:00P. M.
物理学	544	B505	4:00P. M.
心理学	501	A100	3:00P. M.
心理学	617	A110	11:00A. M.
动物学	335	A100	9:00A. M.
动物学	412	A100	8:00A. M.

表 5-7 教学安排

教授	系	课号	教室	时间
Cruz	动物学	335	A100	9:00A. M.
Cruz	动物学	412	A100	8:00A. M.
Farber	心理学	501	A100	3:00P. M.
Farber	心理学	617	A110	11:00A. M.
Grammer	物理学	544	B505	4:00P. M.
Rosen	计算机科学	518	N521	2:00P. M.
Rosen	数学	575	N502	3:00P. M.

从已知关系产生新关系的运算除了投影和连接运算以外还有其他运算。对这些运算的描述可以在关于数据库理论的书中找到。

5.2.5 SQL



数据库查询语言 SQL(结构化查询语言 Structured Query Language 的缩写)可以用来执行本节所描述的运算。例 12 说明了 SQL 命令是怎样与 n 元关系上的运算相关的。

例 12 我们将说明怎样用 SQL 来表达查询，这是通过显示怎样使用 SQL 从表 5-8 中做一次关于航班的查询来说明的。SQL 语句

```
SELECT Departure_time      (选择起飞时间)
FROM Flights                (从航班)
WHERE destination = 'Detroit' (当目的地 = '底特律')
```

是用于在航班数据库找出投影 P_5 ，它能够(在起飞时间属性上)选出 5 元组使得这些 5 元组满足条件: $\text{destination} = \text{'Detroit'}$ 。输出是一张以底特律为目的地的包含航班时间的表, 即 08:10, 08:47 和 09:44。SQL 使用 FROM 子句识别查询施用到的 n 元关系, WHERE 子句说明选择运算的条件, 而 SELECT 子句说明将被使用的投影运算。(小心: SQL 使用 SELECT 表示一个投影, 而不是一个选择运算。这是一个令人遗憾的术语不一致的例子。)

表 5-8 航班

航空公司	航班号	通道	目的地	起飞时间
Nadir	122	34	底特律	08:10
Acme	221	22	丹佛	08:17
Acme	122	33	安克雷奇	08:22
Acme	323	34	檀香山	08:30
Nadir	199	13	底特律	08:47
Acme	222	22	丹佛	09:10
Nadir	322	34	底特律	09:44

例 13 说明 SQL 怎样做涉及多个表的查询。

例 13 SQL 语句

```
SELECT professor, time           (选择教授, 时间)
FROM teaching_assignments, class_schedule (从教学课程, 教室安排)
WHERE department = 'mathematics' (当系 = '数学')
```

用于找出在数据库(如表 5-7 所示)中的 5 元组的投影 $P_{1,5}$, 这个数据库是分别在表 5-5 中的教学课程和表 5-6 中的教室安排, 数据库满足 $\text{department} = \text{'mathematics'}$ 条件下的连接 J_2 。输出由单一的 2 元组 (Rosen, 3:00PM) 构成。这里的 SQL FROM 子句用于找出两个不同数据库的连接。

本节我们仅仅接触到关系数据库的基本概念。更多的信息可以在 [AhUl95] 中找到。

练习

- 列出关系 $\{(a, b, c) \mid a, b \text{ 和 } c \text{ 是整数且 } 0 < a < b < c < 5\}$ 中的三元组。
- 列出表 5-8 关系中的 5 元组。
- 假设不增加新的 n 元组, 对于表 5-8 中的数据库找出一个由两个字段构成的复合关键字, 其中一个字段是航空公司。
 - 学号可能是主关键字吗?
 - 姓名可能是主关键字吗?
 - 电话号码可能是主关键字吗?
- 3 元关系中的 3 元组表示了一个学生数据库中的下述属性: 学号、姓名、电话号码。
 - 对这个关系确定一个主关键字。
 - 在什么条件下(姓名、住址)是复合关键字?
 - 在什么条件下(姓名、住址、城市)是复合关键字?
- 5 元关系中的 5 元组表示了美国所有人的下述属性: 姓名、社会保险号、住址、城市、州。
 - 对这个关系确定一个主关键字。
 - 在什么条件下(姓名、住址)是复合关键字?
 - 在什么条件下(姓名、住址、城市)是复合关键字?
- 设 C 是条件: 目的地=底特律, 当你施用选择运算 s_C 到表 5-8 的数据库时, 可以得到什么?
- 设 C 是条件: $(\text{航空公司} = \text{Nadir}) \vee (\text{目的地} = \text{丹佛})$, 当你施用选择运算 s_C 到表 8-8 的数据库时, 可以得到什么?
- 哪个投影映射用于删除一个 6 元组的第一、第二和第四个分量?
- 给出施用投影 $P_{1,4}$ 到表 5-8 以后得到的表。
- 构造把连接运算 J_2 施用到表 5-9 和表 5-10 的关系中所得到的表。

表 5-9 零件需求

供货商	零件号	项目
23	1092	1
23	1101	3
23	9048	4
31	4975	3
31	3477	2
32	6984	4
32	9191	2
33	1001	1

表 5-10 零件库存

零件号	项目	数量	颜色代码
1001	1	14	8
1092	1	2	2
1101	3	1	1
3477	2	25	2
4975	3	6	2
6984	4	10	1
9048	4	12	2
9191	2	80	4

11. 证明：如果 C_1 和 C_2 是 n 元关系 R 的元素可能满足的条件，那么 $s_{C_1}(s_{C_2}(R)) = s_{C_2}(s_{C_1}(R))$ 。
12. 证明：如果 C 是 n 元关系 R 和 S 的元素可能满足的条件，那么 $s_C(R \cap S) = s_C(R) \cap s_C(S)$ 。
13. 证明：如果 R 和 S 是 n 元关系，那么 $P_{i_1, i_2, \dots, i_m}(R \cup S) = P_{i_1, i_2, \dots, i_m}(R) \cup P_{i_1, i_2, \dots, i_m}(S)$ 。
14. 给出一个例子证明：如果 R 和 S 是两个 n 元关系，那么 $P_{i_1, i_2, \dots, i_m}(R - S)$ 可能与 $P_{i_1, i_2, \dots, i_m}(R) - P_{i_1, i_2, \dots, i_m}(S)$ 不同。
15. a) 与下述用 SQL 语句表示的查询相对应的运算是什么？

SELECT Supplier, Project

FROM Part_needs, part_inventory

WHERE Quantity ≤ 10

(选择供货商，项目)

(从零件需求，零件库存)

(当数量 ≤ 10)
- b) 给定表 5-9 和表 5-10 的数据库作为输入，这个查询的输出是什么？
16. 试确定例 3 中的关系是否有一个主关键字。

5.3 关系的表示

5.3.1 引言

在本节及本章的剩余部分要研究的所有关系均为二元关系。因此，在这些内容中出现的“关系”一词都表示二元关系。有多种方式表示有穷集之间的关系。正如已经看到的，一种方式是列出它的有序对；另一种方式是使用表，如 5.1 节所示。本节将讨论另外两种表示关系的方式：一种方式是使用 0-1 矩阵；另一种方式是使用称为有向图的图形表达方式，这种方法将在本节偏后一些进行讨论。

一般说来，矩阵适用于计算机程序中关系的表示。另一方面，人们常常发现使用有向图来表示关系对理解这些关系的性质是很有用的。

5.3.2 用矩阵表示关系

可以用 0-1 矩阵来表示一个有穷集之间的关系。假设 R 是从 $A = \{a_1, a_2, \dots, a_m\}$ 到 $B = \{b_1, b_2, \dots, b_n\}$ 的关系。(这里集合 A 和集合 B 的元素已经按照某一特定的但是任意的次序列出。此外，当 $A=B$ 时我们对于 A 和 B 使用同样的排序。)关系 R 可以用矩阵 $M_R = [m_{ij}]$ 来表示，其中

$$m_{ij} = \begin{cases} 1 & \text{如果 } (a_i, b_j) \in R \\ 0 & \text{如果 } (a_i, b_j) \notin R \end{cases}$$

换句话说，当 a_i 和 b_j 有关系时表示 R 的 0-1 矩阵的 (i, j) 项是 1，当 a_i 和 b_j 没关系时在这个位置是 0(这种表示依赖于对 A 和 B 使用的排序)。

下面的例子说明用矩阵来表示关系。

例 1 假设 $A = \{1, 2, 3\}$ ， $B = \{1, 2\}$ 。令 R 是从 A 到 B 的关系，且如果 $a \in A$ ， $b \in B$ ， $a > b$ ，则 R 包含 (a, b) 。如果 $a_1 = 1$ ， $a_2 = 2$ ， $a_3 = 3$ ， $b_1 = 1$ ， $b_2 = 2$ ，表示 R 的矩阵是什么？

解 因为 $R = \{(2, 1), (3, 1), (3, 2)\}$ ，关于 R 的矩阵是

$$\mathbf{M}_R = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

$\mathbf{M}_R$ 中的 1 说明了有序对 $(2, 1)$, $(3, 1)$ 和 $(3, 2)$ 属于 R , 0 说明了没有其他的有序对属于 R 。 ■

例 2 设 $A = \{a_1, a_2, a_3\}$, $B = \{b_1, b_2, b_3, b_4, b_5\}$ 。哪些有序对在下面的矩阵所表示的关系 R 中?

$$\mathbf{M}_R = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

解 因为 R 由使得 $m_{ij} = 1$ 的有序对 (a_i, b_j) 构成, 因而

$$R = \{(a_1, b_2), (a_2, b_1), (a_2, b_3), (a_2, b_4), (a_3, b_1), (a_3, b_3), (a_3, b_5)\}$$

集合上的关系的矩阵是一个方阵, 可以用这个矩阵确定关系是否有某种性质。 A 上的关系是自反的, 如果对于每个 $a \in A$ 有 $(a, a) \in R$ 。于是, R 是自反的, 当且仅当对 $i = 1, 2, \dots, n$, $(a_i, a_i) \in R$ 。也就有, R 是自反的, 当且仅当对 $i = 1, 2, \dots, n$, $m_{ii} = 1$ 。换句话说, 如果如图 5-3 所示, $\mathbf{M}_R$ 的主对角线的所有元素都等于 1, 那么 R 是自反的。

如果 $(a, b) \in R$ 推出 $(b, a) \in R$, 那么关系 R 是对称的。因此集合 $A = \{a_1, a_2, \dots, a_n\}$ 上的关系 R 是对称的, 当且仅当只要 $(a_i, a_j) \in R$ 就有 $(a_j, a_i) \in R$ 。用矩阵 $\mathbf{M}_R$ 的术语来说, R 是对称的当且仅当只要 $m_{ij} = 1$ 就有 $m_{ji} = 1$ 。这也意味着只要 $m_{ij} = 0$ 就有 $m_{ji} = 0$ 。因此 R 是对称的, 当且仅当对所有的整数对 i, j (其中 $i = 1, 2, \dots, n, j = 1, 2, \dots, n$) 都有 $m_{ij} = m_{ji}$ 。我们看到 R 是对称的当且仅当

$$\mathbf{M}_R = (\mathbf{M}_R)'$$

即 $\mathbf{M}_R$ 是对称矩阵。对称关系的矩阵形式如图 5-4a 所示。

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$

图 5-3 关于自反关系的 0-1 矩阵

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

a) 对称的

b) 反对称的

图 5-4 关于对称和反对称关系的 0-1 矩阵

关系 R 是反对称的, 当且仅当 $(a, b) \in R$ 和 $(b, a) \in R$ 推出 $a = b$ 。因此, 反对称关系的矩阵有下述性质, 即如果 $m_{ij} = 1, i \neq j$, 则 $m_{ji} = 0$ 。或者换句话说, 当 $i \neq j$ 时, $m_{ij} = 0$ 或 $m_{ji} = 0$ 。反对称关系的矩阵形式显示在图 5-4b 中。

例 3 假设集合上的关系 R 由矩阵

$$\mathbf{M}_R = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

表示, R 是自反的、对称的和反对称的吗?

解 因为这个矩阵所有的对角线元素都等于 1, R 是自反的。又由于 $\mathbf{M}_R$ 是对称的, 因此 R 是对称的。也容易看出 R 不是反对称的。 ■

可以用布尔运算加和乘找到两个关系的并和交的矩阵表示。假设集合 A 上的关系 R_1 和 R_2 分别由矩阵 $\mathbf{M}_{R_1}$ 和 $\mathbf{M}_{R_2}$ 来表示。 $\mathbf{M}_{R_1}$ 或 $\mathbf{M}_{R_2}$ 在某个位置为 1, 则关系的并的矩阵相应位置有 1。 $\mathbf{M}_{R_1}$ 和 $\mathbf{M}_{R_2}$ 在某个位置同时为 1, 则关系的交的矩阵相应位置有 1。于是, 这些关系的并和交的矩阵表示是

$$\mathbf{M}_{R_1 \cup R_2} = \mathbf{M}_{R_1} \vee \mathbf{M}_{R_2}, \quad \mathbf{M}_{R_1 \cap R_2} = \mathbf{M}_{R_1} \wedge \mathbf{M}_{R_2}$$

例 4 假设集合 A 上的关系 R_1 和 R_2 由下述矩阵表示, $R_1 \cup R_2$ 和 $R_1 \cap R_2$ 的矩阵表示是什么?

$$\mathbf{M}_{R_1} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{M}_{R_2} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

解 这两个关系的矩阵是

$$\begin{aligned} \mathbf{M}_{R_1 \cup R_2} &= \mathbf{M}_{R_1} \vee \mathbf{M}_{R_2} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \\ \mathbf{M}_{R_1 \cap R_2} &= \mathbf{M}_{R_1} \wedge \mathbf{M}_{R_2} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \blacksquare \end{aligned}$$

现在我们来考虑怎样确定关系合成的矩阵。这个矩阵可以通过关系矩阵的布尔积得到。特别地, 假设 R 是从 A 到 B 的关系且 S 是从 B 到 C 的关系。又假设 A 、 B 和 C 分别有 m 、 n 和 p 个元素。令关于 $S \circ R$, R 和 S 的 0-1 矩阵分别为 $\mathbf{M}_{S \circ R} = [t_{ij}]$, $\mathbf{M}_R = [r_{ij}]$, $\mathbf{M}_S = [s_{ij}]$ (这些矩阵的大小分别为 $m \times p$, $m \times n$ 和 $n \times p$) 有序对 (a_i, c_j) 属于 $S \circ R$ 当且仅当存在元素 b_k 使得 (a_i, b_k) 属于 R 并且 (b_k, c_j) 属于 S 。从而 $t_{ij} = 1$, 当且仅当对某个 k 有 $r_{ik} = s_{kj} = 1$ 。由布尔积的定义, 这意味着

$$\mathbf{M}_{S \circ R} = \mathbf{M}_R \odot \mathbf{M}_S$$

例 5 找出关系 $S \circ R$ 的矩阵, 其中表示 R 和 S 的矩阵是

$$\mathbf{M}_R = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{M}_S = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

解 关于 $S \circ R$ 的矩阵是

$$\mathbf{M}_{S \circ R} = \mathbf{M}_R \odot \mathbf{M}_S = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \blacksquare$$

表示两个关系合成的矩阵可以用来求 $\mathbf{M}_{R^n}$ 的矩阵, 特别地, 由布尔幂的定义有

$$\mathbf{M}_{R^n} = \mathbf{M}_R^{[n]}$$

节末的练习 16 要求证明这个公式。

例 6 求关系 R^2 的矩阵表示, 其中 R 的矩阵表示是

$$\mathbf{M}_R = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

解 R^2 的矩阵是

$$\mathbf{M}_{R^2} = \mathbf{M}_R^{[2]} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \blacksquare$$

5.3.3 用图表示关系

我们已经显示了一个关系可以通过列出它所有的有序对或使用 0-1 矩阵来表示。还有一种重要的表示关系的方法就是图。把集合的每个元素表示成一个点, 每个有序对表示成一条弧, 弧上的箭头标明了弧的方向。当我们把一个有穷集上的关系看做一个有向图时, 就可以使用这种图形表示。

定义 1 一个有向图由顶点(或结点)集 V 和边(或弧)集 E 组成, 其中边集是 V 中元素的有序对的集合。顶点 a 叫做边 (a, b) 的始点, 而顶点 b 叫做这条边的终点。

形如 (a, a) 的边用一条从顶点 a 到自身的弧表示。这种边叫做环。

例7 具有顶点 a, b, c 和 d , 边 $(a, b), (a, d), (b, b), (b, d), (c, a), (c, b)$ 和 (d, b) 的有向图给在图 5-5 中。

集合 A 上的关系 R 表示成一个有向图, 这个图以 A 的元素作为顶点, 以有序对 (a, b) 作为边, 其中 $(a, b) \in R$ 。这就在集合 A 上的关系和以 A 作为顶点集的有向图之间构成了一一对应。于是, 每一个关于关系的论述对应着一个关于有向图的论述, 反之亦然。有向图给出了一个关于关系信息的可见显示。因此, 也常常用图研究关系及其性质。(注意从集合 A 到集合 B 的关系可以用一个有向的图示来表示, 其中对 A 中的每个元素存在一个顶点, B 中的每个元素存在一个顶点, 如 5.1 节所示。然而, 当 $A=B$ 时, 那种图示表示与这里描述的有向图表示不同, 它对关系信息的显示要少得多。)例 8~例 10 说明了怎样用有向图来表示关系。

例8 集合 $\{1, 2, 3, 4\}$ 上的关系

$$R = \{(1, 1), (1, 3), (2, 1), (2, 3), (2, 4), (3, 1), (3, 2), (4, 1)\}$$

的有向图显示在图 5-6 中。

例9 图 5-7 中的有向图所表示的关系 R 中的有序对是什么?

解 关系中的有序对 (x, y) 是

$$R = \{(1, 3), (1, 4), (2, 1), (2, 2), (2, 3), (3, 1), (3, 3), (4, 1), (4, 3)\}$$

每个有序对对应了有向图的一条边, $(2, 2)$ 和 $(3, 3)$ 对应了环。

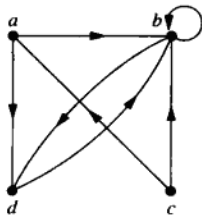


图 5-5 一个有向图

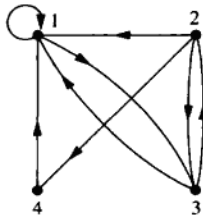


图 5-6 关系 R 的有向图

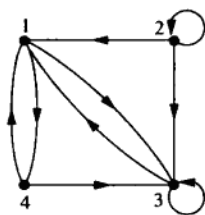


图 5-7 关系 R 的有向图

表示关系的有向图可以用来确定关系是否有各种性质。例如, 一个关系是自反的, 当且仅当有向图的每个顶点都有环, 从而每个形如 (x, x) 的有序对都出现在关系中。一个关系是对称的, 当且仅当对有向图不同顶点之间的每一条边都存在一条方向相反的边, 从而只要 (x, y) 在关系中就有 (y, x) 在关系中。类似地, 一个关系是反对称的, 当且仅当在不同的两个顶点之间不存在两条方向相反的边。最后, 一个关系是传递的, 当且仅当只要存在一条从顶点 x 到顶点 y 的边和一条顶点 y 到顶点 z 的边, 就有一条从顶点 x 到顶点 z 的边(完成一个三角形, 其中每条边都是具有正确方向的有向边)。

注意 对称关系可以用无向图表示, 这个图中的边没有方向。我们将在第 6 章研究无向图。

例10 对于图 5-8 所示的有向图的关系, 确定它们是否为自反的、对称的、反对称的和传递的。

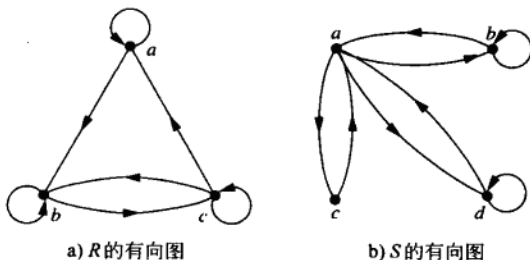


图 5-8 关系 R 和 S 的有向图

解 因为 R 的有向图的每个顶点都有环, 因此它是自反的。 R 既不是对称的也不是反对称的, 因为存在一条从 a 到 b 的边, 但没有从 b 到 a 的边, 并且在连接 b 和 c 的两个方向都有边。最后, R 不是传递的, 因为从 a 到 b 有边, 从 b 到 c 有边, 但是从 a 到 c 没有边。

因为在有向图 S 的所有顶点没出现环, 因此关系不是自反的。因为在不同顶点之间的每条边都伴随一条方向相反的边, 因此它是对称的但不是反对称的。不难看出 S 不是传递的, 因为 (c, a) 和 (a, b) 属于 S 但 (c, b) 不属于 S 。 ■

练习

1. 用矩阵表示下面每个 $\{1, 2, 3\}$ 上的关系(按增序列出这个集合的元素)。

a) $\{(1, 1), (1, 2), (1, 3)\}$

b) $\{(1, 2), (2, 1), (2, 2), (3, 3)\}$

c) $\{(1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3)\}$

d) $\{(1, 3), (3, 1)\}$

2. 列出和下面矩阵对应的 $\{1, 2, 3\}$ 上的关系中的有序对(其中行和列对应于按增序列出的整数)。

a) $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$

b) $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

3. 怎样用表示集合 A 上关系 R 的有向图确定这个关系是否是反自反的?

4. 确定练习 2 中的矩阵所表示的关系是否为自反的、反自反的、对称的、反对称的和传递的。

5. R 是包含了前 100 个正整数的集合 $A = \{1, 2, \dots, 100\}$ 上的关系, 如果 R 满足下述条件, 那么表示 R 的矩阵有多少个非 0 的元素?

a) $\{(a, b) \mid a > b\}$

b) $\{(a, b) \mid a \neq b\}$

c) $\{(a, b) \mid a = b + 1\}$

d) $\{(a, b) \mid a = 1\}$

e) $\{(a, b) \mid ab = 1\}$

6. 当 R 是有穷集 A 上的关系时, 怎样从表示 R 的关系矩阵找到这个关系的补 $\bar{R}$ 的矩阵?

7. 设 R 是矩阵

$$M_R = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

所表示的关系, 求表示下述关系的矩阵。

a) R^{-1}

b) $\bar{R}$

c) R^2

8. 设 R 是矩阵

$$M_R = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$
 表示的关系, 求表示下述关系的矩阵

a) R^2

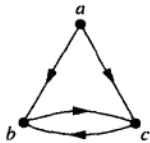
b) R^3

c) R^4

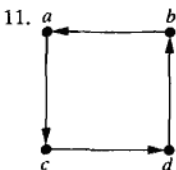
9. 设 R 是 n 元素集合上的关系。如果在表示 R 的矩阵 M_R 中存在 k 个非 0 的元素, 那么在表示 R 的补 $\bar{R}$ 的矩阵 $M_{\bar{R}}$ 中存在多少个非 0 元素?

在练习 10~12 列出由下述有向图所表示的关系中的有序对。

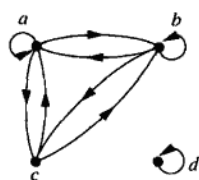
10.



11.



12.



13. 怎样用有穷集 A 上关系 R 的有向图确定关系是否是非对称的?

14. 确定练习 10~11 所示的有向图表示的关系是否为自反的、反自反的、对称的、反对称的和传递的。

15. 设 R 是集合 A 上的关系, 解释怎样用表示 R 的有向图得到表示关系的逆 R^{-1} 的有向图?

16. 证明: 如果 M_R 是表示关系 R 的矩阵, 那么 $M_R^{[n]}$ 是表示关系 R^n 的矩阵。

5.4 关系的闭包

5.4.1 引言

一个计算机网络在波士顿、芝加哥、丹佛、底特律、纽约和圣迭戈设有数据中心。从波士顿到芝加哥，波士顿到底特律，芝加哥到底特律，底特律到丹佛和纽约到圣迭戈都有单向的电话线。如果存在一条从数据中心 a 到 b 的电话线， (a, b) 就属于关系 R 。我们怎样确定从一个中心是否有一条电话线或多条电话线(可能不直接)链接到另一个中心？由于所有的链接不一定是直接的，例如从波士顿可通过底特律到丹佛，因此不能直接使用 R 来回答这个问题。用关系的语言说， R 不是传递的，因而它不包含可能被链接的所有的对。正如我们将在本节证明的，我们可以通过构造包含 R 的最小的传递关系来找出每一对有着链接的数据中心。这个关系叫做 R 的传递闭包。

一般说来，设 R 是集合 A 上的关系。 R 可能具有或者不具有某些性质 P ，例如自反性、对称性或传递性。如果存在包含 R 的具有性质 P 的关系 S ，并且 S 是包含 R 且具有性质 P 的每一个关系的子集，那么 S 叫做 R 的关于 P 的闭包。(注意一个关系关于一个性质的闭包可能不存在，见节末的练习 8 和 17。)我们将说明怎样找关系的自反闭包、对称闭包和传递闭包。

5.4.2 闭包

集合 $A = \{1, 2, 3\}$ 上的关系 $R = \{(1, 1), (1, 2), (2, 1), (3, 2)\}$ 不是自反的。我们怎样才能产生一个包含 R 的尽可能小的自反关系呢？这可以通过把 $(2, 2)$ 和 $(3, 3)$ 加到 R 中来做到，因为只有它们是不在 R 中的形如 (a, a) 的对。很明显，这个新关系包含了 R 。此外，任何包含 R 的自反关系必须包含 $(2, 2)$ 和 $(3, 3)$ 。因为这个关系包含了 R ，是自反的，并且被包含在每一个包含 R 的自反关系之中，因此它就是 R 的自反闭包。

正如这个例子所显示的，给定集合 A 上的关系 R ，对于 $a \in A$ ，可以通过把形如 (a, a) 的所有对，除了已在 R 中的之外，都加到 R 中，就构成了 R 的自反闭包。加入这些对产生了一个新的自反的、包含 R 的关系，并且它被包含在任何包含 R 的自反关系之中。我们看到 R 的自反闭包等于 $R \cup \Delta$ ，其中 $\Delta = \{(a, a) \mid a \in A\}$ 是 A 上的对角关系。(读者应能验证这个结果。)

例 1 整数集上的关系 $R = \{(a, b) \mid a < b\}$ 的自反闭包是什么？

解 R 的自反闭包是

$$R \cup \Delta = \{(a, b) \mid a < b\} \cup \{(a, a) \mid a \in \mathbb{Z}\} = \{(a, b) \mid a \leq b\}$$

$\{1, 2, 3\}$ 上的关系 $\{(1, 1), (1, 2), (2, 2), (2, 3), (3, 1), (3, 2)\}$ 不是对称的。我们怎样产生一个包含 R 的尽可能小的对称关系？只需增加 $(2, 1)$ 和 $(1, 3)$ ，因为只有它们是具有 $(a, b) \in R$ 而 (b, a) 不在 R 中的那种 (b, a) 对。这个新关系是对称的，且包含了 R 。此外，任何包含了 R 的对称关系一定包含这个新关系，因为一个包含 R 的对称关系必须包含 $(2, 1)$ 和 $(1, 3)$ 。因此这个新关系叫做 R 的对称闭包。

正如这个例子所显示的，关系 R 的对称闭包可以通过增加所有形如 (b, a) 的有序对构成，其中 (a, b) 在关系中而 (b, a) 不在关系中。增加这些有序对产生一个关系，它是对称的，包含了 R ，并且它被包含在任何包含 R 的对称关系之中。关系 R 的对称闭包可以通过取关系与它的逆的并来构造，即 $R \cup R^{-1}$ 是 R 的对称闭包，其中 $R^{-1} = \{(b, a) \mid (a, b) \in R\}$ 。读者应能验证这个结果。

例 2 正整数集上的关系 $R = \{(a, b) \mid a > b\}$ 的对称闭包是什么？

解 R 的对称闭包是关系

$$R \cup R^{-1} = \{(a, b) \mid a > b\} \cup \{(b, a) \mid a > b\} = \{(a, b) \mid a \neq b\}$$

Example 这最后一步相等是由于 R 包含了所有正整数构成的有序对，其中第一元素大于第二元素，并且 R^{-1} 包含了所有正整数构成的有序对，其中第一元素小于第二元素。

假设关系 R 不是传递的，我们怎样产生一个包含 R 的传递关系并使得这个新的关系被包含在

任何包含 R 的传递关系之中? 对于已经在 R 中的 (a, b) 和 (b, c) , 可以通过增加所有形如 (a, c) 的有序对构成 R 的传递闭包吗? 考虑集合 $\{1, 2, 3, 4\}$ 上的关系 $R = \{(1, 3), (1, 4), (2, 1), (3, 2)\}$ 。这个关系不是传递的, 因为对于在 R 中的 (a, b) 和 (b, c) , 它不包含所有形如 (a, c) 的有序对。这种不在 R 中的有序对是 $(1, 2)$, $(2, 3)$, $(2, 4)$ 和 $(3, 1)$ 。把这些有序对加到 R 中并不能产生一个传递关系, 因为所得的结果关系包含 $(3, 1)$ 和 $(1, 4)$, 但不包含 $(3, 4)$ 。这说明构造关系的传递闭包比起构造它们的自反闭包或对称闭包更复杂。下面我们将建立一个构造传递闭包的算法。正如将要显示的, 一个关系的传递闭包可以通过增加那些必须出现的有序对来得到, 要重复这个过程, 直到没有必须增加的有序对为止。

5.4.3 有向图的路径

我们将看到用有向图表示关系有助于构造关系的传递闭包。为此现在引入某些将要用到的术语。

通过沿有向图的边(按照这条边的箭头指示的相同方向)移经有向图就得到一条有向图中的路径。

定义 1 在有向图 G 中从 a 到 b 的一条路径是 G 中一条或多条边的序列 $(x_0, x_1), (x_1, x_2), (x_2, x_3), \dots, (x_{n-1}, x_n)$, 其中 $x_0 = a, x_n = b$ 。即一个边的序列, 其中一条边的终点和路径中下一条边的始点相同。这条路径记为 $x_0, x_1, \dots, x_{n-1}, x_n$, 长度为 n 。在同一顶点开始和结束的路径叫做回路或圈。

有向图的一条路径可以多次通过一个顶点。此外, 有向图的一条边也可以多次出现在一条路径中。

例 3 下面哪些路径是图 5-9 的有向图中的路径: a, b, e, d ; a, e, c, d, b ; b, a, c, b, a, a, b ; d, c ; c, b, a ; e, b, a, b, a, b, e ? 这些路径的长度是多少? 这个列表中的哪些路径是回路?

解 因为 (a, b) 、 (b, e) 和 (e, d) 都是边, a, b, e, d 是长为 3 的路径。因为 (c, d) 不是边, a, e, c, d, b 不是路径。因为 (b, a) , (a, c) , (c, b) , (b, a) , (a, a) 和 (a, b) 都是边, b, a, c, b, a, a, b 是长为 6 的路径。我们也看到, 因为 (d, c) 是边, d, c 是长为 1 的路径。还有由于 (c, b) 和 (b, a) 是边, c, b, a 是长为 2 的路径。 (e, b) , (b, a) , (a, b) , (b, a) , (a, b) , (b, e) 都是边, 因此 e, b, a, b, a, b, e 是长为 6 的路径。

两条路径 b, a, c, b, a, a, b 和 e, b, a, b, a, b, e 是回路, 因为它们在同一顶点开始和结束。路径 a, b, e, d ; c, b, a 和 d, c 不是回路。

术语路径也用于关系。把有向图的定义推广到关系可知, 如果存在一个元素的序列 $a, x_1, x_2, \dots, x_{n-1}, b$ 具有 $(a, x_1) \in R, (x_1, x_2) \in R, \dots, (x_{n-1}, b) \in R$, 那么在 R 中存在一条从 a 到 b 的路径。从关系中的路径定义可以得到定理 1。

定理 1 设 R 是集合 A 上的关系。从 a 到 b 存在一条长为 n (n 为正整数) 的路径, 当且仅当 $(a, b) \in R^n$ 。

证 我们将使用数学归纳法。根据定义, 从 a 到 b 存在一条长为 1 的路径, 当且仅当 $(a, b) \in R$ 。因此当 $n=1$ 时定理为真。

假定对于正整数 n 定理为真, 这是归纳假设。从 a 到 b 存在一条长为 $n+1$ 的路径, 当且仅当存在元素 $c \in A$ 使得从 a 到 c 存在一条长为 1 的路径, 即 $(a, c) \in R$, 以及一条从 c 到 b 的长为 n 的路径, 即 $(c, b) \in R^n$ 。因此, 由归纳假设, 从 a 到 b 存在一条长为 $n+1$ 的路径, 当且仅当存在一个元素 c , 使得 $(a, c) \in R$ 和 $(c, b) \in R^n$ 。但是存在这样一个元素, 当且仅当 $(a, b) \in R^{n+1}$ 。因此, 从 a 到 b 存在一条长为 $n+1$ 的路径, 当且仅当 $(a, b) \in R^{n+1}$ 。定理得证。 $\square$

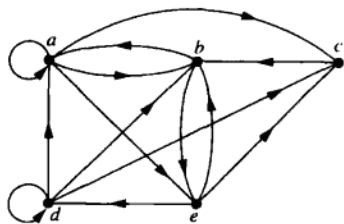


图 5-9 一个有向图

5.4.4 传递闭包

我们现在证明找一个关系的传递闭包与在相关的有向图中确定哪些顶点对被路径连接是等价的。由此要定义一个新的关系。

定义 2 设 R 是集合 A 上的关系。连通性关系 R^* 由对 (a, b) 构成, 使得在 R 中从顶点 a 到 b 之间存在一条至少长为 1 的路径。

因为 R^n 由对 (a, b) 构成, 使得存在一条从 a 到 b 的长为 n 的路径, 从而 R^* 是所有集合 R^n 的并。换句话说,

$$R^* = \bigcup_{n=1}^{\infty} R^n$$

许多模型都用到连通性关系。

例 4 设 R 是世界上所有人的集合上的关系, 如果 a 认识 b , 那么 R 包含 (a, b) 。 R^n 是什么? 其中 n 是大于 2 的正整数。 R^* 是什么?

解 如果存在 c 使得 $(a, c) \in R$ 且 $(c, b) \in R$, 即存在 c 使得 a 认识 c , c 认识 b , 那么关系 R^2 包含 (a, b) 。类似地, 如存在 $x_1, x_2, \dots, x_{n-1}$ 使得 a 认识 x_1 , x_1 认识 x_2 , $\dots$, x_{n-1} 认识 b , 那么 R^n 包含对 (a, b) 。

如果存在从 a 开始至 b 终止的序列, 使得序列中的每个人都认识序列中的下一个人, 那么 R^* 包含对 (a, b) 。(关于 R^* 存在许多有趣的猜想。你认为这个连通关系包含以你作为第一元素, 蒙古的总统作为第二元素的一对元素吗? 在第 6 章我们将用图建立这个应用的模型。)

例 5 设 R 是纽约市所有地铁站集合的关系。如果可以从 a 站不换乘就旅行到 b 站, 那么 R 包含对 (a, b) 。当 n 是正整数时, R^n 是什么? R^* 是什么?

解 如果换 $n-1$ 次车就可以从 a 站旅行到 b 站, 关系 R^n 就包含 (a, b) 。从 a 站旅行到 b 站, 如果需要可以换乘任意多次, 关系 R^* 就由这种有序对 (a, b) 组成。(读者应该能验证这些论断。)

例 6 设 R 是美国所有的州的集合上的关系, 如果 a 州和 b 州有公共的边界, 那么 R 包含 (a, b) 。 R^n 是什么? 其中 n 是正整数。 R^* 又是什么?

解 关系 R^n 由对 (a, b) 构成, 可以从 a 州恰好跨越 n 次州界到达 b 州。 R^* 由对 (a, b) 构成, 可以从 a 州跨越任意多次边界到达 b 州(读者应该能够验证这些论断。)那些包含与美国大陆不相连的州(即含有阿拉斯加或夏威夷)的有序对是不在 R^* 中的。

定理 2 证明一个关系的传递闭包和相关的连通性关系是等同的。

定理 2 关系 R 的传递闭包等于连通性关系 R^* 。

证 注意 R^* 包含 R 。为证明 R^* 是 R 的传递闭包, 我们必须证明 R^* 是传递的且对一切包含 R 的传递关系 S , 有 $R^* \subseteq S$ 。

首先, 我们证明 R^* 是传递的。如果 $(a, b) \in R^*$ 和 $(b, c) \in R^*$, 那么在 R 中存在从 a 到 b 和从 b 到 c 的路径。我们从 a 到 b 的路径开始, 并且沿着从 b 到 c 的路径就得到一条从 a 到 c 的路径。因此, $(a, c) \in R^*$ 。这就得出 R^* 是传递的。

现在假设 S 是包含 R 的传递关系, 因为 S 是传递的, S^n 也是传递的(读者应该能验证这一点), 并且 $S^n \subseteq S$ (由 5.1 节定理 1)。此外, 因为

$$S^* = \bigcup_{k=1}^{\infty} S^k$$

和 $S^k \subseteq S$, 因此 $S^* \subseteq S$ 。现在注意到如果 $R \subseteq S$, 那么 $R^* \subseteq S^*$, 这是由于任何 R 中的路径也是 S 中的路径。从而 $R^* \subseteq S^* \subseteq S$ 。于是, 任何包含 R 的传递关系也一定包含 R^* 。因此, R^* 是 R 的传递闭包。

既然我们知道传递闭包等于连通性关系, 我们考虑这个关系的计算问题。在一个有限的有向图中不需要检测任意长的路径来确定是否在两个顶点之间存在一条路径。正如下面的引理所证明的。检测包含不超过 n 条边的路径就足够了, 这里 n 是集合中的元素个数。

引理 1 设 A 是 n 元素集合, R 是 A 上的关系。如果 R 中存在一条从 a 到 b 的长至少为 1 的

路径, 那么存在一条长度不超过 n 的这种路径。此外, 当 $a \neq b$ 时, 如果在 R 中存在一条从 a 到 b 的路径, 那么存在一条长度不超过 $n-1$ 的这种路径。

证 假设 R 中存在一条从 a 到 b 的路径。令 m 是这种路径的最短长度。假设 $x_0, x_1, x_2, \dots, x_{m-1}, x_m$ 是一条这样的路径, 其中 $x_0 = a, x_m = b$ 。

假设 $a = b$ 和 $m > n$, 使得 $m \geq n+1$ 。由鸽巢原理, 因为 A 中有 n 个顶点, 在 m 个顶点 $x_0, x_1, x_2, \dots, x_{m-1}$ 中至少有两个是相同的(见图 5-10)。

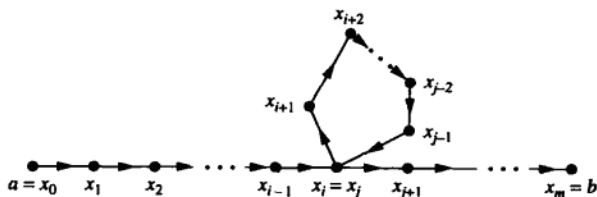


图 5-10 产生一条长度不超过 n 的路径

假设 $x_i = x_j$ 满足 $0 \leq i < j \leq m-1$ 。那么这条路径包含一条从 x_i 到 x_i 自身的回路。可以把这条回路从由 a 到 b 的路径中删除, 剩下的路径, 即 $x_0, x_1, \dots, x_i, x_{j+1}, \dots, x_{m-1}, x_m$ 是从 a 到 b 的更短的路径。因此, 最短长度的这种路径的长度一定小于等于 n 。

$a \neq b$ 的情况留给读者作为练习。 □

由引理 1, 我们看出 R 的传递闭包是 $R, R^2, R^3, \dots, R^n$ 的并。这是由于在 R^* 的两个顶点之间存在一条路径, 当且仅当对某个正整数 $i (i \leq n)$ 在 R^i 的这些顶点之间存在一条路径。因为

$$R^* = R \cup R^2 \cup R^3 \cup \dots \cup R^n$$

并且表示关系的并的 0-1 矩阵是这些关系的 0-1 矩阵的联合, 因此传递闭包的 0-1 矩阵是 R 的 0-1 矩阵的前 n 次幂的 0-1 矩阵的联合。

定理 3 设 M_R 是 n 元素集合上的关系 R 的 0-1 矩阵。那么传递闭包 R^* 的 0-1 矩阵是

$$M_{R^*} = M_R \vee M_R^{[2]} \vee M_R^{[3]} \vee \dots \vee M_R^{[n]}$$

例 7 求关系 R 的传递闭包的 0-1 矩阵, 其中

$$M_R = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

解 由定理 3, R^* 的 0-1 矩阵是

$$M_{R^*} = M_R \vee M_R^{[2]} \vee M_R^{[3]}$$

因为

$$M_R^{[2]} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad M_R^{[3]} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

所以

$$M_{R^*} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \vee \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \vee \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

定理 3 可以作为计算关系 R^* 的矩阵的算法基础。为求出这个矩阵, 要连续计算 M_R 的布尔幂, 直到第 n 次幂为止。当计算每次幂时就构成这个幂与所有较小的幂的联合。当做到第 n 次幂时, 就得到关于 R^* 的矩阵。这个过程见算法 1。

算法 1 计算传递闭包的过程

procedure *transitive closure* ($M_R: n \times n$ 的 0-1 矩阵)

$A := M_R$

$B := A$

for $i := 2$ to n

begin

$A := A \odot M_R$

$B := B \vee A$

end{B 是关于 R' 的 0-1 矩阵}

我们可以容易地求出用算法 1 确定关系的传递闭包所使用的位运算次数。计算布尔幂 $M_R, M_R^{[2]}, \dots, M_R^{[n]}$ 需要找到 $n-1$ 个 $n \times n$ 的 0-1 矩阵的布尔积。每个布尔积可以使用 $n^2(2n-1)$ 次位运算求得。因此, 计算这些乘积使用 $n^2(2n-1)(n-1)$ 次位运算。

为从 n 个 M_R 的布尔幂求 $M_{R'}$, 需要求 $n-1$ 个 0-1 矩阵的联合。计算每一个联合使用 n^2 次位运算。因此, 在这部分计算中使用 $(n-1)n^2$ 次位运算。所以, 当使用算法 1 时, 用 $n^2(2n-1)(n-1) + (n-1)n^2 = 2n^3(n-1) = O(n^4)$ 次位运算就可以求出 n 元素集合上关系的传递闭包的矩阵。下面我们将要描述一个更有效的求传递闭包的算法。

5.4.5 沃舍尔算法

沃舍尔算法得名于史蒂芬·沃舍尔, 他在 1960 年给出该算法。这个算法能够有效计算关系的传递闭包。算法 1 求出 n 元素集合上关系的传递闭包使用 $2n^3(n-1)$ 次位运算。而沃舍尔算法只使用 $2n^3$ 次位运算就可以求出这个传递闭包。

注意 沃舍尔算法有时也叫做罗伊-沃舍尔算法, 因为伯纳德·罗伊(B·Roy)在 1959 年描述了这个算法。

假设 R 是 n 元素集合上的关系, 设 $v_1, v_2, \dots, v_n$ 是这 n 个元素的任意排列。沃舍尔算法中用到一条路径的内点的概念。如果 $a, x_1, x_2, \dots, x_{m-1}, b$ 是一条路径, 它的内点是 $x_1, x_2, \dots, x_{m-1}$, 即除了第一和最后一个顶点之外出现在路径中的所有顶点。例如, 在有向图的一条路径 a, c, d, f, g, h, b, j 的内点是 c, d, f, g, h, b 。 a, c, d, a, f, b 的内点是 c, d, a, f 。(注意这条路径的起点不是内点, 除非这条路径再次访问它, 且不是作为终点来访问的。类似地, 这条路径的终点也不是内点, 除非它在这之前曾被这条路径访问过, 且不是作为起点来访问的。)

沃舍尔算法的基础是构造一系列 0-1 矩阵。这些矩阵是 $W_0, W_1, \dots, W_n$, 其中 $W_0 = M_R$ 是这个关系的 0-1 矩阵, 且 $W_k = [w_{ij}^{(k)}]$ 。如果存在一条从 v_i 到 v_j 的路径使得这条路径的所有内点都在集合 $\{v_1, v_2, \dots, v_k\}$ (表中的前 k 个顶点) 之中, 那么 $w_{ij}^{(k)} = 1$, 否则为 0 (这条路径的起点和终点可能在表中的前 k 个顶点的集合之外)。注意 $W_n = M_{R'}$, 因为 $M_{R'}$ 的第 (i, j) 项是 1, 当且仅当存在一条从 v_i 到 v_j 的路径, 且全部内点都在集合 $\{v_1, v_2, \dots, v_n\}$ 之中 (但这些就是有向图的所有顶点)。例 8 说明矩阵 W_k 表示什么。

例 8 设 R 是一个关系, 它的有向图如图 5-11 所示。设 a, b, c, d 是集合元素的排列。求矩阵 W_0, W_1, W_2, W_3 和 W_4 。矩阵 W_4 是 R 的传递闭包。

解 令 $v_1 = a, v_2 = b, v_3 = c$, 且 $v_4 = d$ 。 W_0 是这个关系的矩阵, 于是

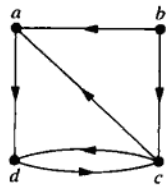


图 5-11 关系 R 的有向图

$$W_0 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

如果存在一条 v_i 到 v_j 的且只有 $v_1=a$ 作为内点的路径, W_1 的 (i, j) 项有 1。注意因为所有长为 1 的路径没有内点, 所以仍旧可以使用这些路径。此外存在一条从 b 到 d 的路径, 即 b, a, d 。因此

$$W_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

如果存在一条 v_i 到 v_j 的且只有 $v_1=a$ 和 $v_2=b$ 作为内点的路径, W_2 的 (i, j) 项有 1。因为没有边以 b 作为终点, 当我们允许 b 作为内点时不会得到新的路径。因此, $W_2=W_1$ 。

若存在一条 v_i 到 v_j 的只用 $v_1=a, v_2=b$ 和 $v_3=c$ 作为内点的路径, 则 W_3 的 (i, j) 项有 1。我们现在有从 d 到 a 的路径, 即 d, c, a 和从 d 到 d 的路径 d, c, d 。因此

$$W_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

最后, 如果存在一条从 v_i 到 v_j 的路径, 并且以 $v_1=a, v_2=b, v_3=c$ 及 $v_4=d$ 作为内点, 那么 W_4 的 (i, j) 项为 1。因为这是图的全部顶点, 此项为 1, 当且仅当存在一条从 v_i 到 v_j 的路径。因此

$$W_4 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

这个最后的矩阵就是传递闭包的矩阵。

沃舍尔算法通过有效地计算 $W_0=M_R, W_1, W_2, \dots, W_n=M_R$ 来计算 M_R 。不难看出可以直接从 W_{k-1} 计算 W_k : 存在一条从 v_i 到 v_j 的只以 $v_1, v_2, \dots, v_k$ 中的顶点作为内点的路径, 当且仅当要么存在一条从 v_i 到 v_j 的且内点是表中前 $k-1$ 个顶点的路径, 要么存在从 v_i 到 v_k 的路径和从 v_k 到 v_j 的路径, 而这些路径的内点仅在表中的前 $k-1$ 个顶点中。这就是说, 要么在 v_k 被允许作为内点之前从 v_i 到 v_j 已经存在一条路径, 要么允许 v_k 作为内点产生一条从 v_i 到 v_k 然后从 v_k 到 v_j 的路径。这两种情况如图 5-12 所示。

第一种类型的路径存在, 当且仅当 $w_{ij}^{[k-1]}=1$, 且第二种类型的路径存在, 当且仅当 $w_{ik}^{[k-1]}=1$ 和 $w_{kj}^{[k-1]}=1$ 。于是, $w_{ij}^{[k]}$ 是 1, 当且仅当或者 $w_{ij}^{[k-1]}=1$, 或者 $w_{ik}^{[k-1]}=1$ 和 $w_{kj}^{[k-1]}=1$ 。这就得到引理 2。

引理 2 设 $W_k=[w_{ij}^{[k]}]$ 是 0-1 矩阵, 它在 (i, j) 位置有 1, 当且仅当存在一条从 v_i 到 v_j 的路径, 其内点取自集合 $\{v_1, v_2, \dots, v_k\}$, 那么

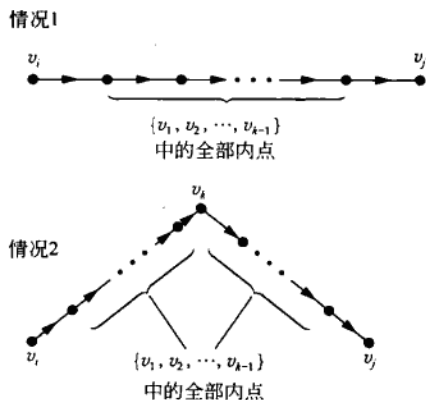


图 5-12 把 v_k 加到允许使用的内点集中

$$w_{ij}^{[k]} = w_{ij}^{[k-1]} \vee (w_{ik}^{[k-1]} \wedge w_{kj}^{[k-1]})$$

其中 i, j 和 k 是不超过 n 的正整数。

引理 2 提供了有效计算矩阵 $W_k (k=1, 2, \dots, n)$ 的手段。我们使用引理 2 把沃舍尔算法的伪码在算法 2 中给出。

算法 2 沃舍尔算法

procedure *Warshall* ($M_R: n \times n$ 的 0-1 矩阵)

$W := M_R$

for $k := 1$ **to** n

begin

for $i := 1$ **to** n

begin

for $j := 1$ **to** n

$w_{ij} := w_{ij} \vee (w_{ik} \wedge w_{kj})$

end

end { $W = [w_{ij}]$ 是 M_R^* }

很容易以位运算次数计算沃舍尔算法的计算复杂度。使用引理 2, 从项 $w_{ij}^{[k-1]}$, $w_{ik}^{[k-1]}$ 和 $w_{kj}^{[k-1]}$ 找到项 $w_{ij}^{[k]}$ 需要 2 次位运算。为从 W_{k-1} 求出 W_k 的所有 n^2 个项需要 $2n^2$ 次位运算。因为沃舍尔算法从 $W_0 = M_R$ 开始, 计算 n 个 0-1 矩阵的序列 $W_1, W_2, \dots, W_n = M_R^*$, 使用的位运算总次数是 $n \cdot 2n^2 = 2n^3$ 。

练习

1. 设 R 是集合 $\{0, 1, 2, 3\}$ 上的关系, R 包含有序对 $(0, 1), (1, 1), (1, 2), (2, 0), (2, 2)$ 和 $(3, 0)$, 求
 - a) R 的自反闭包
 - b) R 的对称闭包
2. 设 R 是整数集上的关系 $\{(a, b) \mid a \text{ 整除 } b\}$, R 的对称闭包是什么?

在练习 3~4 中画出给定有向图所表示关系的自反闭包的有向图。



5. 对于练习 3~4 的有向图表示的关系, 找出关系的对称闭包的有向图。
6. 对于练习 3~4 的每个有向图表示的关系, 求包含它的最小的自反对称关系的有向图。
7. 假设有穷集 A 上的关系 R 由矩阵 M_R 表示, 证明表示 R 的对称闭包的矩阵是 $M_R \vee M_R^t$ 。
8. 什么时候可能定义一个关系 R 的反自反闭包, 即一个包含 R 的关系是反自反的且被包含在每一个包含 R 的反自反关系之中?
9. 设 R 是集合 $\{1, 2, 3, 4, 5\}$ 上的关系, R 包含有序对 $(1, 3), (2, 4), (3, 1), (3, 5), (4, 3), (5, 1), (5, 2)$ 和 $(5, 4)$ 。求
 - a) R^2
 - b) R^3
 - c) R^4
 - d) R^5
 - e) R^6
 - f) R^*
10. 设 R 是所有学生的集合上的关系, 如果 $a \neq b$ 且 a 和 b 至少有一门公共课程, 则 R 包含了有序对 (a, b) 。什么时候 (a, b) 在下面的关系中?
 - a) R^2
 - b) R^3
 - c) R^*
11. 假设关系 R 是对称的, 证明 R^* 是对称的。

12. 使用算法 1 找出下面 $\{1, 2, 3, 4\}$ 上的关系的传递闭包。
- $\{(1, 2), (2, 1), (2, 3), (3, 4), (4, 1)\}$
 - $\{(2, 1), (2, 3), (3, 1), (3, 4), (4, 1), (4, 3)\}$
 - $\{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$
 - $\{(1, 1), (1, 4), (2, 1), (2, 3), (3, 1), (3, 2), (3, 4), (4, 2)\}$
13. 使用沃舍尔算法找出练习 12 中关系的传递闭包。
14. 求出包含关系 $\{(1, 2), (1, 4), (3, 3), (4, 1)\}$ 的最小的关系, 使得它是
- 自反的和传递的。
 - 对称的和传递的。
 - 自反的、对称的和传递的。
15. 已经设计出算法用 $O(n^{2.8})$ 次位运算来计算两个 $n \times n$ 的 0-1 矩阵的布尔积。假设可以使用这些算法, 给出用算法 1 和沃舍尔算法求 n 元素集合上关系的传递闭包所用位运算数的大 O 估计。
16. 修改算法 1 找出 n 元素集合上关系的传递闭包的自反闭包。
17. 证明: 集合 $\{0, 1, 2\}$ 上的关系 $R = \{(0, 0), (0, 1), (1, 1), (2, 2)\}$ 关于下述性质 P 的闭包不存在, 如果 P
- “不是自反的”。
 - “有奇数个元素”。

5.5 等价关系基础

5.5.1 引言

在一些程序设计语言中, 变量的命名可以包含无数字符。当编译器确定要检查两个变量是否相同时, 字符的数量就有限制。例如, 在传统 C 中, 编译器只检查变量名的前八个字符。(这些字符是大写或小写字母、数字或下划线。)所以编译器对于比八个字符长的串就考虑它的前八个字符相同。设 R 是字符串集合上的关系, s 和 t 是两个字符串, 如果 s 和 t 至少有八个字符长, 而 s 和 t 的前八个字符相符, 或者 $s=t$, 则 sRt 。容易得出 R 是自反的、对称的和传递的。而且 R 把所有字符串的集合分成一些类, 传统 C 的编译器认为在某一特定类中的所有字符串是相同的。

如果 4 整除 $a-b$, 整数 a 和 b 有模 4 同余的关系。后面我们将证明这个关系是自反的、对称的和传递的。不难看出 a 和 b 相关, 当且仅当被 4 整除时, a 和 b 有相同的余数。这个关系将整数集划分成不同的 4 类。当我们仅关心一个整数被 4 整除的余数时, 我们只需要知道它在哪个类而不必知道它的特定值。

这两个关系, R 和模 4 同余关系是等价关系, 即是自反的、对称的和传递的。本节将证明这种关系把集合划分成由等价元素构成的不相交的类。当我们仅关心集合的一个元素是否在某个元素类, 而不介意它的特殊身份时, 就出现了等价关系。

5.5.2 等价关系

在这一节我们将研究具有一组特殊性质的关系, 可以用这组性质为在某一方面类似的相关个体之间建立联系。

定义 1 集合 A 上的关系叫做等价关系, 如果它是自反的、对称的和传递的。

等价关系在整个数学和计算机科学中都很重要。原因之一就是 在等价关系中, 当两个元素相联系就可以说它们是等价的。

定义 2 如果两个元素 a 和 b 由于等价关系而相联系, 则称它们是等价的。记法 $a \sim b$ 通常用来表示对于某个特定的等价关系来说, a 和 b 是等价的元素。

为了使等价元素的概念有意义, 每个元素都应该等价于它自身, 因为对于等价关系来说, 自反性是一定成立的。在等价关系中说 a 和 b 相互联系也是正确的(而不仅是 a 联系于 b), 因为如果 a 联系于 b , 由对称性, b 也联系于 a 。此外, 因为等价关系是传递的, 如果 a 和 b 等价且 b 和 c 等价, 则可得出 a 和 c 也是等价的。

例 1~例 5 说明了等价关系的概念。

例 1 设 R 是整数集上的关系, 满足 aRb , 当且仅当 $a=b$ 或 $a=-b$ 。在 5.1 节我们证明了 R 是自反的、对称的和传递的。因此 R 是等价关系。 ■

例2 设 R 是实数集上的关系, 满足 aRb , 当且仅当 $a-b$ 是整数。 R 是等价关系吗?

解 因为对所有的实数 a , $a-a=0$ 是整数, 即对所有的实数有 aRa , 因此 R 是自反的。现在假设 aRb , 那么 $a-b$ 是整数, 所以 $b-a$ 也是整数。因此有 bRa 。 R 是对称的。如果 aRb 和 bRc , 那么 $a-b$ 和 $b-c$ 是整数, 所以 $a-c=(a-b)+(b-c)$ 也是整数。因此 aRc 。于是, R 是传递的。综上所述, R 是等价关系。 ■

最广泛使用的等价关系之一是模 m 同余关系, 其中 m 是大于 1 的正整数。

例3 模 m 同余。设 m 是大于 1 的正整数。证明以下关系是整数集上的等价关系。

$$R = \{(a, b) \mid a \equiv b \pmod{m}\}$$

解 $a \equiv b \pmod{m}$, 当且仅当 m 整除 $a-b$ 。注意 $a-a=0$ 被 m 整除, 因为 $0=0 \cdot m$ 。因此 $a \equiv a \pmod{m}$, 从而模 m 同余关系是自反的。现在假设 $a \equiv b \pmod{m}$, 那么 $a-b$ 被 m 整除, 即 $a-b=km$, 其中 k 是整数。从而 $b-a=(-k)m$, 即 $b \equiv a \pmod{m}$, 因此模 m 同余关系是对称的。下面假设 $a \equiv b \pmod{m}$ 和 $b \equiv c \pmod{m}$, 那么 m 整除 $a-b$ 和 $b-c$ 。因此, 存在整数 k 和 l , 使得 $a-b=km$ 和 $b-c=lm$ 。把这两个等式加起来得 $a-c=(a-b)+(b-c)=km+lm=(k+l)m$ 。于是, $a \equiv c \pmod{m}$, 从而模 m 同余关系是传递的。综上所述, 模 m 同余关系是等价关系。 ■

例4 设 R 是英语字母串的集合上的关系并且使得 aRb 当且仅当 $l(a)=l(b)$, 其中 $l(x)$ 是串 x 的长度。 R 是等价关系吗?

解 因为 $l(a)=l(a)$, 从而只要 a 是一个串, 就有 aRa , 故 R 是自反的。其次, 假设 aRb , 即 $l(a)=l(b)$ 。那么有 bRa , 因为 $l(b)=l(a)$, 因此 R 是对称的。最后, 假设 aRb 和 bRc , 那么有 $l(a)=l(b)$ 和 $l(b)=l(c)$ 。因此 $l(a)=l(c)$, 即 aRc , 从而 R 是传递的。由于 R 是自反的、对称的和传递的, R 是等价关系。 ■

例5 设 n 是正整数, S 是字符串集。假定 R_n 是 S 上的关系, sR_nt 当且仅当 $s=t$ 或者 s 和 t 都至少含有 n 个字符, 且 s 和 t 的前 n 个字符相同。就是说, 少于 n 个字符的字符串只和它自身以关系 R_n 相联系; 至少含有 n 个字符的字符串 s 联系于 t 当且仅当 t 也含有至少 n 个字符且以 s 最前面的 n 个字符开始。例如, 设 $n=3$, S 是所有比特串的集合, sR_3t 当 $s=t$ 或者 s 和 t 的长度至少为 3, 且前 3 个比特相同时成立。例如, $01R_301$, $00111R_300101$, 但 $01R_3010$, $01011R_301110$ 。

证明: 对所有的字符串集 S 和所有的正整数 n , R_n 是 S 上的等价关系。

解 如果 $s=s$, 则 sR_ns 只要 s 是 S 中的一个字符串, 所以关系 R_n 是自反的。如果 sR_nt , 那么或者 $s=t$ 或者 s 和 t 都至少含有 n 个字符, 且以相同的 n 个字符开始。这就意味着 tR_ns 成立。所以 R_n 是对称的。

现在假设 sR_nt 且 tR_nu 。则有, $s=t$ 或者 s 和 t 都至少含有 n 个字符且以相同的 n 个字符开始; 还有 $t=u$ 或者 t 和 u 都至少含有 n 个字符且以相同的 n 个字符开始; 由此可以推出 $s=u$ 或者 s 和 u 都至少含有 n 个字符且以相同的 n 个字符开始(因为在这种情形下, 我们知道 s, t, u 都至少有 n 个字符, 且 s 和 u 都与 t 一样以相同的 n 个字符开始)。所以 R_n 是传递的。这就证明了 R_n 是一个等价关系。 ■

在例 6 和例 7 中, 将看到两个非等价的关系。

例6 证明: 正整数上的“整除”关系不是等价关系。

解 由 5.1 节的例 9 和例 15 可知, “整除”关系是自反和传递的。但是, 由 5.1 节的例 12, 知道此关系不是对称的(例如, $2 \mid 4$ 但 $4 \nmid 2$)。所以得出, 正整数上的“整除”关系不是等价关系。 ■

例7 R 是实数集上的关系, xRy 当且仅当 x 和 y 是差小于 1 的实数, 即 $|x-y| < 1$ 。证明 R 不是等价关系。

解 R 是自反的, 因为只要 $x \in \mathbf{R}$, 就有 $|x-x|=0 < 1$ 。 R 是对称的, 因为如果 xRy , 此处 x 和 y 是实数, 那么有 $|x-y| < 1$, 这告诉我们 $|y-x|=|x-y| < 1$, 因此 yRx 。然而, R 不是等价关系, 因为它不是传递的。取 $x=2.8$, $y=1.9$, 和 $z=1.1$, 这样 $|x-y|=|2.8-1.9|=0.9 < 1$, $|y-z|=|1.9-1.1|=0.8 < 1$, 但是 $|x-z|=|2.8-1.1|=1.7 > 1$ 。这

就是说, $2.8 R 1.9$, $1.9 R 1.1$ 但 $2.8 R 1.1$ 。

5.5.3 等价类

设 A 是在你们学校所有的高中毕业生。考虑 A 上的关系 R , R 由所有的对 (x, y) 构成, 其中 x 和 y 从同一高中毕业。给定学生 x , 我们可以构造所有的关于 R 与 x 等价的学生的集合。这个集合由与 x 在同一高中毕业的所有学生构成。 A 的这个子集叫做这个关系的一个等价类。

定义 3 设 R 是集合 A 上的等价关系。与 A 中的一个元素 a 有关系的所有元素的集合叫做 a 的等价类。 A 的关于 R 的等价类记作 $[a]_R$ 。当只有一个关系被考虑时, 我们将省去下标 R 并把这个等价类写作 $[a]$ 。

换句话说, 如果 R 是集合 A 上的等价关系, 元素 a 的等价类是

$$[a]_R = \{s \mid (a, s) \in R\}$$

如果 $b \in [a]_R$, b 叫做这个等价类的代表元。一个等价类的任何元素都可以作为这个类的代表元。也就是说, 对作为这个类的代表元所选择的特定元素没有特殊要求。

例 8 对于例 1 的等价关系, 一个整数的等价类是什么?

解 在这个等价关系中, 一个整数等价于它自身和它的相反数。从而 $[a] = \{-a, a\}$ 。这个集合包含两个不同的整数, 除非 $a=0$ 。例如, $[7] = \{-7, 7\}$, $[-5] = \{-5, 5\}$, $[0] = \{0\}$ 。

例 9 对于模 4 同余关系, 0 和 1 的等价类是什么?

解 0 的等价类包含使得 $a \equiv 0 \pmod{4}$ 的所有整数 a 。这个类中的整数是被 4 整除的那些整数。因此, 对于这个关系 0 的等价类是

$$[0] = \{\dots, -8, -4, 0, 4, 8, \dots\}$$

1 的等价类包含使得 $a \equiv 1 \pmod{4}$ 的所有整数 a 。这个类中的整数是当被 4 除时余数为 1 的那些整数。因此, 对于这个关系 1 的等价类是

$$[1] = \{\dots, -7, -3, 1, 5, 9, \dots\}$$

在例 9 找到了 0 和 1 关于模 4 同余的等价类。用任何正整数 m 代替 4 很容易把例 9 加以推广。模 m 同余关系的等价类叫做模 m 同余类。整数 a 模 m 的同余类记作 $[a]_m$, 满足 $[a]_m = \{\dots, a-2m, a-m, a, a+m, a+2m, \dots\}$ 。例如, 从例 9 得出 $[0]_4 = \{\dots, -8, -4, 0, 4, 8, \dots\}$ 和 $[1]_4 = \{\dots, -7, -3, 1, 5, 9, \dots\}$ 。

例 10 对于例 5 中全体比特串集合上的等价关系 R_s 而言, 串 0111 的等价类是什么? (回忆 $s R_t$ 当且仅当 s, t 是满足如下条件的比特串: $s=t$ 或者 s 和 t 都至少含有 3 个比特, 且 s 和 t 的前 3 个比特相同。)

解 等价于 0111 的是以 011 开始, 至少含有 3 个比特的比特串。它们是 011, 0110, 0111, 01100, 01101, 01110, 01111, 等等。所以

$$[011]_{R_s} = \{011, 0110, 0111, 01100, 01101, 01110, 01111, \dots\}$$

例 11 C 程序设计语言中的标识符 在 C 语言中, 标识符是变量、函数或者其他类型的实体的名字。每个标识符是一个非空字符串, 串中的每个字符可以是 大写或小写的英文字母、数字或下划线, 而第一个字符必须为 大写或小写的英文字母。标识符的长度是任意的, 这就使得开发者可以按照自己的意愿使用一定数量的字符来命名一个实体, 比如变量。然而, 对于某些版本的 C 编译器来说, 当比较两个名字看它们是否表示同一事物时, 实际检查的字符个数是有限制的。例如, 当两个标识符的前 31 个字符相同时, 标准 C 编译器就认为它们是相同的。所以, 开发者就必须小心, 不要使用前 31 个字符相同的标识符来表示不同的事物。我们可以看出, 如果两个标识符由例 5 中的关系 R_{31} 联系起来, 那么它们将被视做相同。由例 5 知道, 在标准 C 的标识符集上, 关系 R_{31} 是一个等价关系。

现有下标识符: Number_of_tropical_storms, Number_of_named_tropical_storms, Number_of_named_tropical_storms_in_the_Atlantic_in_2005, 它们的等价类各是什么?

解 注意, 当一个标识符的长度小于 31 的时候, 根据 R_{31} 的定义, 它的等价类只包含它自身。因为标识符 `Number_of_tropical_storms` 只含有 25 个字符, 所以它的等价类只含有一个元素, 即它自己。

标识符 `Number_of_named_tropical_storms` 的长度刚好为 31。一个标识符就与它等价意味着以这 31 个字符开始。所以, 每个长度至少为 31、以 `Number_of_named_tropical_storms` 开始的标识符都与这个标识符等价。所以得出, `Number_of_named_tropical_storms` 的等价类是全体以 `Number_of_named_tropical_storms` 这 31 个字符开始的标识符。

一个标识符与 `Number_of_named_tropical_storms_in_the_Atlantic_in_2005` 等价, 当且仅当它以 `Number_of_named_tropical_storms_in_the_Atlantic_in_2005` 的前 31 个字符开始。因为这 31 个字符是 `Number_of_named_tropical_storms`, 我们看到一个标识符与 `Number_of_named_tropical_storms_in_the_Atlantic_in_2005` 等价, 当且仅当它与 `Number_of_named_tropical_storms` 等价。就是说, 最后这两个标识符的等价类是相同的。 ■

5.5.4 等价类与划分

设 A 是你们学校恰好主修一个专业的学生的集合, R 是 A 上的关系, 如果 x 和 y 是主修同一专业的学生, 则 (x, y) 属于 R 。那么正如读者可以验证的, R 是等价关系。我们可以看出 R 将 A 中的所有学生分成不相交的子集, 其中每个子集包含某个特定专业的学生。例如, 一个子集包含所有(只主修)计算机专业的学生, 第二个子集包含所有历史专业的学生。而且这些子集是 R 的等价类。这个例子说明一个等价关系的等价类怎样把一个集合划分成不相交的非空子集。我们将在下面的讨论中把这些概念进一步精确化。

设 R 是集合 A 上的等价关系。下面的定理证明 A 的两个元素的等价类或是相等或是不相交。

定理 1 设 R 是集合 A 上的等价关系, 下面的命题是等价的。

- (i) aRb (ii) $[a] = [b]$ (iii) $[a] \cap [b] \neq \emptyset$

证 首先证明(i)推出(ii)。假设 aRb , 我们将通过 $[a] \subseteq [b]$ 和 $[b] \subseteq [a]$ 来证明 $[a] = [b]$ 。假设 $c \in [a]$, 那么 aRc 。因为 aRb 和 R 的对称性, 有 bRa 。又由于 R 是传递的以及 bRa 和 aRc , 就得到 bRc , 因而有 $c \in [b]$ 。这就证明了 $[a] \subseteq [b]$ 。类似地, 可证明 $[b] \subseteq [a]$, 证明留给读者作为练习。

其次我们将证明(ii)推出(iii)。假设 $[a] = [b]$ 。这就证明了 $[a] \cap [b] \neq \emptyset$, 因为 $[a]$ 是非空的(由 R 的自反性 $a \in [a]$)。

下面证明(iii)推出(i)。假设 $[a] \cap [b] \neq \emptyset$, 那么存在元素 c 满足 $c \in [a]$ 和 $c \in [b]$ 。换句话说, aRc 和 bRc 。由对称性有 cRb 。再根据传递性, 由 aRc 和 cRb 就有 aRb 。

因为(i)推出(ii), (ii)推出(iii), (iii)推出(i), 三个命题(i)、(ii)和(iii)是等价的。 □

我们现在将显示一个等价关系怎样划分一个集合。设 R 是集合 A 上的等价关系, R 的所有等价类的并集就是 A 的全部, 因为 A 的每个元素 a 都在它自己的等价类即 $[a]_R$ 中。换句话说,

$$\bigcup_{a \in A} [a]_R = A$$

此外, 由定理 1, 这些等价类或是相等的或是不相交的, 因此当 $[a]_R \neq [b]_R$ 时,

$$[a]_R \cap [b]_R = \emptyset$$

这两个观察证明等价类构成 A 的划分, 因为它们将 A 分成不相交的子集。更确切地说, 集合 S 的划分是一族 S 的不相交的非空子集, 且 S 就是它们的并。换句话说, 一族子集 A_i , $i \in I$, (其中 I 是指标集)构成 S 的划分, 当且仅当

$$A_i \neq \emptyset, \quad i \in I$$

$$A_i \cap A_j = \emptyset, \quad i \neq j$$

和

$$\bigcup_{i \in I} A_i = S$$

(这里符号 $\bigcup_{i \in I} A_i$ 表示对于所有的 $i \in I$ 集合 A_i 的并集。)图 5-13 说明了集合的划分的概念。

例 12 假设 $S = \{1, 2, 3, 4, 5, 6\}$, 一族集合 $A_1 = \{1, 2, 3\}$, $A_2 = \{4, 5\}$ 和 $A_3 = \{6\}$ 构成 S 的一个划分, 因为这些集合是不相交的, 且它们的并是 S 。■

我们已经看到集合上等价关系的等价类构成这个集合的划分。这个划分中的子集就是这些等价类。反过来, 可以用集合的每个划分来构成等价关系。两个元素关于这个关系是等价的, 当且仅当它们在这个划分的同一子集中。

为看到这一点, 假设 $\{A_i \mid i \in I\}$ 是 S 的划分。设 R 是 S 上的由有序对 (x, y) 组成的等价关系, 其中 x 和 y 属于这个划分的同一子集 A_i 。为证明 R 是等价关系, 我们必须证明 R 是自反的、对称的和传递的。

对于每个 $a \in S$ 有 $(a, a) \in R$, 因为 a 和它自己在同一子集中, 因此 R 是自反的。如果 $(a, b) \in R$, 那么 b 和 a 在这个划分的同一子集中, 因此有 $(b, a) \in R$ 。从而 R 是对称的。如果 $(a, b) \in R$ 和 $(b, c) \in R$, 那么 a 和 b 在这个划分的同一子集中, 比如是 X , b 和 c 也在这个划分的同一子集中, 比如是 Y 。因为划分的子集是不相交的, 并且 b 属于 X 和 Y , 必有 $X = Y$ 。因而 a 和 c 属于这个划分的同一子集, 即 $(a, c) \in R$ 。于是 R 是传递的。

这就证明了 R 是等价关系。 R 的等价类由 S 的子集构成, 这些子集包含了 S 的有关关系的元素, 根据 R 的定义, 它们就是划分的子集。定理 2 是对等价关系和划分之间建立的这种联系的总结。

定理 2 设 R 是集合 S 上的等价关系。那么 R 的等价类构成 S 的划分。相反, 给定集合 S 的划分 $\{A_i \mid i \in I\}$, 存在着等价关系 R , 它以集合 $A_i (i \in I)$ 作为它的等价类。

例 13 说明了怎样从一个划分构造一个等价关系。

例 13 $A_1 = \{1, 2, 3\}$, $A_2 = \{4, 5\}$, $A_3 = \{6\}$ 是例 12 给出的集合 $S = \{1, 2, 3, 4, 5, 6\}$ 的划分, 列出这个划分所产生的等价关系 R 中的有序对。

解 划分中的子集是 R 的等价类。对 $(a, b) \in R$, 当且仅当 a 和 b 在划分的同一个子集中。由于 $A_1 = \{1, 2, 3\}$ 是一个等价类, 因此, 对 $(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)$ 属于 R ; 由于 $A_2 = \{4, 5\}$ 是一个等价类, 因此, 对 $(4, 4), (4, 5), (5, 4), (5, 5)$ 也属于 R ; 最后, 由于 $\{6\}$ 是一个等价类, 对 $(6, 6)$ 属于 R 。此外没有其他的对属于 R 。■

模 m 同余类对定理 2 提供了一个有用的说明。存在 m 个不同的模 m 同余类, 对应于当一个整数除以 m 时可能得到的 m 个不同的余数。这 m 个同余类记作 $[0]_m, [1]_m, \dots, [m-1]_m$ 。它们构成了整数集合的划分。

例 14 模 4 同余产生的整数划分中的集合是什么?

解 存在 4 个同余类, 对应于 $[0]_4, [1]_4, [2]_4$ 和 $[3]_4$, 它们是集合

$$[0]_4 = \{\dots, -8, -4, 0, 4, 8, \dots\}$$

$$[1]_4 = \{\dots, -7, -3, 1, 5, 9, \dots\}$$

$$[2]_4 = \{\dots, -6, -2, 2, 6, 10, \dots\}$$

$$[3]_4 = \{\dots, -5, -1, 3, 7, 11, \dots\}$$

这些同余类是不相交的, 并且每个整数恰好在它们之中的一个。换句话说, 正如定理 2 所说, 这些同余类构成了一个划分。■

现在举一个例子: 从这个集合上的等价关系产生的全体串集合的划分。

例 15 设 R_3 为例 5 中的关系。在全体比特串集合上, 由 R_3 产生的对全体比特串集合的划分中是哪些集合? (回忆 s, t 是比特串, $s R_3 t$, 如果 $s = t$ 或者 s 和 t 都至少含有 3 个比特, 且它们的前 3 个比特相同。)

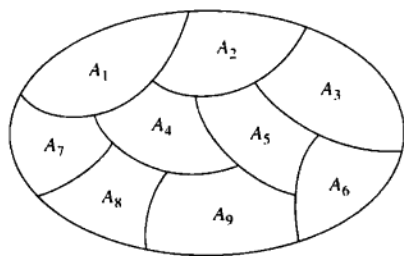


图 5-13 集合的划分

解 注意到每个长度小于3的比特串只和它自身等价, 有 $[\lambda]_{R_1} = \{\lambda\}$, $[0]_{R_1} = \{0\}$, $[1]_{R_1} = \{1\}$, $[00]_{R_1} = \{00\}$, $[01]_{R_1} = \{01\}$, $[10]_{R_1} = \{10\}$, $[11]_{R_1} = \{11\}$. 再注意到每个长度大于或等于3的比特串必和000, 001, 010, 011, 100, 101, 110, 111这8个比特串之一等价, 我们有

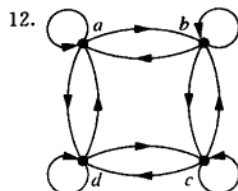
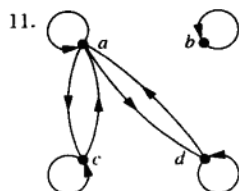
$$\begin{aligned} [000]_{R_1} &= \{000, 0000, 0001, 00000, 00001, 00010, 00011, \dots\} \\ [001]_{R_1} &= \{001, 0010, 0011, 00100, 00101, 00110, 00111, \dots\} \\ [010]_{R_1} &= \{010, 0100, 0101, 01000, 01001, 01010, 01011, \dots\} \\ [011]_{R_1} &= \{011, 0110, 0111, 01100, 01101, 01110, 01111, \dots\} \\ [100]_{R_1} &= \{100, 1000, 1001, 10000, 10001, 10010, 10011, \dots\} \\ [101]_{R_1} &= \{101, 1010, 1011, 10100, 10101, 10110, 10111, \dots\} \\ [110]_{R_1} &= \{110, 1100, 1101, 11000, 11001, 11010, 11011, \dots\} \\ [111]_{R_1} &= \{111, 1110, 1111, 11100, 11101, 11110, 11111, \dots\} \end{aligned}$$

这15个等价类是不相交的, 每个比特串都恰好属于它们之一。正如定理2告诉我们的, 这些等价类是全体比特串构成的集合的一个划分。 ■

练习

- 下面是 $\{0, 1, 2, 3\}$ 上的关系, 其中哪些是等价关系? 确定一个等价关系的性质, 这些性质是其他关系所欠缺的。
 - $\{(0, 0), (1, 1), (2, 2), (3, 3)\}$
 - $\{(0, 0), (0, 2), (2, 0), (2, 2), (2, 3), (3, 2), (3, 3)\}$
 - $\{(0, 0), (1, 1), (1, 2), (2, 1), (2, 2), (3, 3)\}$
 - $\{(0, 0), (1, 1), (1, 3), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)\}$
 - $\{(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 2), (3, 3)\}$
- 下面是从 $\mathbf{Z}$ 到 $\mathbf{Z}$ 的所有函数集合上的关系, 其中哪些是等价关系? 确定一个等价关系的性质, 这些性质是其他关系所欠缺的。
 - $\{(f, g) \mid f(1)=g(1)\}$
 - $\{(f, g) \mid f(0)=g(0) \text{ 或 } f(1)=g(1)\}$
 - $\{(f, g) \mid \text{对所有的 } x \in \mathbf{Z}, f(x)-g(x)=1\}$
 - $\{(f, g) \mid \text{对某个 } C \in \mathbf{Z}, \text{对所有的 } x \in \mathbf{Z}, f(x)-g(x)=C\}$
 - $\{(f, g) \mid f(0)=g(1) \text{ 且 } f(1)=g(0)\}$
- 在大学校园里的建筑物集合上定义3个等价关系, 确定关于这些等价关系的等价类。
- 证明: 在所有复合命题集合上的逻辑等价关系是等价关系。这里T和F的等价类是什么?
- 假设A是非空集合, f是以A作为定义域的函数, 设R是A上的关系, 若 $f(x)=f(y)$, 则 (x, y) 属于R。
 - 证明R是A上的等价关系。
 - R的等价类是什么?
- 设R是长度至少为3的所有二进位串的集合上的关系, R由对 (x, y) 构成, 其中x和y在它们的前3位相同。证明R是等价关系。
- 设R是长度至少为3的所有二进位串的集合上的关系, R由序对 (x, y) 构成, 其中x和y在它们的第1和第3位字符相同。证明R是等价关系。
- 设R是正整数的有序对集合上的关系, $((a, b), (c, d)) \in R$ 当且仅当 $a+d=b+c$ 。证明R是等价关系。
- (需要微积分)
 - 设R是从 $\mathbf{R}$ 到 $\mathbf{R}$ 的所有可微分函数的集合上的关系, R由所有的对 (f, g) 构成, 其中对所有实数x, $f'(x)=g'(x)$ 。证明R是等价关系。
 - 什么函数与函数 $f(x)=x^2$ 在同一个等价类中?
- 设R是所有URL(或Web地址)集合上的关系, xRy 当且仅当在x的Web页与在y的Web页相同, 证明R是等价关系。

在练习11~12中确定具有所有有向图的关系是否为等价关系。



13. 设 R 是所有二进位串的集合上的关系, sRt 当且仅当 s 和 t 包含相同个数的 1, 证明 R 是等价关系。

14. 对于练习 13 中的等价关系, 二进位串 011 的等价类是什么?

15. 当 n 为下列各数时, 同余类 $[n]_5$ (即 n 关于模 5 同余的等价类) 是什么?

a) 2

b) 3

c) 6

d) -3

16. 描述每一个模 6 同余类。

17. a) 对于练习 8 中的等价关系 (1, 2) 的等价类是什么?

b) 对于练习 8 中的等价关系 R 解释等价类的含义。[提示: 差 $a-b$ 对应 $(a, b)_0$ 。]

18. 下面哪些子集族是 $\{1, 2, 3, 4, 5, 6\}$ 的划分?

a) $\{1, 2\}, \{2, 3, 4\}, \{4, 5, 6\}$

b) $\{1\}, \{2, 3, 6\}, \{4\}, \{5\}$

c) $\{2, 4, 6\}, \{1, 3, 5\}$

d) $\{1, 4, 5\}, \{2, 6\}$

19. 下面哪些子集族是 8 位二进位串的集合上的划分?

a) 以 1 开始的二进位串的集合, 以 00 开始的二进位串的集合, 以 01 开始的二进位串的集合。

b) 包含串 00 的二进位串的集合, 包含串 01 的二进位串的集合, 包含串 10 的二进位串的集合, 包含串 11 的二进位串的集合。

c) 以 00 结尾的二进位串的集合, 以 01 结尾的二进位串的集合, 以 10 结尾的二进位串的集合, 以 11 结尾的二进位串的集合。

d) 以 111 结尾的二进位串的集合, 以 011 结尾的二进位串的集合, 以 00 结尾的二进位串的集合。

e) 含 $3k+1$ 个 1 的二进位串的集合, 其中 k 为非负整数; 含 $3k+1$ 个 1 的二进位串的集合, 其中 k 为非负整数; 含 $3k+2$ 个 1 的二进位串的集合, 其中 k 是正整数。

20. 下面哪些是整数的有序对的集合 $\mathbb{Z} \times \mathbb{Z}$ 的划分?

a) x 或 y 是奇数的 (x, y) 对的集合; x 是偶数的 (x, y) 对的集合; y 是偶数的 (x, y) 对的集合。

b) x 和 y 都是奇数的 (x, y) 对的集合; x 和 y 只有一个奇数的 (x, y) 对的集合; x 和 y 都是偶数的 (x, y) 对的集合。

c) x 是正数的 (x, y) 对的集合; y 是正数的 (x, y) 对的集合; x 和 y 都是负数的 (x, y) 对的集合。

d) x 和 y 都被 3 整除的 (x, y) 对的集合; x 被 3 整除且 y 不被 3 整除的 (x, y) 对的集合; x 不被 3 整除且 y 被 3 整除的 (x, y) 对的集合; x 和 y 都不被 3 整除的 (x, y) 对的集合。

e) $x > 0$ 且 $y > 0$ 的 (x, y) 对的集合; $x > 0$ 且 $y \leq 0$ 的 (x, y) 对的集合; $x \leq 0$ 且 $y > 0$ 的 (x, y) 对的集合; $x \leq 0$ 且 $y \leq 0$ 的 (x, y) 对的集合。

f) $x \neq 0$ 且 $y \neq 0$ 的 (x, y) 对的集合; $x = 0$ 且 $y \neq 0$ 的 (x, y) 对的集合; $x \neq 0$ 且 $y = 0$ 的 (x, y) 对的集合。

21. 列出由 $\{0, 1, 2, 3, 4, 5\}$ 的划分产生的等价关系中的有序对。

a) $\{0\}, \{1, 2\}, \{3, 4, 5\}$

b) $\{0, 1\}, \{2, 3\}, \{4, 5\}$

c) $\{0, 1, 2\}, \{3, 4, 5\}$

d) $\{0\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}$

如果在划分 P_1 中的每个集合都是划分 P_2 中每个集合的子集, 则 P_1 叫做 P_2 的加细。

22. 证明: 由模 6 同余类构成的划分是模 3 同余类构成的划分的加细。

23. 证明: 对于 16 位二进位串的集合, 最后 8 位相同的二进位串的等价类所构成的划分是由最后 4 位相同的二进位串的等价类所构成的划分的加细。

在练习 24 中, R_n 表示例 5 中定义的等价关系族。回忆字符串 s, t 满足 $sR_n t$, 如果 $s=t$ 或者 s 和 t 至少含有 n 个字符, 且它们的前 5 个字符相同。

24. 证明: 由等价关系 R_3 决定的标识符等价类构成的 C 语言中所有标识符的划分是由等价关系 R_8 决定的标识符等价类构成的划分的加细。(旧的 C 语言编译器只要多个标识符的前 8 个字符相同就将它们视为相同, 而标准 C 的编译器需要多个标识符的前 31 个字符相同才将它们视为相同。)

25. 求出在集合 $\{a, b, c, d, e\}$ 上包含关系 $\{(a, b), (a, c), (d, e)\}$ 的最小的等价关系。
26. 考虑例 3 的等价关系, 即 $R = \{(x, y) \mid x - y \text{ 是整数}\}$ 。
- a) 关于这个等价关系的 1 的等价类是什么?
- b) 关于这个等价关系的 $1/2$ 的等价类是什么?
- *27. 设 R 是 2×2 棋盘的所有涂色集合上的关系, 其中 4 个方格中的每一个可以被涂成红色或蓝色。 C_1 和 C_2 是被这样涂色的 2×2 棋盘, $(C_1, C_2) \in R$, 当且仅当 C_2 可以由旋转 C_1 , 或旋转 C_1 然后再翻转 C_1 得到。
- a) 证明 R 是等价关系。
- b) R 的等价类是什么?
28. 通过列举确定 3 元集上的不同的等价关系个数。
- *29. 当我们构造一个关系的自反闭包的对称闭包的传递闭包时, 一定得到一个等价关系吗?
30. 假设我们使用定理 2 从一个等价关系 R 构造一个划分 P 。如果再次使用定理 2 从 P 构造一个等价关系, 那么结果的等价关系 R 是什么?
31. 设计一个算法找出包含一个给定关系的最小的等价关系。

5.6 偏序

5.6.1 引言

我们常常用关系对集合的某些元素或全体元素排序。例如, 使用包含着字对 (x, y) 的关系对字排序, 其中 x 按照字典顺序排在 y 的前面。使用包含着对 (x, y) 的关系安排课题, 其中 x 和 y 是课题中的任务并且 x 必须在 y 开始之前完成。使用包含着对 (x, y) 的关系排序整数集合, 其中 x 小于 y 。当我们把所有形如 (x, x) 的对加到这些关系中时, 就得到了一个自反、反对称和传递的关系。这些都是刻画元素有序集合的关系特征的性质, 而这些关系使用它们相对的大小来排序集合的元素。

定义 1 集合 S 上的关系 R , 如果它是自反的、反对称的和传递的, 就称为偏序。集合 S 与偏序 R 一起叫做偏序集, 记作 (S, R) 。

例 1 证明: “大于或等于”关系 $(\geq)$ 是整数集合上的偏序。

解 因为对所有整数 a 有 $a \geq a$, $\geq$ 是自反的。如果 $a \geq b$ 且 $b \geq a$, 那么 $a = b$, 因此 $\geq$ 是反对称的。最后, 因为 $a \geq b$ 和 $b \geq c$ 推出 $a \geq c$, 所以 $\geq$ 是传递的。从而 $\geq$ 是整数集合上的偏序且 $(\mathbb{Z}, \geq)$ 是偏序集。

例 2 整除关系 “ $\mid$ ” 是正整数集合上的偏序, 因为正如 5.1 节所述, 它是自反的、反对称的和传递的。我们看到 $(\mathbb{Z}^+, \mid)$ 是偏序集 ($\mathbb{Z}^+$ 表示正整数集合)。

例 3 证明: 包含关系 $\subseteq$ 是集合 S 的幂集上的偏序。

解 因为只要 A 是 S 的子集, 就有 $A \subseteq A$, $\subseteq$ 是自反的。因为 $A \subseteq B$ 和 $B \subseteq A$ 推出 $A = B$, 因此它是反对称的。最后, 因为 $A \subseteq B$ 和 $B \subseteq C$ 推出 $A \subseteq C$, $\subseteq$ 是传递的。因此, $\subseteq$ 是 $P(S)$ 上的偏序, 且 $(P(S), \subseteq)$ 是偏序集。

例 4 举出了一个不是偏序的关系。

例 4 设 R 是由人构成的集合上的关系, xRy 当且仅当 x 和 y 是人, 且 x 年纪大于 y 。证明: R 不是偏序。

解 注意 R 是反对称的, 因为如果有一个人 x 比另一个人 y 年长, 那么 y 就不会比 x 年长。也就是说, 如果 xRy , 那么 $y \not R x$ 。关系 R 是传递的, 因为如果 x 比 y 年长, 而 y 又比 z 年长, 那么 x 肯定比 z 年长。就是说, 如果 xRy , yRz , 那么 xRz 。但是, R 不是自反的, 因为没有谁会比自己年长。即对于所有的人 x , $x \not R x$ 。这就意味着 R 不是偏序。

在一个偏序集中记号 $a \leq b$ 表示 $(a, b) \in R$ 。使用这个记号是由于“小于或等于”关系是偏序关系的范例。(注意符号 $\leq$ 用来表示任意偏序集的关系, 并不仅仅是“小于或等于”关系。)记号 $a < b$ 表示 $a \leq b$, 但 $a \neq b$ 。如果 $a < b$, 我们说“ a 小于 b ”或“ b 大于 a ”。

当 a 与 b 是偏序集 $(S, \leq)$ 的元素时, 不一定有 $a \leq b$ 或 $b \leq a$ 。例如, 在 $(P(\mathbb{Z}), \subseteq)$ 中,

$\{1, 2\}$ 与 $\{1, 3\}$ 没有关系, 反之亦然, 因为没有集合被另一个集合包含。类似地, 在 $(\mathbb{Z}^+, |)$ 中, 2与3没关系, 3也与2没关系, 因为 $2 \nmid 3$ 且 $3 \nmid 2$ 。由此得到定义2。

定义2 偏序集 $(S, \leq)$ 的元素 a 和 b 叫做可比的, 如果 $a \leq b$ 或 $b \leq a$ 。当 a 和 b 是 S 的元素并且既没有 $a \leq b$, 也没有 $b \leq a$, 则称 a 与 b 是不可比的。

例5 在偏序集 $(\mathbb{Z}^+, |)$ 中, 整数3和9是可比的吗? 5和7是可比的吗?

解 整数3和9是可比的, 因为 $3 \mid 9$ 。整数5和7是不可比的, 因为 $5 \nmid 7$ 且 $7 \nmid 5$ 。 ■

用形容词“部分的(偏的)”描述偏序是由于一些对元素可能是不可比的。当集合中的每对元素都可比时, 这个关系叫做全序。

定义3 如果 $(S, \leq)$ 是偏序集, 且 S 的每对元素都是可比的, 则 S 叫做全序集或线序集, 且 $\leq$ 叫做全序或线序。一个全序集也叫做链。

例6 偏序集 $(\mathbb{Z}, \leq)$ 是全序集, 因为只要 a 和 b 是整数, 就有 $a \leq b$ 或 $b \leq a$ 。 ■

例7 偏序集 $(\mathbb{Z}^+, |)$ 不是全序集, 因为它包含着不可比的元素, 例如5和7。 ■

我们注意到 $(\mathbb{Z}^+, \leq)$ 是良序的, 其中 $\leq$ 是通常的“小于或等于”关系。我们现在定义良序集。

定义4 对于偏序集 $(S, \leq)$, 如果 $\leq$ 是全序, 并且 S 的每个非空子集都有一个最小元素, 就称它为良序集。

例8 正整数的有序对的集合 $\mathbb{Z}^+ \times \mathbb{Z}^+$ 与 $\leq$ 构成良序集。对于 (a_1, a_2) 和 (b_1, b_2) , 如果 $a_1 < b_1$, 或如果 $a_1 = b_1$ 且 $a_2 \leq b_2$ (字典顺序), 则 $(a_1, a_2) \leq (b_1, b_2)$ 。有关的验证留作节后的练习。集合 $\mathbb{Z}$ 与通常的 $\leq$ 序不是良序的, 因为负整数集合是 $\mathbb{Z}$ 的子集, 但没有最小元素。 ■

现在我们叙述使用良序归纳原理证明关于一个良序集的结果这个证明技术是有效的。

定理1 良序归纳原理 设 S 是一个良序集, 如果下述条件成立:

归纳步骤: 对一切 $y \in S$, 如果 $P(x)$ 对所有 $x \in S$ 且 $x < y$ 为真, 则 $P(y)$ 为真。

那么 $P(x)$ 对所有的 $x \in S$ 为真。

证 假若 $P(x)$ 不对所有的 $x \in S$ 为真。那么存在一个元素 $y \in S$ 使得 $P(y)$ 为假。于是集合 $A = \{x \in S \mid P(x) \text{ 为假}\}$ 是非空的。因为 S 是良序的, 集合 A 有最小元素 a 。根据 a 是选自 A 的最小元素, 我们知道对所有的 $x \in S(x < a)$ 都有 $P(x)$ 为真。由归纳步这就可以推出 $P(a)$ 为真。这个矛盾就证明了 $P(x)$ 必须对所有 $x \in S$ 为真。 □

良序归纳原理对证明关于良序集的结果是一种通用的技术。即使可以使用关于正整数集合的数学归纳法证明一个定理时, 使用良序归纳原理甚至可能更简单。

5.6.2 字典顺序

字典中的字按照字母顺序或字典顺序排列, 字典顺序是以字母表中的字母顺序为基础的。这是从一个集合上的偏序构造一个集合上的串的序的特殊情况。我们将说明这种构造在任一个偏序集上是怎样做的。

首先, 我们将说明怎样在两个偏序集 $(A_1, \leq_1)$ 和 $(A_2, \leq_2)$ 的笛卡儿积上构造一个偏序。在 $A_1 \times A_2$ 上的字典顺序定义如下: 如果第一个对的第一个元素(在 A_1 中)小于第二个对的第一个元素, 或者第一个元素相等, 但是第一个对的第二个元素(在 A_2 中)小于第二个对的第二个元素, 那么第一个对小于第二个对。换句话说, (a_1, a_2) 小于 (b_1, b_2) , 即

$$(a_1, a_2) < (b_1, b_2)$$

或者 $a_1 < b_1$, 或者 $a_1 = b_1$ 且 $a_2 < b_2$ 。

把相等加到 $A_1 \times A_2$ 上的序 $<$ 就得到偏序 $\leq$ 。这个验证留作练习。

例9 确定在偏序集 $(\mathbb{Z} \times \mathbb{Z}, \leq)$ 中是否有 $(3, 5) < (4, 8)$, $(3, 8) < (4, 5)$ 和 $(4, 9) < (4, 11)$? 这里的 $\leq$ 是从 $\mathbb{Z}$ 上通常的 $\leq$ 关系构造的字典顺序。

解 因为 $3 < 4$, 故而 $(3, 5) < (4, 8)$ 且 $(3, 8) < (4, 5)$ 。因为 $(4, 9)$ 与 $(4, 11)$ 的第一元素相同, 但是 $9 < 11$, 我们有 $(4, 9) < (4, 11)$ 。 ■

图5-14明显地显示了 $\mathbb{Z}^+ \times \mathbb{Z}^+$ 中比 $(3, 4)$ 小的有序对的集合。

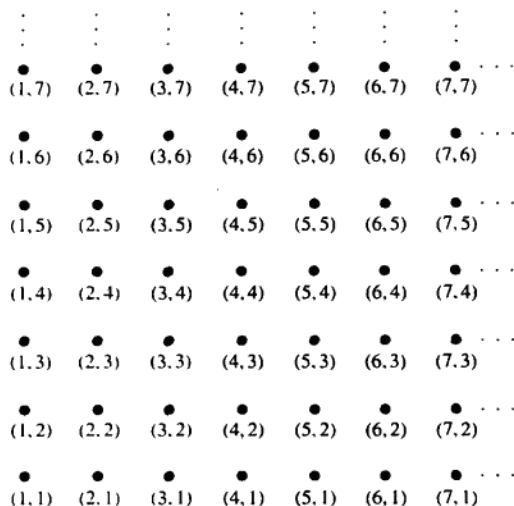


图 5-14 按照字典顺序比(3, 4)小的有序对

可以在 n 个偏序集 $(A_1, \leq_1), (A_2, \leq_2), \dots, (A_n, \leq_n)$ 的笛卡儿积上定义字典顺序。如下定义 $A_1 \times A_2 \times \dots \times A_n$ 上的偏序 $\leq$: 如果 $a_1 <_1 b_1$, 或者存在整数 $i > 0$ 使得 $a_1 = b_1, \dots, a_i = b_i$, 且 $a_{i+1} <_{i+1} b_{i+1}$, 那么

$$(a_1, a_2, \dots, a_n) < (b_1, b_2, \dots, b_n)$$

如果 $a_1 <_1 b_1$, 或者如果存在整数 $i > 0$, 使得 $a_1 = b_1, \dots, a_i = b_i$, 且 $a_{i+1} <_{i+1} b_{i+1}$ 。换句话说, 如果在两个 n 元组不同元素出现的第一位置上第一个 n 元组的元素小于第二个 n 元组的元素, 那么第一个 n 元组小于第二个 n 元组。

例 10 注意 $(1, 2, 3, 5) < (1, 2, 4, 3)$, 因为这些 4 元组的前两位相同, 但是第一个 4 元组的第三位 3 小于第二个 4 元组的第三位 4 (这里的 4 元组上的字典顺序是整数集合上的通常的“小于或等于”关系导出的字典顺序)。

我们现在可以定义串上的字典顺序。考虑偏序集 S 上的串 $a_1 a_2 \dots a_m$ 和 $b_1 b_2 \dots b_n$, 假定这些串不相等。设 t 是 m, n 中较小的数。定义字典顺序如下: $a_1 a_2 \dots a_m$ 小于 $b_1 b_2 \dots b_n$, 当且仅当

$$(a_1, a_2, \dots, a_t) < (b_1, b_2, \dots, b_t) \quad \text{或者}$$

$$(a_1, a_2, \dots, a_t) = (b_1, b_2, \dots, b_t) \text{ 并且 } m < n$$

其中这个不等式中的 $<$ 表示 S' 中的字典顺序。换句话说, 为确定两个不同串的序, 较长的串被切到较短的串的长度 t , 即 $t = \min(m, n)$ 。然后使用 S' 上的字典顺序比较每个串的前 t 位组成的 t 元组。如果对应于第一个串的 t 元组小于第二个串的 t 元组, 或者这两个 t 元组相等, 但是第二个串更长, 那么第一个串小于第二个串。这是偏序的验证留给读者作为练习。

例 11 考虑小写英语字母串构成的集合。使用在字母表中的字母序, 可以构造在串的集合上的字典顺序。如果两个串第一次出现不同字母时, 第一个串的字母先于第二个串的字母, 或者如果第一个串和第二个串在所有的位都相同, 但是第二个串有更多的字母, 那么, 第一个串小于第二个串。这种排序和字典使用的排序相同。例如,

$$\text{discreet} < \text{discrete}$$

因为这两个串在第 7 位首次出现不同字母, 并且 $e < t$ 。

$$\text{discreet} < \text{discreetness}$$

因为这两个串前 8 个字母相同, 但是第二个串更长。此外,

$$\text{discrete} < \text{discretion}$$

因为

$discrete < discreti$

5.6.3 哈塞图

在有穷偏序集的有向图中有许多边可以不必显示出来，因为它们是必须存在的。例如，考虑在集合 $\{1, 2, 3, 4\}$ 上的偏序 $\{(a, b) \mid a \leq b\}$ 的有向图，参见图 5-15a。因为这个关系是偏序，它是自反的并且有向图在所有的顶点都有环。因此，我们不必显示这些环，因为它们是必须出现的。在图 5-15b 中没有显示这些环。由于一个偏序是传递的，我们不必显示那些由于传递性而必须出现的边。例如，在图 5-15c 中没有显示边 $(1, 3)$ ， $(1, 4)$ 和 $(2, 4)$ ，因为它们是必须出现的。如果假设所有边的方向是向上的(如图中所示)，我们不必显示边的方向；图 5-15c 没有显示方向。

一般说来，我们可以使用下面的过程表示一个有穷集上的偏序。从这个关系的有向图开始。由于偏序是自反的，每个顶点有一个环，移走这些环。移走所有由于传递性出现的边，因为偏序是传递的，这些边是必须出现的。例如，如果 (a, b) 和 (b, c) 在偏序中，移走边 (a, c) ，因为它必须要出现。此外，如果 (c, d) 也在偏序中，移走边 (a, d) ，因为它也是必须出现的。最后，排列每条边使得它的起点在终点的下面(正如在纸上所画的)。移走所有的有向边上的箭头，因为所有的边向上指向它们的终点。(只有和偏序集的覆盖关系中的对所对应的边留下来。)

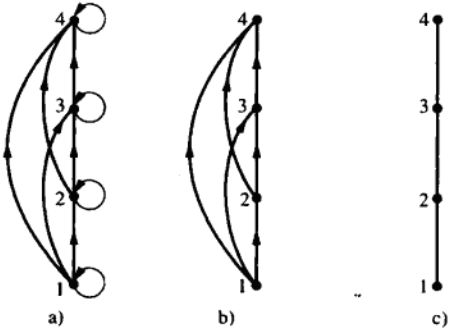


图 5-15 构造关于 $(\{1, 2, 3, 4\}, \leq)$ 的哈塞图

这些步骤是有着明确定义的，并且对于一个有穷偏序集只有有限步需要执行。当所有的步骤执行以后，就得到一个包含着足够的表示偏序信息的图。这个图叫做哈塞图，它是用 20 世纪德国数学家赫尔姆·哈塞的名字命名的。

例 12 画出表示 $\{1, 2, 3, 4, 6, 8, 12\}$ 上的偏序 $\{(a, b) \mid a \text{ 整除 } b\}$ 的哈塞图。

解 由这个偏序的有向图开始，如图 5-16a 所示。移走所有的环，如图 5-16b 所示。然后删除所有由传递性导出的边。这些边是 $(1, 4)$ ， $(1, 6)$ ， $(1, 8)$ ， $(1, 12)$ ， $(2, 8)$ ， $(2, 12)$ 和 $(3, 12)$ 。排列所有的边使得方向向上，并且删除所有的箭头得到哈塞图。结果如图 5-16c 所示。

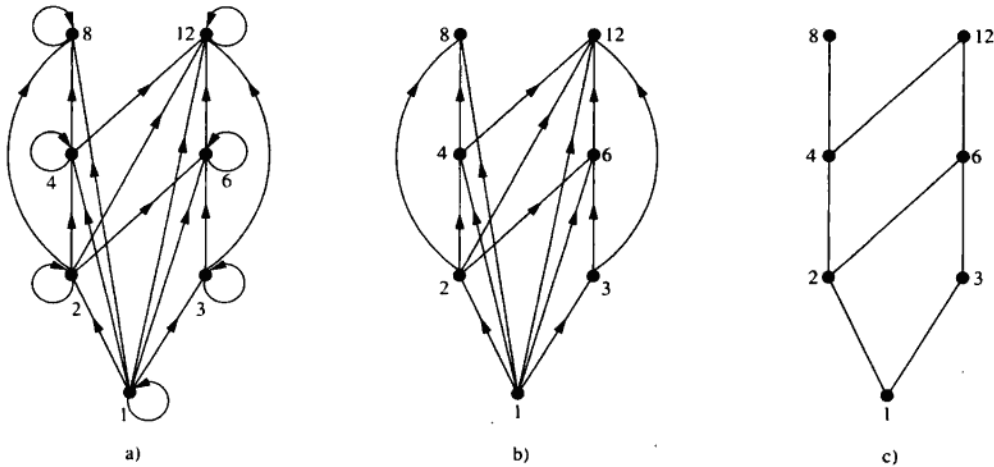


图 5-16 构造 $(\{1, 2, 3, 4, 6, 8, 12\}, \mid)$ 上的哈塞图

例 13 画出幂集 $P(S)$ 上的偏序 $\{(A, B) \mid A \subseteq B\}$ 的哈塞图, 其中 $S = \{a, b, c\}$ 。

解 关于这个偏序的哈塞图是由相关的有向图得到的, 先删除所有的环和所有由传递性产生的边, 即 $(\emptyset, \{a, b\})$, $(\emptyset, \{a, c\})$, $(\emptyset, \{b, c\})$, $(\emptyset, \{a, b, c\})$, $(\{a\}, \{a, b, c\})$, $(\{b\}, \{a, b, c\})$ 和 $(\{c\}, \{a, b, c\})$ 。最后, 使所有的边方向向上并删除箭头。得到的哈塞图如图 5-17 所示。

5.6.4 极大元素与极小元素

具有极值性质的偏序集元素有许多重要应用。偏序集的一个元素叫做极大的, 当它不小于这个偏序集的任何其他元素。即 a 在偏序集 $(S, \leq)$ 中是极大的, 当不存在 $b \in S$ 使得 $a < b$ 。类似地, 偏序集的一个元素叫做极小的, 如果它不大于这个偏序集的任何其他元素。即 a 在偏序集 $(S, \leq)$ 中是极小的, 如果不存在 $b \in S$ 使得 $b < a$ 。使用哈塞图很容易识别极大元素与极小元素。它们是图中的“顶”元素与“底”元素。

例 14 偏序集 $(\{2, 4, 5, 10, 12, 20, 25\}, \mid)$ 的哪些元素是极大的, 哪些是极小的?

解 图 5-18 关于这个偏序集的哈塞图显示了极大元素是 12, 20 和 25, 极小元素是 2 和 5。通过这个例子可以看出, 一个偏序集可以有多于一个的极大元素和多于一个的极小元素。

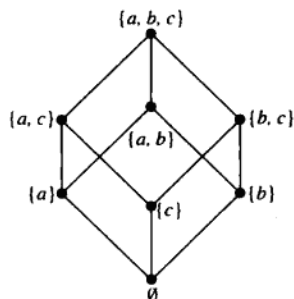


图 5-17 $(P(\{a, b, c\}), \subseteq)$ 的哈塞图

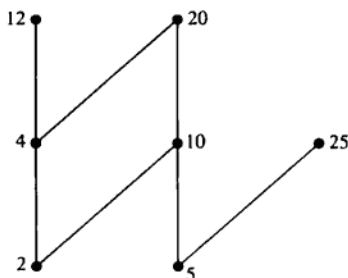


图 5-18 偏序集的哈塞图

有时在偏序集中存在一个元素大于每个其他的元素。这样的元素叫做最大元素。即 a 是偏序集 $(S, \leq)$ 的最大元素, 当对所有的 $b \in S$ 有 $b \leq a$ 。当最大元素存在时它是唯一的。类似地, 一个元素叫做最小元素, 当它小于偏序集的所有其他元素。即 a 是偏序集 $(S, \leq)$ 的最小元素, 如果对所有的 $b \in S$ 有 $a \leq b$ 。当最小元素存在时它也是唯一的。

例 15 确定图 5-19 的每个哈塞图表示的偏序集是否有最大元素和最小元素。

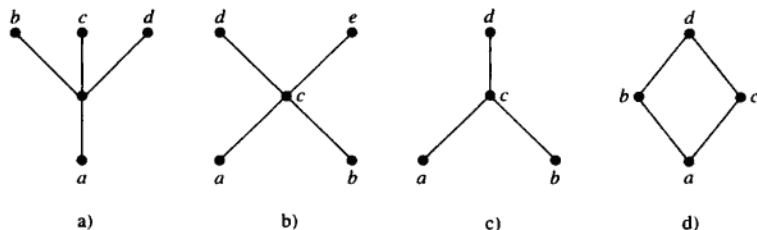


图 5-19 四个偏序集的哈塞图

解 哈塞图图 5-19a 的偏序集的最小元素是 a 。这个偏序集没有最大元素。哈塞图图 5-19b 的偏序集既没有最小元素也没有最大元素。哈塞图图 5-19c 的偏序集没有最小元素, 它的最大元素是 d 。哈塞图图 5-19d 的偏序集有最小元素 a 和最大元素 d 。

例 16 设 S 是集合。确定偏序集 $(P(S), \subseteq)$ 中是否存在最大元素与最小元素。

解 最小元素是空集, 因为对于 S 的任何子集 T , 有 $\emptyset \subseteq T$ 。集合 S 是这个偏序集的最大元素, 因为只要 T 是 S 的子集, 就有 $T \subseteq S$ 。■

例 17 在偏序集 $(\mathbb{Z}^+, |)$ 中是否存在最大元素和最小元素?

解 1 是最小元素, 因为只要 n 是正整数, 就有 $1 | n$ 。因为没有被所有正整数整除的整数, 所以不存在最大元素。■

有时候可以找到一个元素大于偏序集 $(S, \leq)$ 的子集 A 中所有的元素。如果 u 是 S 的元素使得对所有的元素 $a \in A$, 有 $a \leq u$, 那么 u 叫做 A 的一个上界。类似地, 也可能存在一个元素小于 A 中的所有其他元素。如果 l 是 S 的一个元素使得对所有的元素 $a \in A$ 有 $l \leq a$, 那么 l 叫做 A 的一个下界。

例 18 找出在图 5-20 所示哈斯图的偏序集的子集 $\{a, b, c\}$, $\{j, h\}$ 和 $\{a, c, d, f\}$ 的下界和上界。

解 $\{a, b, c\}$ 的上界是 e, f, j 和 h , 它的唯一的下界是 a 。 $\{j, h\}$ 没有上界, 它的下界是 a, b, c, d, e 和 f 。 $\{a, c, d, f\}$ 的上界是 f, h 和 j , 它的下界是 a 。■

元素 x 叫做子集 A 的最小上界, 如果 x 是一个上界并且它小于 A 的其他上界。因为如果这样的元素存在, 只存在一个, 称这个元素为最小上界是有意义的。即如果只要 $a \in A$ 就有 $a \leq x$, 并且只要 z 是 A 的上界, 就有 $x \leq z$, x 就是 A 的最小上界。类似地, 如果 y 是 A 的下界, 并且只要 z 是 A 的下界, 就有 $z \leq y$, y 就是 A 的最大下界。 A 的最大下界如果存在也是唯一的。一个子集 A 的最大下界和最小上界分别记作 $\text{glb}(A)$ 和 $\text{lub}(A)$ 。■

例 19 在图 5-20 所示偏序集中如果 $\{b, d, g\}$ 的最大下界和最小上界存在, 求出这个最大下界和最小上界。

解 $\{b, d, g\}$ 的上界是 g 和 h 。因为 $g < h$, g 是最小上界。 $\{b, d, g\}$ 的下界是 a 和 b 。因为 $a < b$, b 是最大下界。■

例 20 在偏序集 $(\mathbb{Z}^+, |)$ 中, 如果集合 $\{3, 9, 12\}$ 和 $\{1, 2, 4, 5, 10\}$ 的最大下界和最小上界存在, 求出这些最大下界和最小上界。

解 如果 $3, 9, 12$ 被一个整数整除, 那么这个整数就是 $\{3, 9, 12\}$ 的下界。这样的整数只有 1 和 3。因为 $1 | 3$, 3 是 $\{3, 9, 12\}$ 的最大下界。集合 $\{1, 2, 4, 5, 10\}$ 关系到 1 的下界只有 1, 因此 1 是 $\{1, 2, 4, 5, 10\}$ 的最大下界。

一个整数是 $\{3, 9, 12\}$ 的上界, 当且仅当它被 3, 9 和 12 整除。具有这种性质的整数就是那些被 3, 9 和 12 的最小公倍数 36 整除的整数。因此, 36 是 $\{3, 9, 12\}$ 的最小上界。一个正整数是集合 $\{1, 2, 4, 5, 10\}$ 的上界, 当且仅当它被 1, 2, 4, 5 和 10 整除。具有这种性质的整数就是被这些整数的最小公倍数 20 整除的整数。因此, 20 是 $\{1, 2, 4, 5, 10\}$ 的最小上界。■

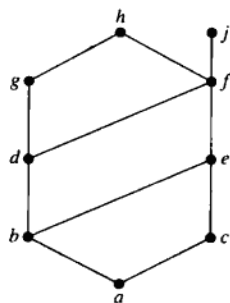


图 5-20 偏序集的哈塞图

5.6.5 格

如果一个偏序集的每对元素都有最小上界和最大下界, 就称这个偏序集为格。格有许多特殊的性质。此外, 格有许多不同的应用, 如用在信息流的模型。格在布尔代数中也起到了重要的作用。

例 21 确定图 5-21 的每个哈塞图表示的偏序集是否是格。

解 在图 5-21a 和图 5-21c 中的哈塞图表示的偏序集是格, 因为在每个偏序集中每对元素都有最小上界和最大下界, 读者应该能验证这一点。另一方面, 图 5-21b 所示的哈斯图的偏序集不是格, 因为元素 b 和 c 没有最小上界。为此只要注意到 d, e 和 f 中每一个都是上界, 但这 3 个元素的任何一个关于这个偏序集中的序都不大于其他 2 个。■

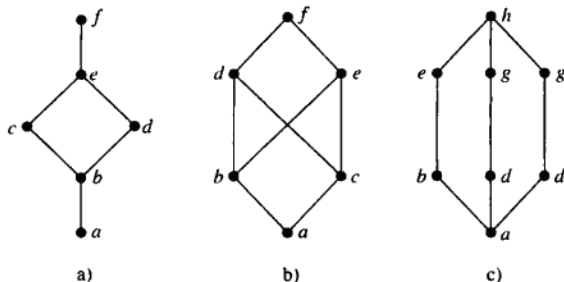


图 5-21 三个偏序集的哈塞图

例 22 偏序集 $(\mathbb{Z}^+, |)$ 是格吗?

解 设 a 和 b 是两个正整数。这两个整数的最小上界和最大下界分别是它们的最小公倍数和最大公约数, 读者应能验证这一点。因此这个偏序集是格。 ■

例 23 确定偏序集 $(\{1, 2, 3, 4, 5\}, |)$ 和 $(\{1, 2, 4, 8, 16\}, |)$ 是否为格。

解 因为 2 和 3 在 $(\{1, 2, 3, 4, 5\}, |)$ 中没有上界, 它们当然没有最小上界。因此第一个偏序集不是格。

第二个偏序集的每两个元素都有最小上界和最大下界。在这个偏序集中两个元素的最小上界是它们中间较大的元素, 而两个元素的最大下界是它们中间较小的元素。读者应能验证这一点。因此第二个偏序集是格。 ■

例 24 确定 $(P(S), \subseteq)$ 是否是格, 其中 S 是集合。

解 设 A 和 B 是 S 的两个子集。 A 和 B 的最小上界和最大下界分别是 $A \cup B$ 和 $A \cap B$, 读者可以证明这一点。因此 $(P(S), \subseteq)$ 是格。 ■

例 25 信息流的格模型 在许多设置中, 从一个人或计算机程序到另一个人或计算机程序的信息流要受到限制, 这可以通过安全权限来实现。我们可以使用格的模型来表示不同的信息流策略。例如, 一个通用的信息流策略是用于政府或军事系统中的多级安全策略。为每组信息分配一个安全级别, 并且每个安全级别用一个对 (A, C) 表示, 其中 A 是权限级别, C 是种类。然后允许人和计算机程序从一个被特别限制的安全类的集合中访问信息。

在美国政府中, 使用的典型的权限级别是不保密(0)、秘密(1)、机密(2)和绝密(3)。在安全级别中使用的种类是一个集合的子集, 这个集合含有与一个特定行业领域相关的所有的分部, 每个分部表示一个指定的对象域。例如, 如果分部的集合是{间谍, 鼯鼠, 双重间谍}, 那么存在 8 个不同的种类, 分部集合有 8 个子集, 对于每个子集有一类, 例如{间谍, 鼯鼠}。

我们可以对安全类排序, 规定 $(A_1, C_1) \leq (A_2, C_2)$ 当且仅当 $A_1 \leq A_2$ 和 $C_1 \subseteq C_2$ 。信息允许从安全类 (A_1, C_1) 流向安全类 (A_2, C_2) 当且仅当 $(A_1, C_1) \leq (A_2, C_2)$ 。例如, 信息允许从安全类(机密, {间谍, 鼯鼠})流向安全类(绝密, {间谍, 鼯鼠, 双重间谍}), 反之, 信息不允许从安全类(绝密, {间谍, 鼯鼠})流向安全类(机密, {间谍, 鼯鼠, 双重间谍})或(绝密, {间谍})。 ■

5.6.6 拓扑排序

假设一个项目由 20 个任务构成。某些任务只能在其他任务结束之后完成。如何找到关于这些任务的顺序? 为了对这个问题构造模型, 我们建立一个任务集合上的偏序, 使得 $a < b$, 当且仅当 a 和 b 是任务且直到 a 结束后 b 才能开始。为安排好这个项目, 需要得出与这个偏序相容的所有 20 个任务的顺序。我们将说明怎样做到这一点。

我们从定义开始。如果只要 aRb 就有 $a \leq b$, 则称一个全序 $\leq$ 与偏序 R 是相容的。从一个偏序构造一个相容的全序叫做拓扑排序。我们需要使用引理 1。

引理 1 每个有穷非空偏序集 $(S, \leq)$ 都有极小元素。

证 选择 S 的一个元素 a_0 。如果 a_0 不是极小元素, 那么存在元素 a_1 , 满足 $a_1 < a_0$ 。如果 a_1 不

是极小元素，那么存在元素 a_2 ，满足 $a_2 < a_1$ 。继续这一过程，使得如果 a_n 不是极小元素，那么存在元素 a_{n+1} 满足 $a_{n+1} < a_n$ 。因为在这个偏序集只有有穷个元素，这个过程一定结束并且具有极小元素 a_n 。□

我们将要描述的拓扑排序算法对任何有穷非空偏序集都有效。为在偏序集 $(A, \leq)$ 上定义一个全序，首先选择一个极小元素 a_1 ；由引理 1 这样的元素存在。接着，正如读者应该验证的， $(A - \{a_1\}, \leq)$ 也是一个偏序集。如果它是非空的，选择这个偏序集的一个极小元素 a_2 。然后再取走 a_2 ，如果还有其他的元素留下来，在 $A - \{a_1, a_2\}$ 中选择一个极小元素 a_3 。继续这个过程，只要还有元素留下来，就在 $A - \{a_1, a_2, \dots, a_k\}$ 中选择极小元素 a_{k+1} 。

因为 A 是有穷集，这个过程一定终止。最终产生一个元素序列 $a_1, a_2, \dots, a_n$ 。所需要的全序定义为

$$a_1 < a_2 < \dots < a_n$$

这个全序是与初始的偏序相容的。为看出这一点，注意如果在初始的偏序中 $b < c$ ， c 在算法的某个阶段被选择为极小元素，则这时 b 已经被移出，否则 c 就不会是极小元素。关于这个拓扑排序算法的伪码在算法 1 给出。

算法 1 拓扑排序

```
procedure topological sort( $S$ : 有穷偏序集)
 $k := 1$ 
while  $S \neq \emptyset$ 
begin
     $a_k := S$  的极小元素 {由引理 1 这样的元素一定存在}
     $S := S - \{a_k\}$ 
     $k := k + 1$ 
end { $a_1, a_2, \dots, a_n$  是  $S$  的相容的偏序}
```

例 26 找出对于偏序集 $(\{1, 2, 4, 5, 12, 20\}, |)$ 的一个相容的全序。

解 第一步是选择一个极小元素。这个元素一定是 1，因为它是唯一的极小元素。下一步选择 $(\{2, 4, 5, 12, 20\}, |)$ 的一个极小元素。在这个偏序集中有两个极小元素，即 2 和 5。我们选择 5。剩下的元素是 $\{2, 4, 12, 20\}$ 。在这一步，唯一极小元素是 2。下一步选择 4，因为它是 $(\{4, 12, 20\}, |)$ 的唯一极小元素。因为 12 和 20 都是 $(\{12, 20\}, |)$ 的极小元素，下一步选哪一个都可以。我们选 20，只剩下 12 作为最后的元素。这产生了全序

$$1 < 5 < 2 < 4 < 20 < 12$$

这个排序算法所使用的步骤在图 5-22 中给出。

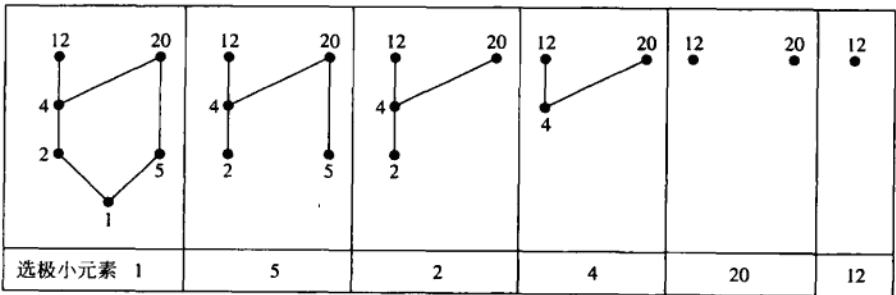


图 5-22 $(\{1, 2, 4, 5, 12, 20\}, |)$ 的拓扑排序

在项目的安排中常会用到拓扑排序。

例 27 一个计算机公司的开发项目需要完成 7 个任务。其中某些任务只能在其他任务结束后才能开始。考虑如下建立任务上的偏序，如果任务 Y 在 X 结束后才能开始，则任务 $X < \text{任务 } Y$ 。这 7 个任务关于这个偏序的哈塞图在图 5-23 中给出。求一个全序，使得可以按照这个全序执行这些任务以完成这个项目。

解 可以通过执行一个拓扑排序得到 7 个任务的排序。排序的步骤显示在图 5-24 中。这个排序的结果， $A < C < B < E < F < D < G$ ，给出了关于任务的一种可能的次序。

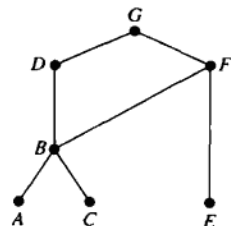


图 5-23 关于 7 个任务的哈塞图

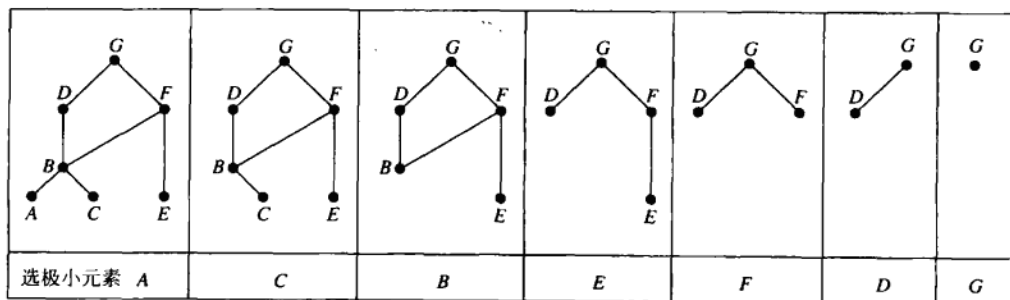


图 5-24 任务的拓扑排序

练习

1. 以下这些集合 $\{0, 1, 2, 3\}$ 上的关系，哪些是偏序？如果不是偏序，请问它缺少偏序的哪些性质。

- $\{(0, 0), (1, 1), (2, 2), (3, 3)\}$
- $\{(0, 0), (1, 1), (2, 0), (2, 2), (2, 3), (3, 2), (3, 3)\}$
- $\{(0, 0), (1, 1), (1, 2), (2, 2), (3, 3)\}$
- $\{(0, 0), (1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3)\}$
- $\{(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 2), (3, 3)\}$

2. 设是 a 和 b 人， S 是全世界所有人构成的集合， $(a, b) \in R$ 。请问 (S, R) 是否为偏序，如果

- a 比 b 个子高？
- a 不比 b 高？
- $a=b$ 或 a 是 b 的祖先？
- a 和 b 有共同的朋友？

3. 下面哪些集合是偏序集？

- $(\mathbf{Z}, =)$
- $(\mathbf{Z}, \neq)$
- $(\mathbf{Z}, \geq)$
- $(\mathbf{Z}, \neq)$

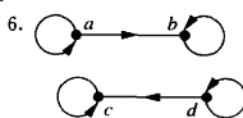
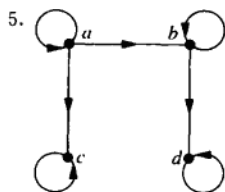
4. 确定以下 0-1 矩阵表示的关系是否为偏序。

a) $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

b) $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$

在练习 5~6 中确定有向图所表示的关系是否为偏序。



7. 求下面偏序集的对偶。

a) $(\{0, 1, 2\}, \leq)$

b) $(\mathbb{Z}, \geq)$

c) $(P(\mathbb{Z}), \supseteq)$

d) $(\mathbb{Z}^+, |)$

8. 在下面偏序集中找出两个不可比的元素。

a) $(P(\{0, 1, 2\}), \subseteq)$

b) $(\{1, 2, 4, 6, 8\}, |)$

9. 找出下面的 n 元组的字典顺序。

a) $(1, 1, 2), (1, 2, 1)$

b) $(0, 1, 2, 3), (0, 1, 3, 2)$

c) $(1, 0, 1, 0, 1), (0, 1, 1, 1, 0)$

10. 找出二进制串 $0, 01, 11, 001, 010, 011, 0001$ 和 0101 的基于 $0 < 1$ 的字典顺序。

11. 画出 $\{0, 2, 5, 10, 11, 15\}$ 上的“小于或等于”关系的哈塞图。

12. 画出下述集合上整除关系的哈塞图。

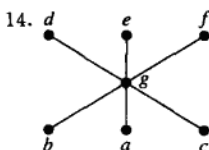
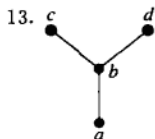
a) $\{1, 2, 3, 4, 5, 6, 7, 8\}$

b) $\{1, 2, 3, 5, 7, 11, 13\}$

c) $\{1, 2, 3, 6, 12, 24, 36, 48\}$

d) $\{1, 2, 4, 8, 16, 32, 64\}$

列出练习 13~14 所示哈塞图的偏序中的所有有序对。



设 $(S, \leq)$ 是偏序集。我们说一个元素 $y \in S$ 覆盖元素 $x \in S$, 当 $x < y$ 并且没有元素 $z \in S$ 使得 $x < z < y$ 。使得 y 覆盖 x 的对 (x, y) 的集合叫做 $(S, \leq)$ 的覆盖关系。

15. S 的幂集上的偏序 $(A, B) \mid A \subseteq B$ 的覆盖关系是什么? 其中 $S = \{a, b, c\}$ 。

16. 证明: 一个有穷偏序集可以从它的覆盖关系重新构造出来。[提示: 证明偏序集是它的覆盖关系的自反传递闭包。]

17. 对偏序集 $(\{3, 5, 9, 15, 24, 45\}, |)$, 回答下述问题。

a) 求极大元素。

b) 求极小元素。

c) 存在最大元素吗?

d) 存在最小元素吗?

e) 找出 $\{3, 5\}$ 的所有上界。

f) 如果存在的话, 求 $\{3, 5\}$ 的最小上界。

g) 求 $\{15, 45\}$ 的所有下界。

h) 如果存在的话, 求 $\{15, 45\}$ 的最大下界。

18. 对偏序集 $(\{\{1\}, \{2\}, \{4\}, \{1, 2\}, \{1, 4\}, \{2, 4\}, \{3, 4\}, \{1, 3, 4\}, \{2, 3, 4\}\}, \subseteq)$, 回答下述问题。

a) 求极大元素。

b) 求极小元素。

c) 存在最大元素吗?

d) 存在最小元素吗?

e) 求 $\{\{2\}, \{4\}\}$ 的所有上界。

f) 如果存在的话, 求 $\{\{2\}, \{4\}\}$ 的最小上界。

g) 求 $\{\{1, 3, 4\}, \{2, 3, 4\}\}$ 的所有下界。

h) 如果存在的话, 求 $\{\{1, 3, 4\}, \{2, 3, 4\}\}$ 的最大下界。

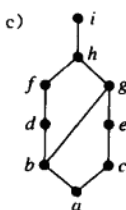
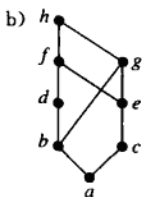
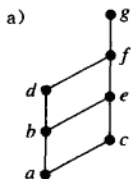
19. 证明: 字典顺序是两个偏序集的笛卡儿积上的偏序。

20. 假设 $(S, \leq_1)$ 和 $(T, \leq_2)$ 是偏序集。证明 $(S \times T, \leq)$ 也是偏序集, 其中 $(s, t) \leq (u, v)$, 当且仅当 $s \leq_1 u$ 且 $t \leq_2 v$ 。

21. a) 证明: 在一个具有最大元素的偏序集中恰好存在一个极大元素。

b) 证明: 在一个具有最小元素的偏序集中恰好存在一个极小元素。

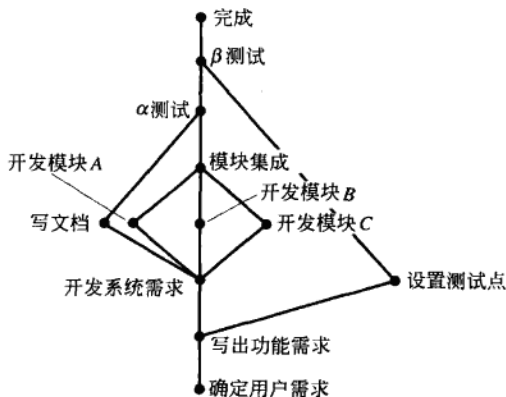
22. 确定具有下面哈塞图的偏序集是否为格。



23. 证明：一个格的每个有限非空子集有最小上界和最大下界。
24. 在一个公司里用信息流的格模型控制敏感信息，这些信息具有由有序对 (A, C) 表示的安全类别。这里 A 是权限级别，这种权限级别可以是非私有的(0)、私有的(1)、受限制的(2)或注册的(3)。种类 C 是所有项目集合{猎豹，黑斑羚，美洲狮}的子集(在公司里常常使用动物的名字作为项目的代码名字)。
- 信息是否允许从(私有的，{猎豹，美洲狮})流向(受限制的，{美洲狮})？
 - 信息是否允许从(受限制的，{猎豹})流向(注册的，{猎豹，黑斑羚})？
 - 允许信息从(私有的，{猎豹，美洲狮})流向哪个类？
 - 允许信息从哪个类流向安全类(受限制的，{黑斑羚，美洲狮})？
- *25. 证明：一个集合上的所有划分的集合与关系 S 构成一个格，其中如果划分 P_1 是划分 P_2 的加细，则 $P_1 \leq P_2$ (见 5.5 节的练习 22 前面的说明)。
26. 证明：每个有限格都有一个最小元素和一个最大元素。
27. 验证 $(\mathbb{Z}^+ \times \mathbb{Z}^+, \leq)$ 是一个良序集，其中 $\leq$ 是如例 8 所声明的字典顺序。

如果偏序集中不存在元素的无限递减序列，即元素 $x_1, x_2, \dots, x_n \dots$ 使得 $\dots < x_n < \dots < x_2 < x_1$ ，则这个偏序集是良基的。设偏序集 $(R, \leq)$ ，如果对所有的 $x \in S$ 和 $y \in S(x < y)$ 都存在元素 $z \in R$ 使得 $x < z < y$ ，则称这个偏序集是稠密的。

28. 证明：偏序集 $(\mathbb{Z}, \leq)$ (其中 $x < y$ 当且仅当 $|x| < |y|$) 是良基的，但不是全序集。
29. 证明：有理数和通常的“小于或等于”关系构成的偏序集 $(\mathbb{Q}, \leq)$ 是稠密偏序集。
30. 证明：偏序集是良序的，当且仅当它是全序的并且是良基的。
31. 求一个与例 27 构造的序不同的完成开发项目中任务的序。
32. 如果关于一个软件项目的任务的哈塞图如下所示，对这个软件项目的任务进行排序。



关键术语和结果

术语

从 A 到 B 的二元关系： $A \times B$ 的子集。

A 的关系：从 A 到自身的二元关系(即 $A \times A$ 的子集)。

$S \circ R$ ： R 与 S 的合成。

R^{-1} ： R 的逆关系。

R^n ： R 的 n 次幂。

自反的： A 上的一个关系是自反的，如果对所有的 $a \in A$ 有 $(a, a) \in R$ 。

对称的： A 上的一个关系是对称的，如果只要 $(a, b) \in R$ 就有 $(b, a) \in R$ 。

反对称的： A 上的关系是反对称的，如果只要 $(a, b) \in R$ 和 $(b, a) \in R$ 就有 $a = b$ 。

传递的： A 上的关系 R 是传递的，如果从 $(a, b) \in R$ 和 $(b, c) \in R$ 推出 $(a, c) \in R$ 。

$A_1, A_2, \dots, A_n$ 的 n 元关系： $A_1 \times A_2 \times \dots \times A_n$ 的子集。

关系数据模型：一个使用 n 元关系表示数据库的模型。

主关键字： n 元关系的一个域，使得一个 n 元组被它在这个域的值唯一确定。

复合关键字：一个 n 元关系域的笛卡儿积，使得一个 n 元组被它在这些域的值唯一确定。

选择算子：一个函数，它在 n 元关系中选出满足特定条件的 n 元组。

投影：一个函数，它从一个 n 元关系通过删除域产生一个阶较小的关系。

连接：一个函数，它把具有某些相同的域的 n 元关系组合起来。

有向图：称为顶点的元素以及这些元素的有序对(也叫做边)的集合。

环：形如 (a, a) 的边。

关系 R 关于性质 P 的闭包：包含 R 的关系 S (如果存在的话)具有性质 P ，并且被任何包含 R 且具有性质 P 的关系所包含。

有向图中的路径：边的序列 $(a, x_1), (x_1, x_2), \dots, (x_{n-2}, x_{n-1}), (x_{n-1}, b)$ 使得序列中每条边的终点是后面一条边的起点。

有向图的回路(或圈)：以同一顶点作为起点和终点的路径。

R^* (连通性关系)：由有序对 (a, b) 构成的关系，条件是存在一条从 a 到 b 的路径。

等价关系：自反的、对称的和传递的关系。

等价：如果 R 是等价关系，若 aRb ，那么 a 等价于 b 。

$[a]_R$ (a 关于 R 的等价类)： A 中所有等价于 a 的元素的集合。

$[a]_m$ (模 m 的同余类)：与 a 模 m 同余的整数的集合。

集合 S 的划分：一族两两不相交的非空子集，且这些子集的并就是 S 。

偏序：自反的、反对称的和传递的关系。

偏序集 (S, R) ：集合 S 与这个集合上的偏序 R 。

可比的：偏序集 $(A, \leq)$ 的元素 a 和 b 是可比的如果 $a \leq b$ 或 $b \leq a$ 。

不可比的：一个偏序集的元素不是可比的。

全序(或线序)：一个偏序，并且它的每对元素都是可比的。

全序(或线序)集：具有一个全序(或线序)的偏序集。

良序集：偏序 $(S, \leq)$ ，其中 $\leq$ 是全序， S 的每个非空子集至少有一个元素。

字典顺序：笛卡儿积或串的一个偏序(参见 5.6.2 节)。

哈塞图：偏序集的图表示，其中所有的环和由传递性导出的边不出现，并且顶点的位置指示了边的方向。

极大元素：偏序集的一个元素，它不小于这个偏序集的所有其他元素。

极小元素：偏序集的一个元素，它不大于这个偏序集的所有其他元素。

最大元素：偏序集的一个元素，它大于或等于这个集合的所有其他元素。

最小元素：偏序集的一个元素，它小于或等于这个集合的所有其他元素。

集合的上界：偏序集的一个元素，它大于这个集合的所有其他元素。

集合的下界：偏序集的一个元素，它小于这个集合的所有其他元素。

集合的最小上界：集合的一个上界，它小于所有其他上界。

集合的最大下界：集合的一个下界，它大于所有其他下界。

格：一个偏序集，其中每对元素都有一个最大下界和一个最小上界。

与一个偏序相容的全序：包含了给定偏序的一个全序。

拓扑排序：用给定偏序构造一个相容的全序。

结果

- 集合 A 的关系 R 的自反闭包等于 $R \cup \Delta$ ，其中 $\Delta = \{(a, a) \mid a \in A\}$ 。
- 集合 A 的关系 R 的对称闭包等于 $R \cup R^{-1}$ ，其中 $R^{-1} = \{(b, a) \mid (a, b) \in R\}$ 。
- 一个关系的传递闭包等于从这个关系构成的连通性关系。

- 为了求一个关系的传递闭包的沃舍尔算法(见 5.4.5 节)。
- 设 R 是等价关系, 那么下面三条语句是等价的:
(1) aRb (2) $[a]_R \cap [b]_R \neq \emptyset$ (3) $[a]_R = [b]_R$
- 集合 A 的等价关系的等价类构成 A 的划分。相反, 从一个划分可以构造一个等价关系使得等价类就是划分中的子集。
- 良序归纳原理。
- 拓扑排序算法(见 5.6.5 节)。

复习题

- a) 什么是集合上的关系? b) 一个 n 元素集合上有多少个关系?
- a) 什么是自反关系? b) 什么是对称关系?
c) 什么是反对称关系? d) 什么是传递关系?
- 给出一个集合 $\{1, 2, 3, 4\}$ 上的关系的例子, 使得它是
a) 自反的、对称的但不是传递的。 b) 不是自反的, 是对称的和传递的。
c) 自反的、反对称的, 但不是传递的。 d) 自反的、对称的和传递的。
e) 自反的、反对称的和传递的。
- a) 在一个 n 元素集合上有多少个自反的关系?
b) 在一个 n 元素集合上有多少个对称的关系?
c) 在一个 n 元素集合上有多少个反对称的关系?
- a) 解释在一个大学里怎样用一个 n 元关系表示有关学生的信息。
b) 怎样用一个包含学生姓名、地址、电话号码、专业和平均学分绩点的 5 元关系构造包含学生姓名、专业和平均学分绩点的 3 元关系。
c) 怎样将包含学生姓名、地址、电话号码和专业的 4 元关系和包含学生姓名、学号、专业和学分数的 4 元关系组合成一个单一的 n 元关系?
- a) 解释怎样使用一个 0-1 矩阵表示有穷集上的关系。
b) 解释怎样使用表示关系的 0-1 矩阵来确定这个关系是否为自反的、对称的和反对称的。
- a) 解释怎样使用一个有向图来表示有穷集上的关系。
b) 解释怎样使用表示关系的有向图来确定这个关系是否为自反的、对称的和反对称的。
- a) 定义一个关系的自反闭包和对称闭包。
b) 可以怎样构造一个关系的自反闭包?
c) 可以怎样构造一个关系的对称闭包?
d) 求集合 $\{1, 2, 3, 4\}$ 上的关系 $\{(1, 2), (2, 3), (2, 4), (3, 1)\}$ 的自反闭包和对称闭包。
- a) 定义一个关系的传递闭包。
b) 一个关系的传递闭包能够通过包含所有的对 $\langle a, c \rangle$ 得到吗? 其中 $\langle a, c \rangle$ 满足 $\langle a, b \rangle$ 和 $\langle b, c \rangle$ 属于 R 。
c) 描述求关系的传递闭包的两个算法。
d) 求关系 $\{(1, 1), (1, 3), (2, 1), (2, 3), (2, 4), (3, 2), (3, 4), (4, 1)\}$ 的传递闭包。
- a) 定义等价关系。
b) 集合 $\{a, b, c, d\}$ 上的哪些关系是包含了 $\langle a, b \rangle$ 和 $\langle b, d \rangle$ 的等价关系?
- a) 证明模 m 同余关系是等价关系, 其中 m 是正整数。
b) 证明关系 $\{\langle a, b \rangle \mid a \equiv \pm b \pmod{7}\}$ 是整数集上的等价关系。
- a) 什么是一个等价关系的等价类?
b) 什么是模 5 同余关系的等价类?
c) 什么是问题 11(b) 中等价关系的等价类?
- 解释在集合的等价关系与集合划分之间的联系。
- a) 定义偏序。 b) 证明正整数集合上的整除关系是偏序。
- 解释怎样用集合 A_1 和 A_2 上的偏序定义在集合 $A_1 \times A_2$ 上的偏序。
- a) 解释怎样构造有穷集上的偏序的哈塞图。

- b) 画出集合 $\{2, 3, 5, 9, 12, 15, 18\}$ 上的整除关系的哈塞图。
17. a) 定义一个偏序集的极大元素和最大元素。
 b) 给出一个有 3 个极大元素的偏序集的例子。
 c) 给出一个有 1 个最大元素的偏序集的例子。
18. a) 定义格。
 b) 给出一个 5 元偏序集是格的例子和一个 5 元偏序集不是格的例子。
19. a) 证明一个格的每个有穷子集有一个最大下界和一个最小上界。
 b) 证明每个具有有限元素的格有一个最小元素和一个最大元素。
20. a) 定义一个良序集。
 b) 描述一个从偏序集产生良序集的算法。
 c) 如果仅当某个或某些其他任务完成以后, 一个任务才可以开始, 解释怎样用 (b) 中的算法排序这个项目中的任务。

补充练习

1. 设 S 是所有英语字母的串的集合。确定下面的关系是否是自反的、反自反的、对称的、反对称的和传递的。
 a) $R_1 = \{(a, b) \mid a \text{ 和 } b \text{ 没有公共字母}\}$
 b) $R_2 = \{(a, b) \mid a \text{ 和 } b \text{ 长度不相等}\}$
 c) $R_3 = \{(a, b) \mid a \text{ 比 } b \text{ 长}\}$
2. 构造集合 $\{a, b, c, d\}$ 上的关系, 使得它是
 a) 自反的、对称的, 但不是传递的。
 b) 反自反的、对称的和传递的。
 c) 反自反的、反对称的, 但不是传递的。
 d) 自反的, 既不是对称的也不是反对称的, 是传递的。
 e) 既不是自反的、反自反的、对称的和反对称的, 也不是传递的。
3. $\mathbb{Z} \times \mathbb{Z}$ 上的关系 R 定义如下: $(a, b)R(c, d)$, 当且仅当 $a+d=b+c$ 。证明: R 是等价关系。
4. 证明: 一个反对称关系的子集也是一个反对称关系。
5. 设 R 是 A 上的自反关系, 证明: $R \subseteq R^2$ 。
6. 假设 R_1 和 R_2 是集合 A 上的自反关系, 证明: $R_1 \oplus R_2$ 是反自反的。
7. 假设 R_1 和 R_2 是集合 A 上的自反关系, $R_1 \cap R_2$ 也是自反的吗? $R_1 \cup R_2$ 也是自反的吗?
8. 假设 R 是集合 A 上的对称关系, $\bar{R}$ 也是对称的吗?
9. 设 R_1 和 R_2 是集合 A 上的对称关系, $R_1 \cap R_2$ 也是对称的吗? $R_1 \cup R_2$ 也是对称的吗?
10. 一个关系 R 叫做循环的, 如果 aRb 和 bRc 推出 cRa 。证明 R 是自反的和循环的, 当且仅当它是等价关系。
11. 证明: 一个 n 元关系的主关键字也是这个关系的任何投影的主关键字, 其中这个投影包含这个关键字作为它的一个域。
12. 一个 n 元关系的主关键字也就是由取这个关系与第二个关系的连接而得到的较大的关系的主关键字吗?
13. 证明: 一个关系的对称闭包的自反闭包和它的自反闭包的对称闭包是相同的。
14. 设 R 是所有数学家的集合上的关系, R 包含有序对 (a, b) , 当且仅当 a 与 b 合写了一篇文章。
 a) 描述关系 R^2 。
 b) 描述关系 R^* 。
15. a) 如果一个数学家与多产的匈牙利数学家保罗·埃德斯合写了一篇文章, 那么这个数学家的埃德斯数是 1。如果这个数学家没有与埃德斯合写一篇文章, 但是与某个与埃德斯合写过论文的人合写了一篇文章, 那么这个数学家的埃德斯数是 2, 依此下去 (埃德斯本人的埃德斯数是 0)。用 R 中路径的概念给出埃德斯数的定义。
16. a) 给出一个例子, 证明一个关系的对称闭包的传递闭包不一定与这个关系的传递闭包的对称闭包相等。
 b) 证明: 一个关系的对称闭包的传递闭包一定包含这个关系的传递闭包的对称闭包。
17. a) 设 S 是一个计算机问题的子程序的集合。定义关系 R , 如果在执行中子程序 P 调用子程序 Q , 那么 $P R Q$ 。描述 R 的传递闭包。
 b) 对于哪些子程序 P , (P, P) 属于 R 的传递闭包?

c)描述 R 的传递闭包的自反闭包。

17. 假设 R 和 S 是集合 A 上的关系, $R \subseteq S$, 且 R 和 S 的关于性质 P 的闭包存在。证明 R 关于 P 的闭包是 S 关于 P 的闭包的子集。
18. 证明: 两个关系的并的对称闭包是它们的对称闭包的并。
- *19. 设计一个基于内点概念的算法, 求有向图中两个顶点之间的最长路径的长度, 或确定在这些顶点之间存在任意长的路径。
20. 下面的哪些关系是所有人集合上的等价关系?
 - a) $\{(x, y) \mid x \text{ 与 } y \text{ 有同样的星座}\}$
 - b) $\{(x, y) \mid x \text{ 与 } y \text{ 出生在同一年}\}$
 - c) $\{(x, y) \mid x \text{ 与 } y \text{ 曾待在同一城市}\}$
- *21. 在 5 元素集合上有多少个不同的等价关系恰有 3 个不同的等价类?
22. 证明: $\{(x, y) \mid x - y \in \mathbb{Q}\}$ 是实数集上的等价关系, 其中 $\mathbb{Q}$ 是有理数集合。 $[1]$, $[1/2]$ 和 $[\pi]$ 是什么?
23. 设 $P_1 = \{A_1, A_2, \dots, A_m\}$ 和 $P_2 = \{B_1, B_2, \dots, B_n\}$ 都是集合 S 的划分。证明: 形如 $A_i \cap B_j$ 的非空子集族是 S 的划分, 且是 P_1 和 P_2 的加细(见 5.5 节练习 22 前面的说明)。
- *24. 证明: 关系 R 的自反闭包的对称闭包的传递闭包是包含 R 的最小的等价关系。
25. 设 $\mathbf{R}(S)$ 是集合 S 上的所有关系的集合。如下定义 $\mathbf{R}(S)$ 上的关系 $\leq$, 如果 $R_1 \subseteq R_2$ 则 $R_1 \leq R_2$, 这里的 R_1 和 R_2 是 S 上的关系。证明: $(\mathbf{R}(S), \leq)$ 是偏序集。
26. 设 $\mathbf{P}(S)$ 是 S 上的所有划分的集合。定义 $\mathbf{P}(S)$ 上的关系如下: 如果 P_1 是 P_2 的加细, 则 $P_1 \leq P_2$ (见 5.5 节的练习 22)。证明: $(\mathbf{P}(S), \leq)$ 是偏序集。
27. 右下图为烹调中餐任务的哈塞图, 通过规定那些必需任务的顺序来安排烹调。

如果一个偏序集子集的每一对元素都是可比的, 就称这个子集为一条链。如果它的每一对元素都是不可比的, 就称这个子集为一条反链。

28. 找出 5.6 节练习 13~14 的哈塞图所表示的偏序集中所有的链。
29. 找出 5.6 节练习 13~14 的哈塞图所表示的偏序集中所有的反链。
30. 证明: 在一个有穷偏序集 $(S, \leq)$ 中每个极大链包含 S 的一个极小元素。(一条极大链是一条链, 但不是一条更大的链的子链。)
- **31. 证明: 一个偏序集可以被分成 k 条链, 其中 k 是在这个偏序集中一条反链的最多元素数。
- *32. 证明: 在任意一组 $mn+1$ 个人中要么存在 $m+1$ 个人的一个列表, 其中每个人(除了表的第一个人以外)都是表中前一个人的后代; 要么存在 $n+1$ 个人, 其中没有一个人是其他 n 个人中任何一个人的后代。[提示: 用练习 31.]

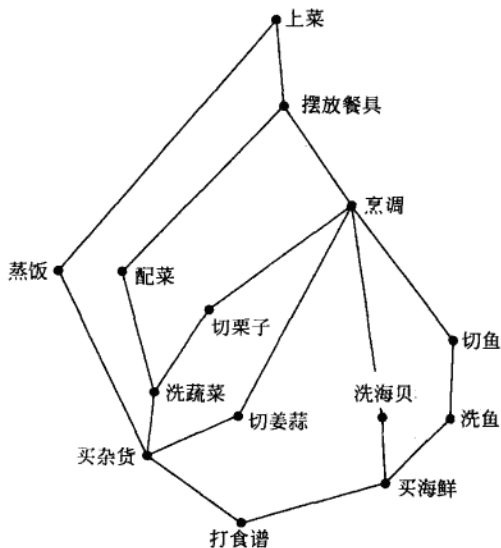
假设 $(S, \leq)$ 是良基的偏序集。良基归纳原理说,

如果 $\forall x (\forall y (y < x \rightarrow P(y)) \rightarrow P(x))$, 那么对所有的 $x \in S$, 有 $P(x)$ 为真。

33. 证明: 对于良基归纳原理不需要另外的基础情况, 就是说, 如果 $\forall x (\forall y (y < x \rightarrow P(y)) \rightarrow P(x))$, $P(u)$ 对于 S 中所有的极小元素 u 为真。
- *34. 证明: 良基归纳原理是有效的。

集合 A 上的关系 R 叫做 A 上的近似序, 如果 R 是自反的和传递的。

35. 设 R 是从 $\mathbb{Z}^+$ 到 $\mathbb{Z}^+$ 的所有函数的集合上的关系, 使得 (f, g) 属于 R , 当且仅当 f 是 $O(g)$, 证明: R 是近似序。
36. 设 R 是 A 上的近似序, 证明: $R \cap R^{-1}$ 是等价关系。
- *37. 设 R 是近似序, 设 S 是 $R \cap R^{-1}$ 的等价类的集合上的关系, C 和 D 是 R 的等价类, (C, D) 属于 S , 当且



仅当存在 C 的元素 c , D 的元素 d , 使得 (c, d) 属于 R 。证明 S 是偏序。

设 L 是一个格。由 $x \wedge y = \text{glb}(x, y)$ 和 $x \vee y = \text{lub}(x, y)$ 定义交 ($\wedge$) 和并 ($\vee$) 运算。

38. 证明下面的性质对格 L 的一切元素 x, y, z 成立:

$$a) x \wedge y = y \wedge x, x \vee y = y \vee x \quad (\text{交换律})$$

$$b) (x \wedge y) \wedge z = x \wedge (y \wedge z), (x \vee y) \vee z = x \vee (y \vee z) \quad (\text{结合律})$$

$$c) x \wedge (x \vee y) = x, x \vee (x \wedge y) = x \quad (\text{吸收律})$$

$$d) x \wedge x = x, x \vee x = x \quad (\text{幂等律})$$

39. 证明: 如果 x 和 y 是格的元素, 那么 $x \vee y = y$ 当且仅当 $x \wedge y = x$ 。

一个格 L 是有界的, 如果它有一个上界, 记作 1 , 即对所有的 $x \in L$ 有 $x \leq 1$; 并且有一个下界, 记作 0 , 即对所有的 $x \in L$ 有 $0 \leq x$ 。

40. 证明: 如果 L 是具有上界 1 和下界 0 的有界格, 那么对所有的元素 $x \in L$, 具有下面的性质:

$$a) x \vee 1 = 1$$

$$b) x \wedge 1 = x$$

$$c) x \vee 0 = x$$

$$d) x \wedge 0 = 0$$

41. 证明: 每个有限格是有界的。

一个格叫做分配的, 如果对 L 中所有的 x, y, z , 有

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z) \text{ 和 } x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

*42. 给出一个不是分配格的例子。

43. 证明: 格 $(P(S), \subseteq)$ 是分配格, 其中 $P(S)$ 是有穷集 S 的幂集。

44. 格 $(\mathbb{Z}^+, |)$ 是分配格吗?

有界格 L 的元素 a 关于上界 1 和下界 0 的补元是元素 b , 使得 $a \vee b = 1$ 和 $a \wedge b = 0$ 。如果一个格的每个元素都有补元, 那么这个格叫做有补格。

45. 给出一个有限格的例子, 其中至少 1 个元素有多于 1 个的补元且至少 1 个元素没有补元。

46. 证明: 格 $(P(S), \subseteq)$ 是有补格, 其中 $P(S)$ 是有穷集 S 的幂集。

*47. 证明: 如果 L 是有限分配格, 那么 L 的一个元素至多有 1 个补元。

1.7 节例 12 中介绍的 Chomp 游戏可以推广到在任意最小元素为 a 的有限偏序集 $(S, \leq)$ 上玩。在这个游戏中, 一个移动由选择 S 中的元素 x , 及移动 x 和所有 S 中比 x 大的元素构成。失败者是被迫选择最小元素 a 的那个玩家。

48. 证明: 1.7 节例 12 所描述的 Chomp 游戏(曲奇饼摆放在 $m \times n$ 的矩形网格中)与偏序集 $(S, |)$ 上的 Chomp 游戏相同, 其中 S 是所有除 $p^{m-1}q^{n-1}$ 的正整数集合, 这里 p 和 q 是两个不同的素数。

49. 证明: 如果 $(S, \leq)$ 有最大元素 b , 那么在偏序集上存在一个获胜策略。[提示: 推广 1.7 节例 12 中的讨论。]

计算机题目

按下述给定的输入和输出写出程序。

1. 给定表示有穷集上关系的矩阵, 确定这个关系是否是自反或反自反的。

2. 给定表示有穷集上关系的矩阵, 确定这个关系是否是对称或反对称的。

3. 给定表示有穷集上关系的矩阵, 确定这个关系是否是传递的。

4. 给定正整数 n , 显示一个 n 元素集合上所有的关系。

*5. 给定一个正整数 n , 确定 n 元素集合上的传递关系的个数。

*6. 给定一个正整数 n , 确定 n 元素集合上的等价关系的个数。

*7. 给定一个正整数 n , 显示 n 个最小的正整数集合上的所有等价关系。

8. 给定一个 n 元关系, 当某些特定的字段被删除以后求这个关系的投影。

9. 给定一个 m 元关系、一个 n 元关系和一个公共字段的集合, 找出这些关系关于这些公共字段的连接。

10. 给定表示一个有穷集上关系的矩阵, 求表示这个关系自反闭包的矩阵。

11. 给定表示一个有穷集上关系的矩阵, 求表示这个关系对称闭包的矩阵。

12. 给定表示一个有穷集上关系的矩阵, 通过计算表示这个关系的矩阵的幂的联合求表示这个关系传递闭包的矩阵。

13. 给定表示一个有穷集上关系的矩阵, 使用沃舍尔算法求表示这个关系的传递闭包的矩阵。
14. 给定表示一个有穷集上关系的矩阵, 求表示包含这个关系的最小的等价关系的矩阵。
15. 给定一个有穷集上的偏序, 使用拓扑排序找出一个与它相容的全序。

计算和研究

使用一个计算程序或你已完成的程序做下面的练习。

1. 显示一个 4 元素集合上的所有不同的关系。
2. 显示一个 6 元素集合上的所有不同的自反和对称的关系。
3. 显示一个 5 元素集合上的所有不同的自反和传递的关系。
- *4. 对所有的正整数 n , $n \leq 7$, 确定在 n 元素集合上有多少个传递关系。
5. 使用对应于某个特定的运输或通信网络的有向链路的关系, 或者使用一个随机生成的关系, 在至少 20 个元素的集合上求所用关系的传递闭包。
6. 对于所有不超过 20 的正整数 n , 计算 n 元素集合上的不同的等价关系个数。
7. 显示 7 元素集合上的所有的等价关系。
- *8. 显示 5 元素集合上的所有的偏序。
- *9. 显示 5 元素集合上的所有的格。

写作题目

用本书以外的资料写成短文回答下列问题。

1. 讨论模糊关系的概念。怎样使用模糊关系?
2. 不限于 5.2 节所介绍的内容, 描述关系数据库的基本原理。关系数据库与其他类型的数据库相比, 使用面有多广?
3. 查找沃舍尔和罗伊(Roy)的原始论文(法文), 在那篇论文中他们提出了求传递闭包的算法。讨论他们的方法。为什么可以认为沃舍尔算法是被多人独立发现的?
4. 描述怎样用等价类把有理数定义为整数对的类, 以及遵照这种方法怎样定义有理数的基本算术运算。
5. 解释赫尔姆·哈塞怎样使用我们现在称为哈塞图的图示。
6. 描述在计算机操作系统中用来执行信息流策略的某些机制。
7. 讨论计划评审技术(Program Evaluation and Review Technique, PERT)在安排一个大的复杂项目的任务中的应用。PERT 的使用面有多广?
8. 讨论关键路径方法(Critical Path Method, CPM)对找出完成项目的最短时间的应用。CPM 的使用面有多广?
9. 讨论格中的对偶性的概念。解释怎样用对偶性建立新的结果?
10. 解释模格的意义。描述模格的某些性质, 描述模格是怎样在投影几何的研究中产生的。

图

图是由顶点和连接顶点的边构成的离散结构。几乎可以想到的每个学科中的问题都可以运用图模型来求解。我们将给出实例说明如何在各种领域中运用图来建模。例如，如何用图表示生态环境里不同物种的竞争，如何用图表示组织中谁影响谁，如何用图表示循环锦标赛的输出。我们将描述如何用图对人们之间的相识关系、研究人员之间的合作关系、电话号码间的呼叫关系以及网站之间的链接关系进行建模。我们将说明如何用图对地图和一个组织内员工的工作指派进行建模。

运用图模型，可以确定能不能遍历一个城市的所有街道而不在任一条街道上走两遍，还能找出对地图上的区域着色所需要的颜色数；可以用图来确定某一个电路是否能够在平面电路板上实现；用图区分两种有着同样的分子式但结构不同的两种化合物；我们能够运用计算机网络的图模型确定两台计算机是否由通信链路连接；对其边赋予了权重的图可以求解诸如在运输网络中寻找两个城市间的最短路径这类问题。我们还可以用图来安排考试和指定电视台的频道。本章将介绍图论的基本内容，还将给出许多不同的图模型。为了求解能够用图研究的多种问题，我们将介绍许多不同的图的算法，还将研究这些算法的复杂度。

6.1 图和图模型

首先给出图的定义。

定义 1 一个图 G 由顶点(或结点)的非空集 V 和边集 E 构成，每条边有一个或两个顶点与它相连，这样的顶点称为边的端点。边连接它的端点。

注意 图 G 的顶点集 V 可能是无限的。顶点集为无限集的图称为无限图，与之相对，顶点集为有限集的图称为有限图。在本书中，通常只考虑有限图。

现在假设一个网络由数据中心和计算机之间的通信网络组成。可以把每个数据中心的位置用一个点来表示，把每个通信链接用一段线来表示，如图 6-1 所示。

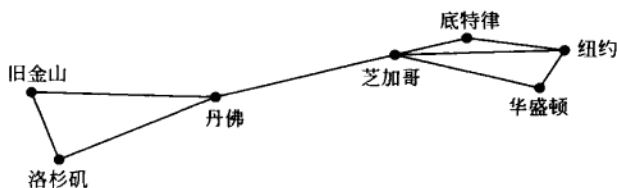


图 6-1 计算机网络

这个计算机网络可以用图来建模，图中的顶点表示数据中心，边表示通信链接。通常，用点表示顶点、用线段或者曲线表示边来可视化图。其中，表示边的线段的端点就是表示相应边的端点的点。当我们画图时，尽量不让它们的边相交。然而，并不是必须这样做，因为任意的用点表示顶点、任意形式的顶点间的连接表示边的描述方法均可使用。实际上，有些图不能够没有边相交地画在平面上(见 6.7 节)。关键一点是，只要正确地描述了顶点间的连接，画一个图的路线方式可以是任意的。

说明一下，表示这个计算机网络图的每条边都连接两个不同的顶点，即没有任何一个顶点连

接它自身，另外，也没有不同的两条边连接着相同的一对顶点。每条边都连接两个不同的顶点且没有不同的两条边连接着相同的一对顶点，这样的图称为简单图。注意，在简单图中，每条边与一对无序的顶点相关联，而且没有其他的边和这条边连接相同顶点。因此，当有一条简单图的边与 $\{u, v\}$ 相关联，也可以说 $\{u, v\}$ 是该图的一条边，这不会产生误解。

一个计算机网络可能在两个数据中心之间有多重链接，如图 6-2 所示。为这样的网络建模，需要有不只一条边连接同一对顶点的图。可能会有多重边连接同一对顶点的图称为多重图。当有 m 条不同的边与相同的无序顶点对相关联，我们也说 $\{u, v\}$ 是一条多重度为 m 的边。就是说，可以把这个边集考虑是边 $\{u, v\}$ 的 m 个不同副本。

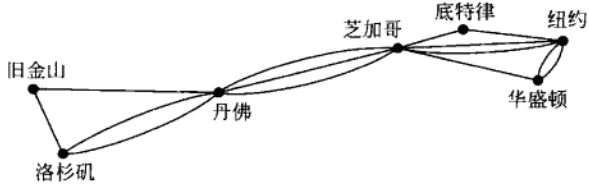


图 6-2 数据中心之间带有多重连接的计算机网络

有时候一个数据中心与它自己连接一条通信线，也许是一个为了诊断目的的反馈环。图 6-3 说明了这样的网络。为这个网络建模，需要包括把一个顶点连接到它自身的边的图。这样的边称为环。可能包含环、还有可能包含连接同一对顶点的多重边的图，称作伪图。

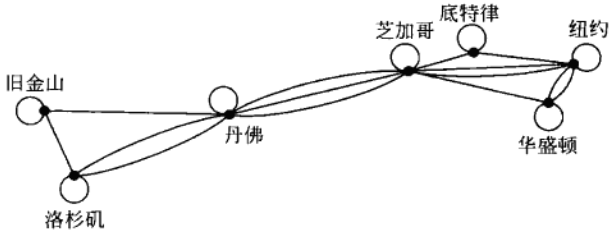


图 6-3 带诊断线的计算机网络

到目前为止，我们所介绍的图是无向图。它们的边也被认为是无向的。然而，要建立一个图模型，可能会发现有必要给这些边赋予方向。例如，在计算机网络中，一些连接可以只对一个方向操作（这种边称为单工线路）。这可能是这种情况，有大量的数据送到某些数据中心，但只有很少或者根本没有相反方向的数据传输。这样的网络如图 6-4 所示。

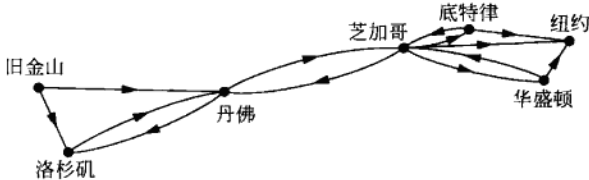


图 6-4 带有单向通信线的通信网络

我们使用有向图为这样的计算机网络建模。有向图的每条边与一个有序对相关联。这里给有向图的定义比在第 5 章使用的更加广义，在第 5 章中使用有向图来表示关系。

定义 2 一个有向图 (V, E) 由一个非空顶点集 V 和一个有向边(或弧)集 E 组成。每条有向边与一个顶点有序对相关联。认为与有序对 (u, v) 相关联的有向边开始于 u 、结束于 v 。

当画线描述一个有向图时，我们用一个从 u 指向 v 的箭头来表示这条边的方向是开始于 u 结束

于 v 。一个有向图可能包含环，也有可能包含开始和结束于相同顶点的多重有向边。有向图也有可能包含连接 u 和 v 的两个方向上的有向边；就是说，当一个有向图含有从 u 到 v 的边，它也可能包含从 v 到 u 的一条或多条边。注意，当对一个无向图的每一条边都赋予方向后，就得到了一个有向图。当一个有向图不包含环和多重有向边时，就称为简单有向图。因为简单有向图在每一对有序顶点 (u, v) 最多连一条边。如果在图中它们只连一条边，则称为 (u, v) 边。

在某些计算机网络中，两个数据中心之间可能会有多重的通信链接，正如图 6-5 所示的那样。用可以包含从一个顶点指向第二个(也许是同一个)顶点的多重有向边的有向图来对这样的网络建模，我们称这样的图为有向多重图。当 m 条有向边中的每一条都与一对有序顶点对 (u, v) 相关联，我们就说 (u, v) 是一条多重度为 m 的边。

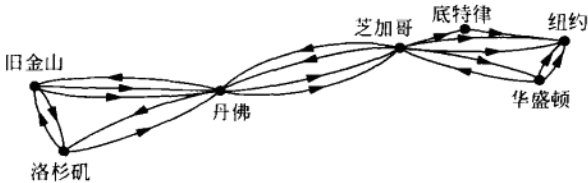


图 6-5 带多重单向线的计算机网络

对于某些模型，我们可能需要这样的图，其中有些边是无向的，而另一些边是有向的。有向边和无向边都有的图称为混合图。例如，可能会用一个混合图来给这样的计算机网络建模，该网络包含的链接有些可以双向运行，另外一些只能单向运行。

表 6-1 总结了各种图的专门术语。有时我们将用“图”作为一个通用的术语指代有向或无向的(或两者皆有)、有环或无环的，以及有多重边或无多重边的图。在其他情形下，当上下文清楚时，我们使用术语“图”只表示无向图。

表 6-1 图术语

类 型	边	允许多重边	允许环
简单图	无向	否	否
多重图	无向	是	否
伪图	无向	是	是
有向图	有向	否	是
有向多重图	有向	是	是
混合图	有向的和无向的	是	是

由于现代对图论的研究兴趣，还因为其在各个学科的广泛应用，图论中引入了许多不同的术语。不管什么时候遇到这些术语，读者应该注意它们的实际含义。数学家们用以描述图的术语已经逐步得到规范，但是讨论运用于其他学科的图的术语仍然多种多样。

尽管描述图的术语可能区别很大，但是以下三个问题能够帮助我们理解图的结构：

- 图的边是有向的还是无向的(还是两者皆有)?
- 如果是无向图，是否存在连接相同顶点对的多重边? 如果是有向图，是否存在多重有向边?
- 是否存在环?

回答这些问题有助于我们理解图。而记住所使用的特定术语就不那么重要。

图模型

图可用在各种模型里。这里将介绍几种领域的图模型。本章的后续小节和后面几章将介绍其他模型。当建立一个图模型，需要确认已经正确回答了我们提出的关于图结构的三个关键问题。

例1 生态学里栖息地重叠图 图可用在涉及不同种类动物在一起活动的许多模型里。例如用栖息地重叠图来为生态系统里物种之间的竞争建模。用顶点表示每个物种。若两个物种竞争(即它们共享某些食物来源),则用无向边连接表示它们的顶点。栖息地重叠图是一个简单图,因为在此模型中不需要环和多重边。图6-6里的图表示森林生态系统。从这个图看出松鼠与浣熊竞争,但是乌鸦不与鼯鼠竞争。

例2 熟人关系图 可用图模型来表示人与人之间的各种关系。例如可用图来表示两个人是否互相认识,即他们是否是熟人。用顶点表示具体人群里的每个人。当两个人互相认识时,用无向边连接这两个人。不使用多重边,通常也不使用环。(如果我们想表达“自己认识自己”这层意思,就在图中包含环。)图6-7显示了一个小型熟人关系图。世界上所有人的熟人关系图有超过60亿个顶点和可能超过1万亿条边!在6.4节里将要进一步讨论这个图。

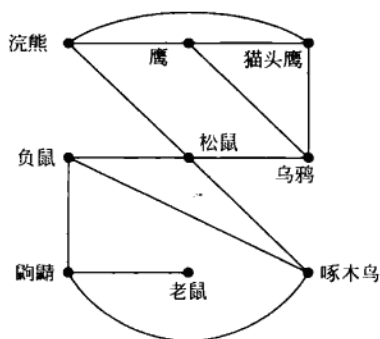


图 6-6 栖息地重叠图

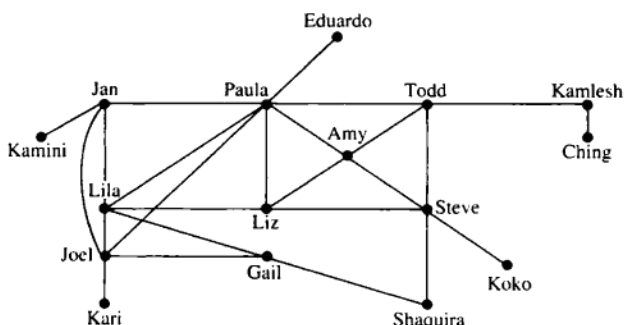


图 6-7 熟人关系图

例3 影响图 在对群体行为的研究里,可以观察到某些人能够影响其他人的思维。一种称为影响图的有向图可以用来为这样的行为建模。用顶点表示群体的每个人。当顶点 a 所表示的人影响到顶点 b 所表示的人时,就有从顶点 a 到顶点 b 的有向边。图6-8表示了群体成员的影响图的例子。在这个用影响图建模的群体里,Deborah影响Brain、Fred以及Linda,但是没有人影响Deborah。另外,Yvonne与Brain互相影响。

例4 好莱坞图 好莱坞图用顶点表示演员,并且当两个顶点所表示的演员共同出演一部电影时,就连接这两个顶点。这个图是简单图,因为它的边是无向边,不包含多重边和环。根据互联网电影数据库,在2006年1月,好莱坞图有637099个顶点和超过2000万条边,这些顶点所表示的演员出现在339896部电影中。稍后在6.4节里将要讨论好莱坞图。

例5 循环赛 每个队都与其他每队恰好比赛一次的联赛称为循环赛。用其中顶点表示每个队的有向图来为这样的比赛建模。注意若 a 队击败 b 队,则 (a, b) 是边。该图是简单有向图,不包含环和多重有向边(因为没有任何两支队的比赛多于一次)。图6-9表示这样的有向图模型。注意在这个比赛里,队1无败绩而队3无胜绩。

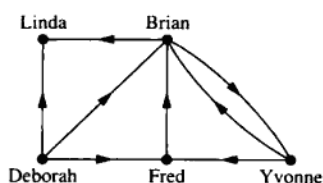


图 6-8 影响图

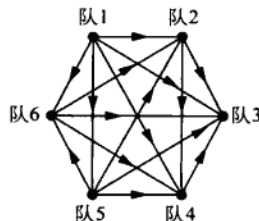


图 6-9 循环赛的图模型



例 6 合作图 合作图可用来为学术论文的合作者关系建模。在合作图中，顶点表示人（可能限制为某个学术圈子的成员），如果两个人合写论文，则用边连接这两个人。已经发现在数学研究论文上合作者的合作图有超过 400 000 个顶点和 675 000 条边。在 6.4 节里将对这个图补充说明。



例 7 呼叫图 可用来为网络(比如长途电话网)里的电话呼叫建模。具体地说，有向图用来为呼叫建模，其中用顶点表示每个电话号码，用有向边表示每次电话呼叫。表示呼叫的边是以发出呼叫的电话号码为起点，以接受呼叫的电话号码为终点。我们需要有向边，因为其表示的通话方向是有意义的。我们需要多重有向边，因为需要表示每一个从特定号码拨到第二个号码的通话。

一个小型的电话呼叫图如图 6-10a 所示，它表示 7 个电话号码。例如，这个图表示从 732-555-1234 到 732-555-9876 有 3 次呼叫和 2 次反向呼叫，但是从 732-555-4444 到其余 6 个号码(732-555-0011 除外)没有呼叫。当只关心是否有连接两个电话号码的呼叫时，就使用无向图，其中当两个号码之间有呼叫时，就用边连接这两个电话号码。这个版本的呼叫图如图 6-10b 所示。

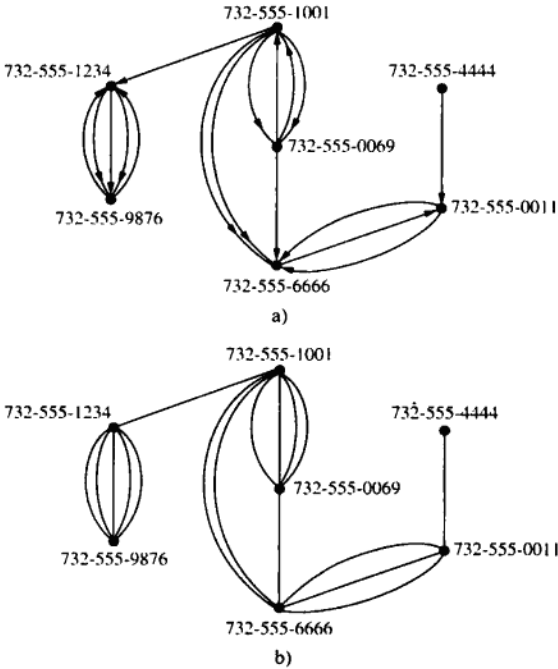


图 6-10 呼叫图

表示实际呼叫活动的呼叫图可以非常大。例如，在 AT&T 研究的一个呼叫图里，有大约 30 亿 9 千万个顶点和 40 亿条边，这个图表示在 20 天里进行的呼叫。在 6.4 节里将要进一步讨论呼叫图。



例 8 网络图 万维网可用有向图来建模，其中用顶点表示每个网页，并且若有从网页 a 指向网页 b 的链接，则有以 a 为起点以 b 为终点的边。因为几乎每秒钟都有新页面在网络上某处产生并且有其他页面被删除，所以网络图几乎是连续变化的。目前网络图有超过 30 亿个顶点和 200 亿条边。许多人正在研究网络图的性质，以便更好地理解网络的特性。6.4 节将继续讲解网络图，第 7 章将要解释网络爬虫(搜索引擎用它来产生网页的索引)是如何利用网络图的。

例9 优先图与并发处理 通过并发地执行某些语句, 计算机程序可以执行得更快。重要的是避免语句执行时还要用到尚未执行语句的结果。语句与前面语句的相关性可以表示成有向图。用顶点表示每个语句, 若在执行完第一个顶点所表示的语句之前不能执行第二个顶点所表示的语句, 则从第一个顶点到第二个顶点有一条边。这样的图称为优先图。图 6-11 显示计算机程序和优先图。例如, 该图说明在执行语句 S_1 、 S_2 和 S_4 之前不能执行语句 S_5 。

例10 路线图 可以用图对路线图建模。在这样的模型中, 定点表示交叉点, 而边表示路, 无向边表示双向道路, 有向边表示单行道; 多重无向边表示连接相同交叉点的多条双向道路, 多重有向边表示从一个交叉点开始、到第二个交叉点结束的多条单行道; 环表示环形路。因此, 如果一个路线图中只有双向道路, 没有环形路, 没有两条路连接相同的一对交叉点, 那么这个路线图可以用一个简单无向图来表示。一个路线图中只有单行道, 没有环形路, 没有两条路起始于相同的交叉点且终止于相同的交叉点, 那么它可以用一个简单有向图来表示。描述包含单行道和双向道路的线路图则需要混合图。

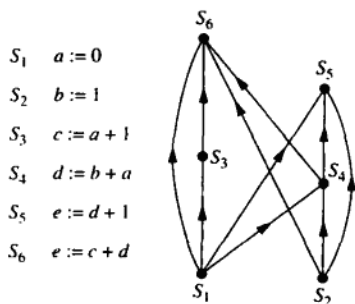
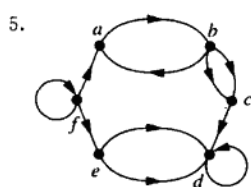
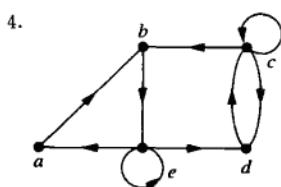
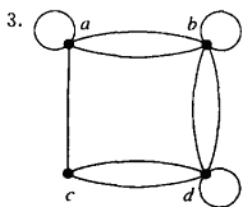
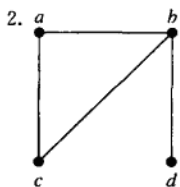


图 6-11 优先图

练习

- 画出表示航空公司航线的图模型, 并说出所用图的类型, 其中每天有 4 个航班从波士顿到纽华克, 2 个航班从纽华克到波士顿, 3 个航班从纽华克到迈阿密, 2 个航班从迈阿密到纽华克, 1 个航班从纽华克到底特律, 2 个航班从底特律到纽华克, 3 个航班从纽华克到华盛顿, 2 个航班从华盛顿到纽华克, 1 个航班从华盛顿到迈阿密, 其中
 - 若城市之间有航班(任何方向), 则在表示城市的顶点之间有边。
 - 对城市之间每个航班(任何方向)来说, 在表示城市的顶点之间有边。
 - 对城市之间每个航班(任何方向)来说, 在表示城市的顶点之间有边, 以及有表示在迈阿密起飞和降落特殊观光旅行的环。
 - 从表示航班出发城市的顶点到表示航班终止城市的顶点之间有边。
 - 对每个航班, 从表示出发城市的顶点到表示终止城市的顶点之间有边。

试确定练习 2~5 中所示的各个图是有向边还是无向边, 是否有多重边, 是否有一个或多个环。用你的答案指出该图属于表 6-1 中的哪种图。



- 设 G 是一个简单图。 R 是 G 的节点集上的关系, uRv 当且仅当 G 中有与 $\{u, v\}$ 相关联的边。证明: 关系 R 是对称的和非自反的关系。
- 集合 $A_1, A_2, \dots, A_n$ 的交图对每个集合来说都有一个顶点, 若两个集合有非空交集, 则有一条边连接代表这两个集合的顶点。构造下列集合的交图。
 - $A_1 = \{0, 2, 4, 6, 8\}$, $A_2 = \{0, 1, 2, 3, 4\}$
 $A_3 = \{1, 3, 5, 7, 9\}$, $A_4 = \{5, 6, 7, 8, 9\}$
 $A_5 = \{0, 1, 8, 9\}$
 - $A_1 = \{\dots, -4, -3, -2, -1, 0\}$
 $A_2 = \{\dots, -2, -1, 0, 1, 2, \dots\}$
 $A_3 = \{\dots, -6, -4, -2, 0, 2, 4, 6, \dots\}$

$$A_4 = \{\dots, -5, -3, -1, 1, 3, 5, \dots\}$$

$$A_5 = \{\dots, -6, -3, 0, 3, 6, \dots\}$$

$$c) A_1 = \{x \mid x < 0\}, A_2 = \{x \mid -1 < x < 0\}$$

$$A_3 = \{x \mid 0 < x < 1\}$$

$$A_4 = \{x \mid -1 < x < 1\}$$

$$A_5 = \{x \mid x > -1\}$$

$$A_6 = \mathbf{R}$$

8. 构造6种鸟的栖息地重叠图, 其中隐士鸫与旅鸫以及蓝松鸦竞争, 旅鸫也与嘲鸫竞争, 嘲鸫也与蓝松鸦竞争, 以及鹤鸟与多毛啄木鸟竞争。
9. 可用图来表示两个人是否生活在同一时代。画出这样的图来表示本书前4章里有生平介绍的在1900年前去世的属于同一时代的数学家和计算机科学家。(如果在同一年里两个人都在世, 就假设他们生活在同一时代。)
10. 构造公司董事会成员的影响图, 主席影响技术总监、市场总监和营运总监; 技术总监影响营运总监; 市场总监影响营运总监; 无人影响首席财务官或受其影响。
11. 在循环赛里, 老虎队击败蓝松鸦队, 蓝松鸦队击败红衣主教队, 红衣主教队击败北美金莺队, 用有向图有这样的结果建模。
12. 解释如何用一月份和二月份的电话呼叫图来确定改变了电话号码的人的新电话号码。
13. 如何用表示网络里发送电子邮件的图来找出最近改变了原来电子邮件地址的人?
14. 描述一个图模型, 它表示在某个聚会上每个人是否知道另一个人的名字。图中的边应该是有向的还是无向的? 是否应该允许多重边? 是否应该允许环?
15. 描述一种表示传统婚姻的图模型。这个图有什么特殊性质?
16. 构造下列程序的优先图:

$$\begin{aligned} S_1: x &= 0 \\ S_2: x &= x + 1 \\ S_3: y &= 2 \\ S_4: z &= y \\ S_5: x &= x + 2 \\ S_6: y &= x + z \\ S_7: z &= 4 \end{aligned}$$
17. 描述一种用来为群体里成对个人之间关系建模的基于图的离散结构, 其中每个人可能喜欢或者不喜欢另一个人, 或者中立, 而反过来的关系可以是不同的。[提示: 给一个有向图添加结构。分别处理表示两个人的顶点之间的反向边。]

6.2 图的术语和几种特殊的图

6.2.1 引言



本节将介绍图论的一些基本词汇。当解决许多不同类型的问题时可以使用这些词汇。

其中一个这样的问题涉及判定能否把图画在平面里, 使得没有两条边是交叉的。另一个例子是判定两个图是否具有顶点之间的一一对应, 使得这样的对应能够产生边之间的一一对应。我们还将介绍在例子和模型里经常用到的几种重要的图族。在这些特殊类型的图出现的地方, 将会介绍几种重要的应用。

6.2.2 基本术语

首先给出描述无向图的顶点和边的一些术语。

定义1 若 u 和 v 是无向图 G 中的一条边的端点, 则称两个顶点 u 和 v 称为在 G 里邻接(或相邻)。若 e 和 $\{u, v\}$ 相联系, 则边 e 称为关联顶点 u 和 v 。也可以说边 e 连接 u 和 v 。顶点 u 和 v 称为与 $\{u, v\}$ 相联系的边的端点。

为了反映出有多少条边关联着一个顶点, 有下述的定义。

定义2 在无向图里顶点的度是与该顶点关联的边的数目, 例外的情形是, 顶点上的环为顶点的度做出双倍贡献。顶点 v 的度表示成 $\deg(v)$ 。

例1 如图6-12所示, 图 G 和图 H 的顶点的度是什么?

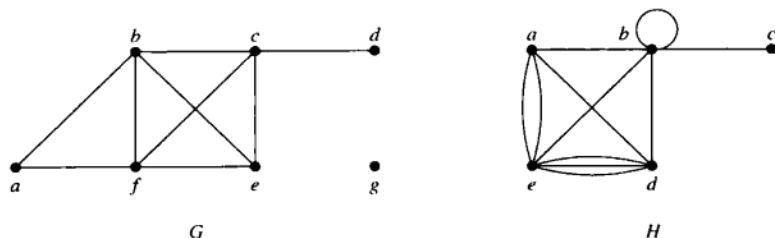


图6-12 无向图 G 和 H

解 在 G 里, $\deg(a)=2$, $\deg(b)=\deg(c)=\deg(f)=4$, $\deg(d)=1$, $\deg(e)=3$, $\deg(g)=0$. 在 H 里, $\deg(a)=4$, $\deg(b)=\deg(e)=6$, $\deg(c)=1$, $\deg(d)=5$. ■

把度为0的顶点称为孤立的. 因此孤立点不与任何顶点相邻. 例1里图 G 的顶点 g 是孤立的. 顶点是悬挂的, 当且仅当它的度是1. 因此悬挂点恰与1个其他顶点相邻. 例1里图 G 的顶点 d 是悬挂的.

当对图 $G=(V, E)$ 的所有顶点的度求和时, 得出了什么? 每条边都为顶点的度之和贡献2, 因为一条边恰好关联2个(可能相同)顶点. 这意味着顶点的度之和是边数的2倍. 下面描述握手定理, 这是因为在一条边上两个端点可以类比为一次握手涉及两只手这种情形.

分析一个图模型中顶点的度, 能够提供关于该模型的有用信息, 如例2所示.

例2 栖息地重叠图(6.1节例1中介绍的)中一个顶点的度表示什么意义? 该图中的哪些顶点是悬挂的, 哪些是孤立的? 运用6.1节图6-6中所示的栖息地重叠图解释你的答案.

解 栖息地重叠图中的两个顶点之间有边, 当且仅当这两个顶点所代表的两个物种相互竞争. 因此, 栖息地重叠图中的一个顶点的度表示了该生态系统里与此顶点代表的物种竞争的物种数目. 如果一个物种恰好与另一种物种竞争, 则相应的顶点是悬挂的. 最后, 如果某一物种不与其他任何物种竞争, 那么代表该物种的顶点就是孤立的.

例如, 6.1节图6-6中代表松鼠的顶点的度是4, 因为松鼠于其他4种物种竞争: 乌鸦、负鼠、浣熊和啄木鸟. 图6-6所示的图中, 老鼠是唯一的一个由悬挂顶点表示的物种, 因为老鼠只与鼯鼠竞争, 而其余的所有物种都至少与两种以上的他物种竞争. 如果一个物种仅与另一种物种竞争, 则表示它的顶点是悬挂的. 图6-6所示的图中没有孤立的顶点, 因为每种物种都至少与生态系统中的其他一种物种竞争.

定理1 握手定理 设 $G=(V, E)$ 是有 e 条边的无向图, 则

$$2e = \sum_{v \in V} \deg(v)$$

(注意即使出现多重边和环, 这个式子也仍然成立.)

例3 一个具有10个顶点而且每个顶点的度都为6的图, 有多少条边?

解 因为顶点的度之和是 $6 \cdot 10 = 60$, 所以 $2e = 60$. 因此 $e = 30$. ■

定理1说明无向图顶点的度之和是偶数. 这可以推导出许多结论, 其中一个结论作为定理2给出.

定理2 无向图有偶数个奇数度顶点.

证 在无向图 $G=(V, E)$ 里, 设 V_1 和 V_2 分别是偶数度顶点和奇数度顶点的集合. 于是

$$2e = \sum_{v \in V} \deg(v) = \sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v)$$

因为对 $v \in V_1$ 来说, $\deg(v)$ 是偶数, 所以上面等式右端的第一项是偶数. 另外, 上面等式右端的两项之和是偶数, 因为和是 $2e$. 因此, 和里的第二项也是偶数. 因为在第二项的和里, 所有的项都是奇数, 所以必然有偶数个这样的项. 因此, 有偶数个奇数度顶点. □

对带有有向边的图来说也有一些有用的术语。

定义 3 当 (u, v) 是带有有向边的图 G 的边时, 说 u 邻接到 v , 而且说 v 从 u 邻接。顶点 u 称为 (u, v) 的起点, v 称为 (u, v) 的终点。环的起点和终点是相同的。

因为带有有向边的图的边是有序对, 所以这时顶点度的定义细化成反映这个顶点作为起点和作为终点的不同度数。

定义 4 在带有有向边的图里, 顶点 v 的入度(表示成 $\deg^-(v)$)是以 v 作为终点的边数。顶点 v 的出度(表示成 $\deg^+(v)$)是以 v 作为起点的边数(注意顶点上的环对这个顶点的入度和出度的贡献都是 1)。

例 4 求出图 6-13 所示带有有向边的图 G 里每个顶点的入度和出度。

解 入度是: $\deg^-(a)=2, \deg^-(b)=2, \deg^-(c)=3,$
 $\deg^-(d)=2, \deg^-(e)=3, \deg^-(f)=0$ 。出度是: $\deg^+(a)=$
 $4, \deg^+(b)=1, \deg^+(c)=2, \deg^+(d)=2, \deg^+(e)=3,$
 $\deg^+(f)=0$ 。 ■

因为每条边都有一个起点和一个终点, 所以在带有有向边的图里, 所有顶点的入度之和等于出度之和。这两个和都等于图的边数。把这个结果表述成定理 3。

定理 3 设 $G=(V, E)$ 是带有有向边的图。于是

$$\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = |E|$$

带有有向边的图有许多性质是不依赖于边的方向的。因此忽略这些方向经常是有点用的。忽略边的方向后得出的无向图称为底无向图。带有有向边的图与它的底无向图有相同的边数。

6.2.3 一些特殊的简单图

下面要介绍几类简单图。这些图常常用作例子, 来自于许多应用。

例 5 完全图 n 个顶点的完全图(表示成 K_n)是在每对不同顶点之间都恰有一条边的简单图。对 $n=1, 2, 3, 4, 5, 6$ 来说, 图 6-14 显示图 K_n 。 ■

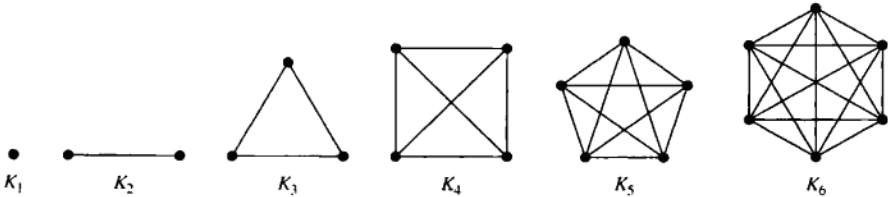


图 6-14 图 $K_n, n=1, 2, 3, 4, 5, 6$

例 6 圈图 圈图 $C_n (n \geq 3)$ 是由 n 个顶点 $v_1, v_2, \dots, v_n$ 以及边 $\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$ 组成的。图 6-15 显示圈图 C_3, C_4, C_5, C_6 。 ■

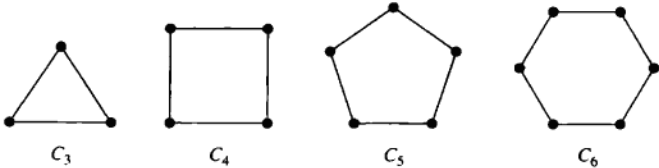


图 6-15 圈图 C_3, C_4, C_5, C_6

例 7 轮图 对 $n \geq 3$ 来说, 当给圈图 C_n 添加另一个顶点, 而且把这个新顶点与 C_n 里 n 个顶

点逐个连接时,就得出轮图 W_n 。图 6-16 显示轮图 W_3, W_4, W_5, W_6 。

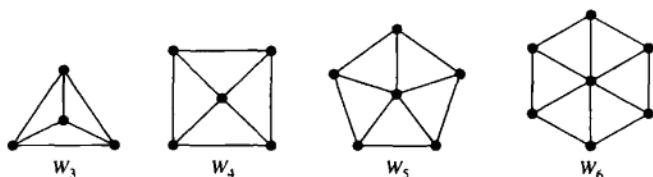


图 6-16 轮图 W_3, W_4, W_5, W_6

例 8 n 立方体图 n 立方体图(表示成 Q_n)是用顶点表示 2^n 个长度为 n 的位串的图。两个顶点相邻,当且仅当它们所表示的位串恰相差一位。图 6-17 显示图 Q_1, Q_2, Q_3 。注意可以从 n 立方体图 Q_n 来构造 $(n+1)$ 立方体图 Q_{n+1} ,方法是建立 Q_n 的两个副本,在 Q_n 的一个副本的顶点标记前加 0,在 Q_n 的另一个副本的顶点标记前加 1,并且加入连接那些标记只在第一位不同的两个顶点的边。在图 6-17 中,从 Q_2 来构造 Q_3 ,方法是画出 Q_2 的两个副本作为 Q_3 的顶面和底面,在底面每个顶点的标记前加 0,在顶面每个顶点的标记前加 1。(这里,“面”意为三维空间中立方体的一个面。试想在三维空间中画出以 Q_2 的两份拷贝作为立方体顶面和底面的 Q_3 的图形,然后在平面上画出结果描述方法(此 Q_3 图形——译者注)的投影。)

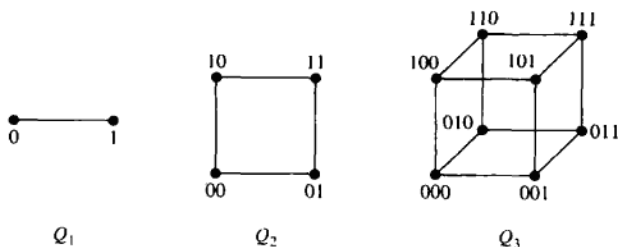


图 6-17 对于 $n=1, 2, 3$ 的 n 立方体图 Q_n

6.2.4 偶图

有时可以把图的顶点分成两个不相交的子集,使得每条边都连接一个子集里的顶点与另一个子集里的顶点。例如,考虑一下表示村民之间的婚姻关系的图,其中用顶点表示每个人,用边表示婚姻。在这个图里,每条边都连接表示男人的顶点子集里的顶点与表示女人的顶点子集里的顶点。这引出了定义 5。

定义 5 若把简单图 G 的顶点集分成两个不相交的非空集合 V_1 和 V_2 ,使得图里的每一条边都连接着 V_1 里的一个顶点与 V_2 里的一个顶点(因此 G 里没有边是连接着 V_1 里的两个顶点或 V_2 里的两个顶点),则 G 称为偶图[⊖]。当此条件成立,称 (V_1, V_2) 为 G 的顶点集的一个二部划分。

在例 9 里说明 C_6 是偶图,在例 10 里说明 K_3 不是偶图。

例 9 图 6-18 所示的 C_6 是偶图,因为把它的顶点集分成两个集合 $V_1 = \{v_1, v_3, v_5\}$ 和 $V_2 = \{v_2, v_4, v_6\}$, C_6 的每一条边都连接 V_1 里的一个顶点与 V_2 里的一个顶点。

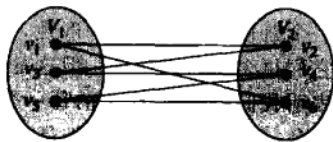


图 6-18 C_6 是偶图

例 10 K_3 不是偶图。为了看明白这一点,注意若把 K_3 的顶点集分成两个不相交的集合,则两个集合之一必然包含两个顶点。假如这个图是偶图,那么这两个顶点就不能用边连接,但是在 K_3 里每一个顶点都用边连接到其他每个顶点。

[⊖] 也称为二部图、二分图。——译者注

例 11 图 6-19 所示的图 G 和 H 是否为偶图？

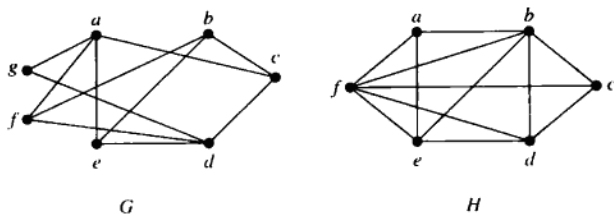


图 6-19 无向图 G 和 H

解 图 G 是偶图，因为它的顶点集是两个不相交集 $\{a, b, d\}$ 和 $\{c, e, f, g\}$ 的并，每条边都连接一个子集里的一个顶点与另一个子集里的一个顶点。（注意对偶图 G 来说，不必让 $\{a, b, d\}$ 里每一个顶点与 $\{c, e, f, g\}$ 里每一个顶点都相邻。例如 b 与 g 就不相邻。）

图 H 不是偶图，因为它的顶点集不能分成两个子集，使得边都不连接同一个子集的两个顶点（读者应当通过考虑顶点 a, b, f 来验证它）。 ■

定理 4 给出了决定图是否为偶图的有用准则。

定理 4 一个简单图是偶图，当且仅当能够对图中的每个顶点赋以两种不同的颜色，而不让相邻接的顶点被赋以相同的颜色。

证 首先，假设 $G=(V, E)$ 是一个二分简单图。那么， $V=V_1 \cup V_2$ ，其中 V_1 和 V_2 是不相交的顶点集且 E 中的每一条边都连接一个 V_1 中的顶点和一个 V_2 中的顶点。如果对 V_1 中的每个顶点赋予一种颜色而 V_2 中的顶点赋予第二种颜色，那么相邻接的顶点就不会被赋予相同的颜色。

现在假设可以仅用两种颜色对图中的顶点着色，以使得相邻接的顶点都被赋予不同的颜色。让 V_1 为一种颜色的顶点集， V_2 为另一种颜色的顶点集，则 V_1 和 V_2 不相交且 $V=V_1 \cup V_2$ 。此外，每条边都连接一个 V_1 中的顶点和一个 V_2 中的顶点，因为并无相邻接的顶点同在 V_1 中或同在 V_2 中。所以， G 是偶图。 □

例 12 将说明如何用定理 4 判断一个图是不是偶图。

例 12 用定理 4 判断例 11 中的图是不是偶图。

解 首先考虑图 G 。将图 G 中的每个顶点赋予两种颜色（如红色和蓝色）中的一种，使得 G 中的每一条边都连接一个红色顶点和一个蓝色顶点。不失一般性，我们先任意地赋予顶点 a 红色。然后，必须对 c, e, f 和 g 顶点赋予蓝色，因为这些顶点与顶点 a 相邻接。为了避免一条边有两个蓝色的端点，所有与 c, e, f 或 g 顶点相邻接的顶点必须被染成红色。这就是说，必须把 b 和 d 染成红色（也意味着， a 必须染成红色，而 a 已经是红色的了）。现在，已经将所有的顶点都进行了染色， a, b 和 d 为红色， c, e, f 和 g 为蓝色。查看每一条边，我们看见每条边都连接一个红色顶点和一个蓝色顶点。因此，由定理 4，图 G 是偶图。

接下来，将对图 H 中的每个顶点赋予红色或蓝色，以使 H 中的每一条边都连接一个红色顶点和一个蓝色顶点。不失一般性，我们任意地对 a 赋以红色。然后，必须对 b, e 和 f 赋以蓝色，因为它们每个都与 a 邻接。但这是不可能的，因为 e 和 f 相邻接，因此不能两个都赋以蓝色。这一矛盾表明我们不能对 H 中的每一个顶点赋以两种颜色中的一种，以使没有邻接的顶点被赋以相同的颜色。根据定理 4， H 不是偶图。 ■

定理 4 是图论中“图的着色”部分的一个结果示例。图的着色是图论中一个重要的部分，有着许多重要的应用。我们将在 6.8 节进一步学习图的着色。

决定一个图是否是偶图的另一个标准是基于路径的概念，将在 6.4 节学习这个概念。一个图是偶图，当且仅当从一个顶点出发经过奇数条不同的边再回到它本身是不可能的。当我们在 6.4 节讨论图中的路径和环路时，将会让这个概念变得更加精确（参见那一节的练习 26）。

例 13 完全偶图 完全偶图 $K_{m,n}$ 是顶点集分成分别含有 m 和 n 个顶点的两个子集的图。两个顶点之间有边当且仅当一个顶点属于第一个子集而另一个顶点属于第二个子集。图 6-20 显示完全偶图 $K_{2,3}$, $K_{3,3}$, $K_{3,5}$ 以及 $K_{2,6}$ 。

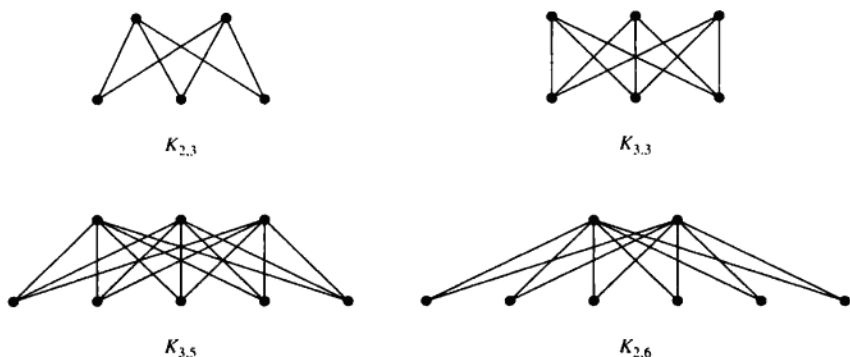


图 6-20 部分完全偶图

6.2.5 特殊类型的图的一些应用

以下例 14~16 将要说明如何把特殊类型的图用在数据通信和并行处理的模型里。

例 14 任务分配 设一个小组内有 m 名雇员，有 j 个不同的任务需要完成，其中 $m \leq j$ 。每个员工都受过相关训练，可以做这 j 个任务中的一个或多个。我们可以用一个图来为员工的能力建模。每个员工和每个任务都用一个顶点表示。对于每个员工，从表示该员工的顶点到表示该员工受训能做的所有任务，都作一条边。注意到这个图的顶点集可以被划分成两个不相交的集合，一个代表员工的顶点集，另一个代表任务的顶点集。而且每一条边都连接一个表示员工的顶点和一个表示任务的顶点。因此，该图是一个偶图。

例如，设一个小组由 4 名员工：Alvarez, Berkowitz, Chen 和 Davis；再设完成一个项目有 4 种工作要做：需求，架构，实现和测试。设 Alvarez 受训可以完成需求分析和测试；Berkowitz 接受的训练是完成架构、实现和测试；Chen 接受的训练是完成需求、架构和实现；而 Davis 只接受训练做需求。可以用图 6-21 所示的偶图对员工的这些能力建模。

为了完成项目，必须给员工指派任务，以满足每个任务都有一个员工接手，而且每个员工最多只分配一个任务。在这种情况下，可以让 Alvarez 做测试，Berkowitz 做实现，Chen 做架构，Davis 做需求。如图 6-21 所示（带颜色的线表示了任务的分配）。

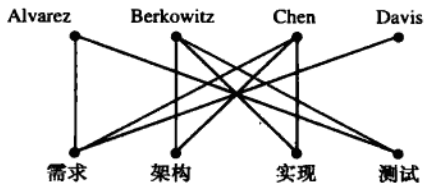


图 6-21 为受训员工分配工作建模

寻求一种对员工的任务分配方案可以看成是在图模型中寻找一个匹配。简单图的匹配是该图边集的一个子集，该子集中没有两条边与同一个顶点相关联。最大匹

配是所含边数最多的匹配。换句话说，匹配是边集的子集使得如果 $\{s, t\}$ 和 $\{u, v\}$ 是匹配中的边，那么 s, t, u 和 v 就是不同的顶点。在给职员分配任务的过程中，要给多数数的职员都分配，实际上就是寻找员工能力的图模型中的最大匹配。（感兴趣的读者可以在关于图论的书中找到更多有关匹配的内容，这些书中包括 [GrYe06]。）



例 15 局域网 在一座大楼里，像小型计算机和个人电脑这样的各种计算机，以及像打印机和绘图仪这样的外设，都可以用局域网来连接。一些这样的网络是基于星形拓扑，其中所有设备都连接到中央控制设备。局域网用图 6-22a 所示的完全偶图 $K_{1,n}$ 来表示。通过中

央控制设备在设备间传输信息。

其他局域网是基于环形拓扑, 其中每个设备恰恰连接到两个其他设备。带环形拓扑的局域网用图 6-22b 所示的 n 圈图 C_n 来建模。消息围绕着圈从设备送到设备, 直到抵达消息目的地为止。

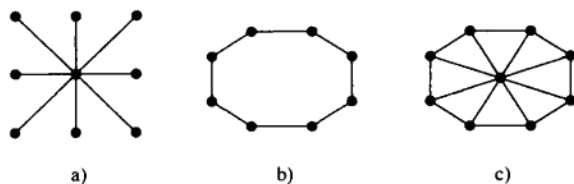


图 6-22 局域网的星形、环形以及混合拓扑

最后, 一些局域网采用这两种拓扑的混合形式。消息是围绕着环或通过中央设备来传送。这样的冗余使得网络更加可靠。带冗余的局域网用图 6-22c 所示的轮图 W_n 来建模。

例 16 并行计算的互连网络 直到最近, 计算机执行程序还是一次完成一个操作。因此, 为解决问题而写的算法都设计成一次执行一步; 这样的算法称为串行的(几乎所有本书描述的算法都是串行的)。不过, 像气象模拟、医学图像分析以及密码分析这样的许多高强度计算问题, 即使在超级计算机上, 也不能通过串行操作在合理的时间范围内解决。另外, 计算机执行基本操作的速度, 还存在物理的限制, 所以总是有问题不能用串行操作在合理的时间范围内解决。

并行处理利用包含许多自带内存的单处理器的计算机来帮助克服串行计算机的局限性。并行算法把问题分成可并发解决的若干子问题, 于是可以设计并行算法, 用带有多处理器的计算机来快速解决问题。在并行算法里, 单个指令流控制着算法的执行, 包括把子问题送到不同的处理器, 以及把子问题的输入和输出定向到适当的处理器。

采用并行处理时, 一个处理器需要另一个处理器产生的输出。因此处理器需要互联。可用适当类型的图来表示带有多重处理器的计算机里处理器的互连网络。在以下讨论里将要描述最常用类型的并行处理器互连网络。用来实现具体并行算法的互连网络的类型, 差异在于处理器之间交换数据的需求量和所需要的速度, 当然还有可用的硬件等。

最简单却又最昂贵的网络互联处理器, 包含每对处理器之间的双向连接。当有 n 个处理器时, 这样的网络表示成 n 个顶点上的完全图 K_n 。不过, 这种类型的互连网络有严重的问题, 它所需要的连接数太大。实际上, 处理器的直接连接数目是有限的, 所以当处理器数很大时, 处理器不能直接连接到所有其他处理器。例如, 当有 64 个处理器时, 就需要 $C(64, 2) = 2016$ 个连接, 每个处理器都得直接连接到其他 63 个处理器。

另一方面, 互联 n 个处理器的最简单方式或许是使用称为线性阵列的排列方式。除 P_1 和 P_n 外的每个处理器 P_i 都通过双向连接来连接到相邻处理器 P_{i-1} 和 P_{i+1} 。 P_1 只连接 P_2 , P_n 只连接 P_{n-1} 。图 6-23 显示 6 处理器线性阵列。这种方式的缺点是为了让处理器共享信息, 有时需要使用大量的称为 **hop** 的中间连接。

栅格网络(或二维阵列)是通用的互连网络。在这样的网络里, 处理器个数是一个完全平方数, 比方说 $n = m^2$ 。 n 个处理器标记成 $P(i, j)$, $0 \leq i \leq m-1$, $0 \leq j \leq m-1$ 。双向连接把处理器 $P(i, j)$ 连接到它的 4 个相邻处理器 $P(i \pm 1, j)$ 和 $P(i, j \pm 1)$, 只要这些处理器是在栅格里。(注意, 栅格角上的 4 个处理器只有 2 个相邻处理器, 边界上其他处理器只有 3 个相邻处理器; 有时也用每个处理器恰有 4 个连接的变种的栅格网络。)栅格网络限制每个处理器的连接数。某些成对处理器之间的通信需要 $O(\sqrt{n}) = O(m)$ 个中间连接(见本节末的练习 33)。图 6-24 表示 16 个处理器的栅格网络的图。



图 6-23 6 个处理器的线性阵列

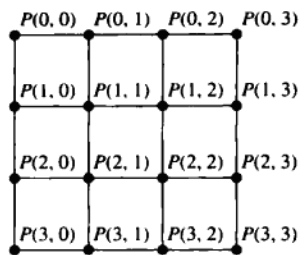


图 6-24 16 个处理器的栅格网络

超立方体是互连网络的一个重要类型。对这样的网络来说，处理器数是 2 的幂， $n=2^m$ 。 n 个处理器标记成 $P_0, P_1, \dots, P_{n-1}$ 。每个处理器都有到 m 个其他处理器的双向连接。连接到处理器 P_i 上的处理器，其下标的二进制表示与 i 的二进制表示恰恰相差 1 位。超立方体网络在每个处理器的直接连接数与保证处理器通信的中间连接数之间取得了平衡。已经用超立方体网络建造了许多计算机，而且用超立方体网络设计了许多算法。 n 立方体图 Q_n 表示带 n 个处理器的超立方体网络。图 6-25 显示 8 个处理器的超立方体网络(图 6-25 中画 Q_3 的方式与图 6-17 所示的方式不同)。

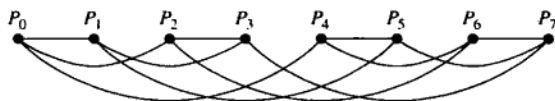


图 6-25 8 个处理器的超立方体网络

6.2.6 从旧图到新图

有时解决问题只需要图的一部分。例如，只关心大型计算机网络中涉及纽约、丹佛、底特律以及亚特兰大的计算机中心的那一部分。其他的计算机中心以及不连接到这 4 个具体计算机中心里任何 2 个的所有电话线可以忽略。在大网络的图模型里，可以删除这 4 处外的计算机中心所对应的顶点，可以删除所有与所删除顶点关联的边。删除后所剩下的图称为原图的子图。

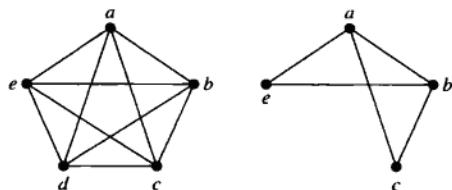
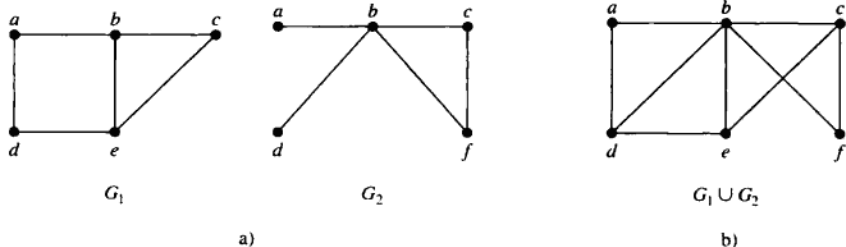
定义 6 图 $G=(V, E)$ 的子图是图 $H=(W, F)$ ，其中 $W \subseteq V$ 而且 $F \subseteq E$ 。

例 17 图 6-26 所示的图 G 是 K_5 的子图。

可以用各种方式组合两个或更多的图。把这些图的所有顶点和边所组成的图称为这些图的并图。两个简单图的并图的定义如下。

定义 7 两个简单图 $G_1=(V_1, E_1)$ 和 $G_2=(V_2, E_2)$ 的并图是带有顶点集 $V_1 \cup V_2$ 和边集 $E_1 \cup E_2$ 的简单图。 G_1 和 G_2 的并图表示成 $G_1 \cup G_2$ 。

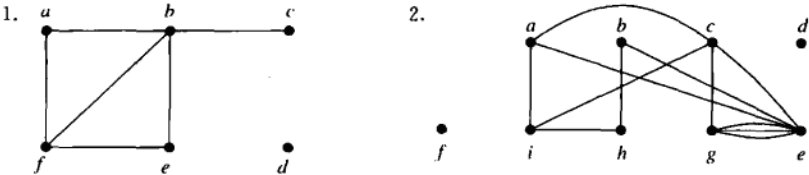
例 18 求图 6-27a 所示的图 G_1 和 G_2 的并图。

图 6-26 K_5 的子图图 6-27 a) 简单图 G_1 和 G_2 ；b) 它们的并 $G_1 \cup G_2$

解 并图 $G_1 \cup G_2$ 的顶点集是两个顶点集的并, 即 $\{a, b, c, d, e, f\}$ 。并图的边集是两个边集的并。并图显示在图 6-27b 中。

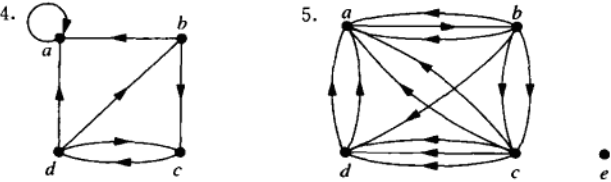
练习

在练习 1~2 里, 求所给无向图的顶点数、边数以及每个顶点的度。指出所有孤立点和悬挂点。



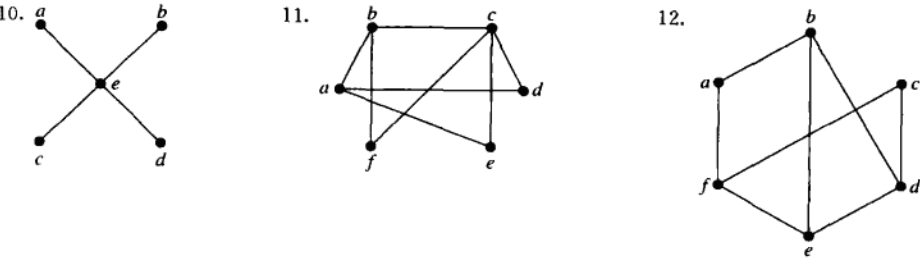
3. 带有 15 个顶点而且每个顶点的度都为 5 的简单图能否存在?

在练习 4~5 里, 对给定的有向多重图来说, 确定顶点数和边数, 并求出每个顶点的入度和出度。



6. 构造出图 6-13 里带有向边的图的底图。
7. 在合作图里, 顶点的度表示什么? 孤立点和悬挂点表示什么?
8. 在 6.1 节例 7 所描述的电话呼叫图里, 顶点的入度和出度表示什么? 在这个图的无向版本里, 顶点的度表示什么?
9. 在为循环赛建模的有向图里, 顶点的入度和出度表示什么?

在练习 10~12 里, 确定图是否为偶图。



13. 设一个新公司有 5 名员工: Zamora、Agraharam、Smith、Chou 和 Macintyre。每个员工承担下列 6 种责任之一: 策划, 宣传, 销售, 市场, 开发, 以及工业关系。每个员工能够做这些工作中的一种或多种: Zamora 能做策划、销售、市场或工业关系; Agraharam 能做策划或开发; Smith 能做宣传、销售或工业关系; Chou 能做策划、销售或工业关系; Macintyre 能做策划、宣传、销售或工业关系。
a) 用一个偶图对这些员工的能力建模。
b) 找出一个能力的指派, 使得每个员工被指定一种能力。
14. 下列图有多少个顶点和多少条边?
a) K_n b) C_n c) W_n
d) $K_{m,n}$ e) Q_n

一个图的度序列是由该图的各节点的度按非递增顺序排列的序列。例如, 本节例 1 中图 G 的度序列就是 4, 4, 4, 3, 2, 1, 0。

15. 求下列各个图的度序列。

- a) K_4 b) C_4 c) W_4
d) $K_{2,3}$ e) Q_3

16. 图 K_n 的度序列是什么(其中 n 是正整数)? 并解释你的答案。
17. 若图的顶点有度 5, 2, 2, 2, 2, 1, 则它有多少条边? 画出这样的图。

如果一个序列是一个简单图的度序列, 那么该序列是成图的。

18. 判断下列序列是否是成图的。如果是, 请画出一个简单图使其具有给定的度序列。

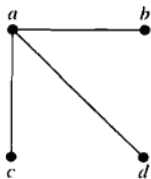
- a) 3, 3, 3, 3, 2 b) 5, 4, 3, 2, 1
c) 4, 4, 3, 2, 1 d) 4, 4, 3, 3, 3
e) 3, 2, 2, 1, 0 f) 1, 1, 1, 1, 1

19. 证明: 一个由非负整数非递增排列的序列 $d_1, d_2, \dots, d_n$ 是成图序列当且仅当把序列 $d_2 - 1, \dots, d_{d_1 - 1} - 1, d_{d_1 - 2}, \dots, d_n$ 中的元素重新排序为非递增而得到的序列是成图序列。

20. 证明: 每个元素之和为偶数的非递增非负整数序列都是某个伪图的度序列。伪图是允许有环的无向图。
[提示: 首先通过给每个顶点添加尽可能多的环来构造一个图, 然后添加一些边连接度为奇数的顶点。解释为什么这种构造方法能够证明此问题。]

21. 至少带有 1 个顶点的 K_3 的子图有多少个?

22. 画出右面这个图的所有子图。



若简单图每个顶点的度都相等, 则这个图称为正则的。若正则图每个顶点的度都为 n , 则这个图称为 n 正则的。

23. 对哪些 n 值来说, 下列图是正则图?

- a) K_n b) C_n c) W_n d) Q_n

24. 度都为 4 而且带有 10 条边的正则图有多少个顶点?

25. 求右面的简单图对的并图(假设带有相同端点的边是相同的)。

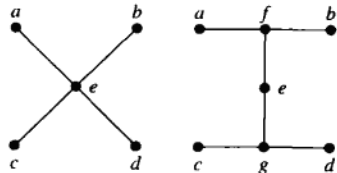
26. 简单图 G 的补图 $\bar{G}$ 与 G 有相同的顶点。两个顶点在 $\bar{G}$ 里相邻, 当且仅当它们在 G 里不相邻。求下列图。

- a) $\bar{K_n}$ b) $\bar{K_{m,n}}$ c) $\bar{C_n}$ d) $\bar{Q_n}$

27. 若简单图 G 有 v 个顶点和 e 条边, 则 $\bar{G}$ 有多少条边?

28. 简单图 G 的度序列为 $d_1, d_2, \dots, d_n$ 。求 $\bar{G}$ 的度序列。

29. 证明: 若 G 是有 n 个顶点的简单图, 则 G 和 $\bar{G}$ 的并图是 K_n 。



表示成 G 的有向图 $G=(V, E)$ 的逆图是图 (V, F) , 其中 $(u, v) \in F$, 当且仅当 $(v, u) \in E$ 。

30. 描述 6.1 节练习 4~5 里每个图的逆图。

31. 证明: 图 G 是它自身的逆图, 当且仅当 G 所关联的关系(参见 8.3 节)是对称的。

32. 画出 9 个并行处理器互联的栅格网络。

33. 证明: 在 $n=m^2$ 个处理器的栅格网络里, 用 $O(\sqrt{n})=O(m)$ 个 hop 就能让每一对处理器互相通信。

6.3 图的表示和图的同构

6.3.1 引言

图的表示方式有很多种。在本章里将看到选择最方便的表示会有助于对图的处理。本节将要说明如何用多种不同的方式来表示图。

有时两个图在这样一种意义下恰好具有相同的形式, 就是在两个图的顶点之间存在着一一对应, 这个对应保持边的对应关系。在这种情形下就说两个图是同构的。确定两个图是否同构, 这是本节里将要研究的重要图论问题。

6.3.2 图的表示

表示不带多重边的图的一种方式列出这个图的所有边。另一种表示不带多重边的图的方式是用邻接表, 它规定与图的每个顶点相邻的顶点。

例 1 用邻接表描述图 6-28 所示的简单图。

解 表 6-2 列出与图的每个顶点相邻的顶点。

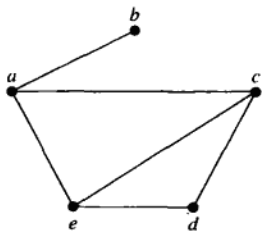


图 6-28 简单图

表 6-2 简单图的边表

顶 点	相 邻 顶 点
a	b, c, e
b	a
c	a, d, e
d	c, e
e	a, c, d

例 2 通过列举在图的每个顶点上发出的边的终点，表示图 6-29 所示的有向图。

解 表 6-3 表示图 6-29 所示的有向图。

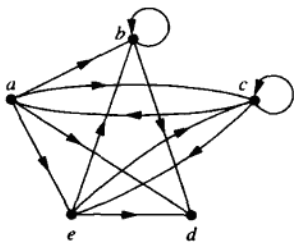


图 6-29 一个有向图

表 6-3 有向图的边表

起 点	终 点
a	b, c, d, e
b	a, c
c	a, b, d, e
d	a, c, e
e	a, b, c, d

6.3.3 邻接矩阵

若图里有许多边，则把图表示成边表或邻接表，会不便于执行图的算法。为了简化计算，可用矩阵表示图。在此将给出两种类型的常用的表示图的矩阵。一种类型是基于顶点的相邻关系，另一种类型是基于顶点与边的关联关系。

假设 $G=(V, E)$ 是简单图，其中 $|V|=n$ 。假设把 G 的顶点任意地排列成 $v_1, v_2, \dots, v_n$ 。对这个顶点表来说， G 的邻接矩阵 A (或 A_G) 是一个 $n \times n$ 的 0-1 矩阵，它满足这样的性质：当 v_i 和 v_j 相邻时第 (i, j) 项是 1，当 v_i 和 v_j 不相邻时第 (i, j) 项是 0。换句话说，若邻接矩阵是 $A=[a_{ij}]$ ，则

$$a_{ij} = \begin{cases} 1 & \text{若 } \{v_i, v_j\} \text{ 是 } G \text{ 的一条边} \\ 0 & \text{否则} \end{cases}$$

例 3 用邻接矩阵表示图 6-30 所示的图。

解 把顶点排列成 a, b, c, d 。表示这个图的矩阵是

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

例 4 画出带有相对于顶点顺序 a, b, c, d 的邻接矩阵的图。

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

解 图 6-31 显示带有这个邻接矩阵的图。

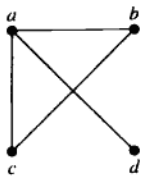


图 6-30 简单图



图 6-31 带有给定的邻接矩阵的图

注意，图的邻接矩阵依赖于所选择的顶点的顺序。因此带 n 个顶点的图有 $n!$ 个不同的邻接矩阵，因为 n 个顶点有 $n!$ 个不同的顺序。

简单图的邻接矩阵是对称的，即 $a_{ij} = a_{ji}$ ，因为当 v_i 和 v_j 相邻时，这两个项都是 1，否则都是 0。另外，因为简单图无环，所以每一项 a_{ii} 都是 0， $i = 1, 2, 3, \dots, n$ 。

邻接矩阵也可用来表示带环和多重边的无向图。把顶点 a_i 上的环表示成邻接矩阵第 (i, i) 位置上的 1。当出现多重边时，邻接矩阵不再是 0-1 矩阵，这是因为邻接矩阵的第 (i, j) 项等于与 $\{a_i, a_j\}$ 关联的边数。包括多重图与伪图在内的所有无向图都具有对称的邻接矩阵。

例 5 用邻接矩阵表示图 6-32 所示的伪图。

解 顶点顺序为 a, b, c, d 的邻接矩阵是

$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}$$

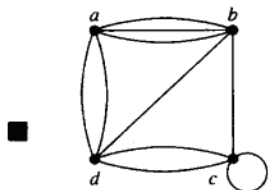


图 6-32 伪图

我们曾在第 5 章里用 0-1 矩阵表示有向图。若有向图 $G = (V, E)$ 从 v_i 到 v_j 有边，则它的矩阵在 (i, j) 位置上有 1，其中 $v_1, v_2, \dots, v_n$ 是有向图任意的顶点序列。换句话说，若 $A = [a_{ij}]$ 是相对于这个顶点表的邻接矩阵，则

$$a_{ij} = \begin{cases} 1 & \text{若 } \{v_i, v_j\} \text{ 是 } G \text{ 的一条边} \\ 0 & \text{否则} \end{cases}$$

有向图的邻接矩阵不必是对称的，因为当从 a_i 到 a_j 有边时，从 a_j 到 a_i 可以没有边。

邻接矩阵也可用来表示有向多重图。同样，当有连接两个顶点的同向多重边时，这样的矩阵不是 0-1 矩阵。在有向多重图的邻接矩阵里， a_{ij} 等于关联到 (v_i, v_j) 的边数。

在邻接表和邻接矩阵之间取舍 当一个简单图包含的边相对较少，即该图是一个稀疏图的时候，通常邻接表比邻接矩阵更适合表示它。例如，当每个顶点的度都不超过 c ， c 是一个比 n 小很多的常数，则邻接表中的每一条链表都最多只含有 c 个节点。所以整个邻接表中的元素不超过 cn 个。另一方面， n 个顶点的图的邻接矩阵含有 n^2 个元素。但是，需要注意的是，稀疏图的邻接矩阵是一个稀疏矩阵，即矩阵中只有少量元素不为 0。有特别的技术专门表示和处理稀疏矩阵。

现在设想一个稠密的简单图，它含有很多条边，例如，它含有的边数超过总的所有可能有的边的一半。在这种情形下，用邻接矩阵来表示图就比用邻接表好。为了知道原因，我们来比较判断某条边 $\{v_i, v_j\}$ 是否存在的复杂度。在邻接矩阵中，可以通过查看第 (i, j) 个元素来决定这条边是否存在。如果该元素是 1 边就存在，如果是 0 边就不存在。所以，只需要一次比较，即将第 (i, j) 个元素和 0 比较，就可以判断这条边是否存在。而另一方面，如果使用邻接表表示这个图，就需要搜索 v_i 或 v_j 的链表中的顶点才能判断这条边是否存在。当图含有的边很多时，这需要 $O(|V|)$ 次的比较。

6.3.4 关联矩阵

表示图的另一种常用方式是用关联矩阵。设 $G=(V, E)$ 是无向图。设 $v_1, v_2, \dots, v_n$ 是顶点而 $e_1, e_2, \dots, e_m$ 是边。则相对于 V 和 E 的这个顺序的关联矩阵是 $n \times m$ 矩阵 $M=[m_{ij}]$ ，其中

$$m_{ij} = \begin{cases} 1 & \text{当边 } e_j \text{ 关联 } v_i \text{ 时} \\ 0 & \text{否则} \end{cases}$$

例 6 用关联矩阵表示图 6-33 所示的图。

解 关联矩阵是

	e_1	e_2	e_3	e_4	e_5	e_6
v_1	1	1	0	0	0	0
v_2	0	0	1	1	0	1
v_3	0	0	0	0	1	1
v_4	1	0	1	0	0	0
v_5	0	1	0	1	1	0

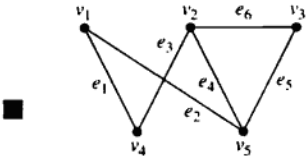


图 6-33 无向图

关联矩阵也可用来表示多重图边和环。在关联矩阵里用各项相等的列来表示多重边，因为这些边关联同样一对顶点。用恰有一项等于 1 的列来表示环，它对应于环所关联的顶点。

例 7 用关联矩阵表示图 6-34 所示的伪图。

解 这个图的关联矩阵是

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
v_1	1	1	1	0	0	0	0	0
v_2	0	1	1	1	0	1	1	0
v_3	0	0	0	1	1	0	0	0
v_4	0	0	0	0	0	0	1	1
v_5	0	0	0	0	1	1	0	0

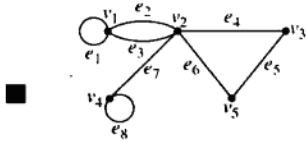


图 6-34 伪图

6.3.5 图的同构

我们经常需要知道是否有可能以同样的方式来画出两个图。例如，在化学里用图来为化合物建模。不同的化合物可能分子式相同但结构不同。这样的化合物不能用以同样方式来画的图表示。表示过去已知化合物的图可以用来判定想象中的新化合物是否已经研究过了。

对具有同样结构的图来说，存在着一些有用的术语。

定义 1 设 $G_1=(V_1, E_1)$ 和 $G_2=(V_2, E_2)$ 是简单图，若存在一对一的和映上的从 V_1 到 V_2 的函数 f ，且 f 具有这样的性质：对 V_1 里所有的 a 和 b 来说， a 和 b 在 G_1 里相邻当且仅当 $f(a)$ 和 $f(b)$ 在 G_2 里相邻，就说 G_1 与 G_2 是同构的。这样的函数 f 称为同构[⊖]。

换句话说，当两个简单图同构时，两个图的顶点之间具有保持相邻关系的一一对应。

例 8 证明：图 6-35 所示图 $G=(V, E)$ 和 $H=(W, F)$ 同构。

解 函数 f 满足 $f(u_1)=v_1, f(u_2)=v_4, f(u_3)=v_3, f(u_4)=v_2$ ，它是 V 和 W 之间的一一对应。为了看出这个对应保持相邻关系，注意 G 里相邻的顶点是 u_1 和 u_2, u_1 和 u_3, u_2 和 u_4 ，以及 u_3 和 u_4 ，由 $f(u_1)=v_1$ 和 $f(u_2)=v_4, f(u_1)=v_1$ 和 $f(u_3)=v_3, f(u_2)=v_4$ 和 $f(u_4)=v_2$ ，以及 $f(u_3)=v_3$ 和 $f(u_4)=v_2$ 所组成的每一对顶点都是在 H 里相邻的。 ■

判定两个简单图是否同构常常是一件困难的事情。在两个带有 n 个顶点的简单图顶点集之间有 $n!$ 种可能的一一对应。若 n 太大，则通过检验每一种对应来看它是否保持相邻关系是不可行的。

⊖ 同构(isomorphism)这个词来自两个希腊语字根：表示“相等”的 isos 和表示“形式”的 morphe。

有时候要说明两个图不同构并不困难。特别地,如果能找到某个属性,两个图中只有一个图具有这个属性,且该属性只在讨论同构时使用,就可以说这两个图不同构。这种保留在图的同构问题中使用的属性被称为图的非变体性。例如,同构的简单图必然具有相同的顶点数,因为在这些图的顶点集之间有着——对应。

同构的简单图必然有相同的边数,因为在顶点之间的一一对应建立了边之间的一一对应。另外,同构的简单图的对应顶点的度必然相同。即图 G 中的 d 度顶点 v 必然对应图 H 中的 d 度顶点 $f(v)$,这是因为在图 G 中顶点 w 与 v 相邻,当且仅当在图 H 中 $f(v)$ 与 $f(w)$ 相邻。

例 9 说明图 6-36 所示的图不同构。

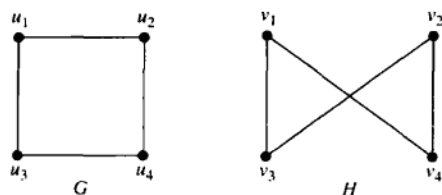


图 6-35 图 G 和 H

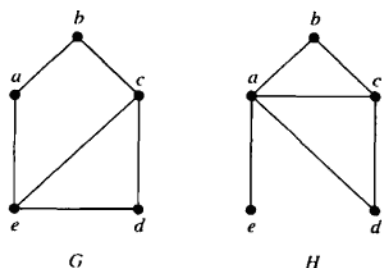


图 6-36 图 G 和 H

解 G 和 H 都具有 5 个顶点 6 条边。不过, G 有 1 度顶点 e 而 H 没有 1 度顶点。所以 G 与 H 是不同构的。 ■

顶点数、边数以及顶点度都是在同构下的不变量。若两个简单图的这些量有任何不同,则这两个图就不是同构的。不过,当这些量都相同时,也不一定意味着两个图同构。目前还没有已知的用来判定简单图是否同构的不变量集。

例 10 判定图 6-37 所示的图是否同构。

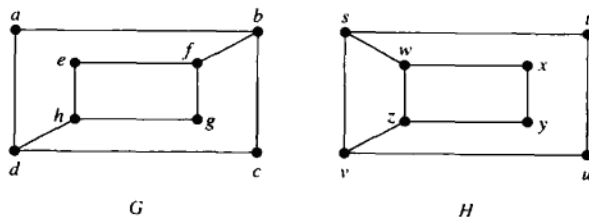


图 6-37 图 G 和 H

解 图 G 和 H 都具有 8 个顶点 10 条边。它们都具有 4 个 2 度顶点和 4 个 3 度顶点。因为这些不变量都相同,所以它们可能会是同构的。

然而 G 和 H 不是同构的。为了看明白这一点,注意到因为在 G 里 $\deg(a)=2$, 所以 a 必然对应于 H 里的 t , u , x 或 y , 这是因为这些顶点是 H 里的 2 度顶点。不过, H 里的这 4 个顶点中的每一个都与 H 里另一个 2 度顶点相邻,但是在 G 里 a 却不是这样的。

看出 G 与 H 不同构的另一种方式是,注意到若这两个图同构,则由 3 度顶点和连接它们的边所组成的子图同构(读者应当验证它)。然而图 6-38 所示的这些子图却

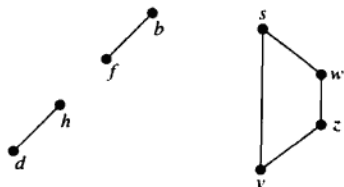


图 6-38 由 3 度顶点和连接它们的边所组成的 G 和 H 的子图

不是同构的。

为了说明从图 G 的顶点集到图 H 的顶点集的函数 f 是一个同构，需要说明 f 保持边的关系。一种有助于这样做的方式是利用邻接矩阵。具体地说，为了说明 f 是一个同构，就说明 G 的邻接矩阵与 H 的邻接矩阵相同，其中 G 的邻接矩阵的行和列的标记都是 G 里的顶点， H 的邻接矩阵的行和列的标记都是 G 里的对应顶点在 f 下的像。例 11 解释如何这样做。

例 11 判定图 6-39 所示的图 G 和 H 是否同构。

解 G 和 H 都具有 6 个顶点 7 条边，都具有 4 个 2 度顶点和 2 个 3 度顶点。还容易看出由 2 度顶点和连接它们的边所组成的 G 和 H 的子图是同构的(读者应当验证它)。因为 G 和 H 对这些不变量来说是相同的，这就有理由试着找出一个同构 f 。

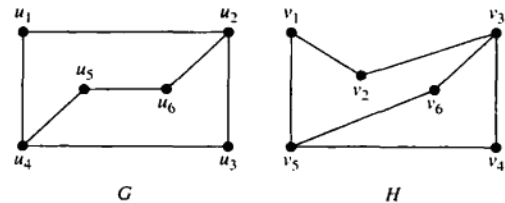


图 6-39 图 G 和 H

现在定义函数 f ，然后判定它是否同构。因为 $\deg(u_1)=2$ 而且 u_1 不与任何其他 2 度顶点相邻，所以 u_1 的像必然是 v_1 或 v_6 ，它们是 H 里仅有的不与 2 度顶点相邻的顶点。任取 $f(u_1)=v_6$ 。(如果发现这个选择得不出同构，就接着试验 $f(u_1)=v_1$ 。)因为 u_2 与 u_1 相邻，所以 u_2 可能的像是 v_3 和 v_5 。任取 $f(u_2)=v_3$ 。照这样继续下去，用顶点的相邻关系和度作为指引，令 $f(u_3)=v_4$ ， $f(u_4)=v_5$ ， $f(u_5)=v_1$ ，以及 $f(u_6)=v_2$ 。现在已经有了在 G 的顶点集与 H 的顶点集之间的一一对应，即 $f(u_1)=v_6$ ， $f(u_2)=v_3$ ， $f(u_3)=v_4$ ， $f(u_4)=v_5$ ， $f(u_5)=v_1$ ，以及 $f(u_6)=v_2$ 。为了查看 f 是否保持边，就检查 G 的邻接矩阵：

$$A_G = \begin{matrix} & \begin{matrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \end{matrix} \\ \begin{matrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

以及用 G 里的对应顶点的像来标记行和列的 H 的邻接矩阵：

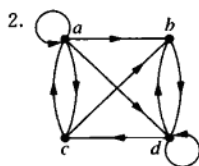
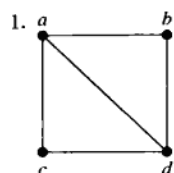
$$A_H = \begin{matrix} & \begin{matrix} v_6 & v_3 & v_4 & v_5 & v_1 & v_2 \end{matrix} \\ \begin{matrix} v_6 \\ v_3 \\ v_4 \\ v_5 \\ v_1 \\ v_2 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

因为 $A_G=A_H$ ，所以 f 是保持边的。由此得出 f 是一个同构，所以 G 与 H 是同构的。注意，若事实证明 f 不是一个同构，是无法得出 G 与 H 不是同构的，因为 G 和 H 里的顶点的另一个对应仍然可以是同构。

判定两个图是否同构，已知的最好算法具有指数的最坏情形时间复杂性(对图的顶点数来说)。不过，解决这个问题的线性平均情形时间复杂性的算法已经找到，而且有希望找到判定两个图是否同构的多项式最坏情形时间复杂性的算法。一种名为 NAUTY 的最佳实用算法，在现代个人电脑上可在 1 秒钟之内判定带有 100 个顶点的两个图是否同构。可以在因特网上下载 NAUTY 软件并用它做实验。

练习

在练习 1~2 中, 用邻接表表示给定的图。



3. 用邻接矩阵表示练习 1 里的图。

4. 用邻接矩阵表示练习 2 里的图。

5. 用邻接矩阵表示下列每一个图。

a) K_4

b) $K_{1,4}$

c) $K_{2,3}$

d) C_4

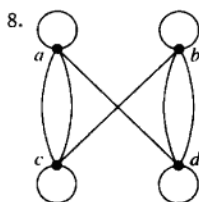
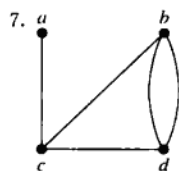
e) W_4

f) Q_3

6. 画出用下面的邻接矩阵表示的图。

$$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

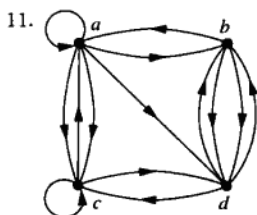
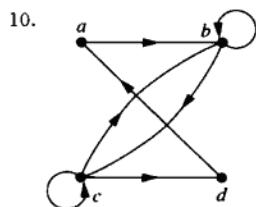
在练习 7~8 中, 用邻接矩阵表示给定的图。



9. 画出用下面的邻接矩阵所表示的无向图。

$$\begin{bmatrix} 1 & 2 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

在练习 10~11 中, 求给定的有向图的邻接矩阵。



12. 画出用下面的邻接矩阵表示的图。

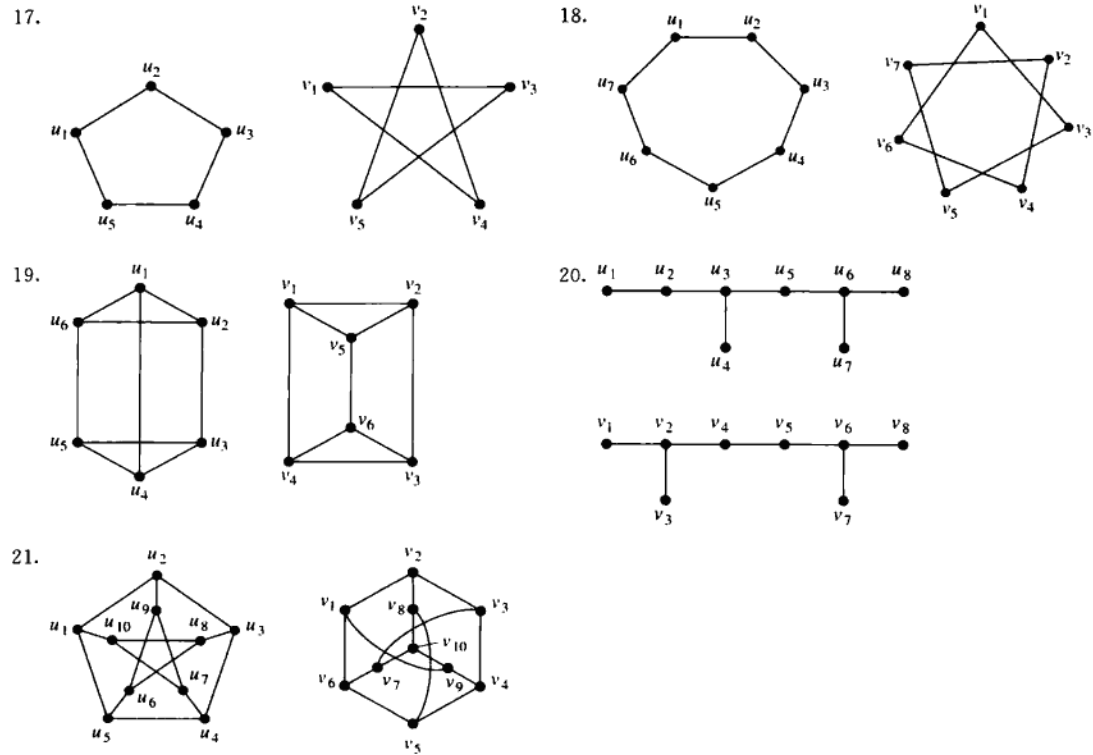
$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 0 \\ 0 & 2 & 2 \end{bmatrix}$$

13. 每一个对称的和对角线全为 0 的 0-1 方阵是否都是简单图的邻接矩阵?

14. 用关联矩阵表示练习 7~8 中的图。

- *15. 什么是无向图的邻接矩阵的一列里的各项之和？对有向图来说呢？
- 16. 什么是无向图的关联矩阵的一列里的各项之和？对有向图来说呢？

在练习 17~21 中，判定所给定的一对图是否同构。给出一个同构或给出不存在同构的严格证明。



- 22. 证明：简单图的同构关系是等价关系。
- 23. 描述一下图的邻接矩阵里对应于孤立点的行和列。
- 24. 证明：带有 2 个以上顶点的偶图的顶点可以排序，使得邻接矩阵形如

$$\begin{bmatrix} 0 & A \\ B & 0 \end{bmatrix}$$

其中所示的四项都是矩形块。

若简单图 G 和 $\bar{G}$ 是同构的，则 G 称为自补图。

- 25. 求带有 5 个顶点的自补简单图。
- 26. 对哪些整数 n 来说 C_n 是自补图？
- 27. 带有 5 个顶点和 3 条边的非同构的简单图有多少个？
- 28. 带有下列邻接矩阵的简单图是否同构？

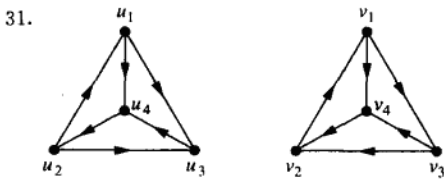
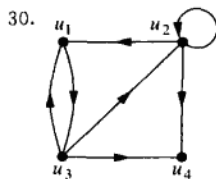
a) $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

b) $\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

c) $\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

29. 把简单图的同构定义推广到包含环和多重边的无向图。

在练习 30~31 里, 判定所给定的一对图是否同构。



32. 证明: 若 G 和 H 是同构的有向图, 则 G 和 H 的逆(在 9.2 节练习 61 之前定义)也是同构的。

33. 找出一对非同构的图, 它们具有相同的度序列, 但一个是偶图而另一个不是。

*34. 无向图的关联矩阵与它的转置之积是什么?

6.4 连通性

6.4.1 引言

许多问题可以用沿图的边旅行所形成的通路来建模。例如, 判定在两个计算机之间用中间连接能否传递消息的问题, 就可以用图模型来研究。利用图里的通路所组成的模型, 可以解决投递邮件、收取垃圾以及计算机网络诊断等有效规划路线的问题。

6.4.2 通路

非形式化地说, 通路是边的序列, 它从图的一个顶点上开始并且行经图的边, 总是连接相邻的顶点对。定义 1 给出通路的形式化定义和相关术语。

定义 1 设 n 是非负整数且 G 是无向图。在 G 中从 u 到 v 的长度为 n 的通路是 G 的 n 条边 $e_1, \dots, e_n$ 的序列, 使得 $f(e_1) = \{x_0, x_1\}$, $f(e_2) = \{x_1, x_2\}$, $\dots$, $f(e_n) = \{x_{n-1}, x_n\}$, 其中 $x_0 = u$ 而 $x_n = v$ 。当这个图是简单图时, 就用顶点序列 $x_0, x_1, \dots, x_n$ 表示这条通路(这是因为列出这些顶点就唯一地确定了通路)。若一条通路在相同的顶点上开始和结束, 即 $u = v$ 且长度大于 0, 则它是一条回路。把这条通路或回路说成是经过顶点 $x_1, x_2, \dots, x_{n-1}$ 或遍历边 $e_1, e_2, \dots, e_n$ 。若通路或回路不重复地包含相同的边, 则它是简单的。

当没有必要区分多重边时, 就用顶点序列 $x_0, x_1, \dots, x_n$ 表示通路 $e_1, e_2, \dots, e_n$, 其中对 $i = 1, 2, \dots, n$ 来说, $f(e_i) = \{x_{i-1}, x_i\}$ 。这种记法仅仅指出通路所经过的顶点。经过这个顶点序列可以有多条通路。注意长度为 0 的通路由单个顶点组成。

注意 关于定义 1 中的概念, 有很多不同的术语。例如在有些书里, 使用路径(walk)而不是通路(path), 这时路径被定义为图的顶点和边相互交替的序列, $v_0, e_1, v_1, e_2, \dots, v_{n-1}, e_n, v_n$, 其中 v_{i-1} 和 v_i 是 e_i 的端点, $i = 1, 2, 3, \dots, n$ 。当使用“路径”这个术语时, 就会使用“闭合路径”(closed walk)而不是回路(circuit)来表示起始和终止于相同顶点的路径; 使用路线(trail)来表示没有重复边的路径(代替“简单通路”)。当使用路线这一术语时, 术语通路(path)通常就会用来表示没有重复顶点的路线, 这与定义 1 中的术语相冲突。由于这些术语上的各种变体, 当你在特定的书或者文章中阅读有关遍历图的边的内容时, 需要弄清楚使用的是哪一组定义。文章[GrYe06]是关于本注释中提到的各替代术语的一本好的参考文献。

例 1 在图 6-40 所示的图里, a, d, c, f, e 是长度为 4 的简单通路, 因为 $\{a, d\}, \{d, c\}, \{c, f\}$ 和 $\{f, e\}$ 都是边。但是 d, e, c, a 不是通路, 因为 $\{e, c\}$ 不是边。注意 b, c, f, e, b 是长度为 4 的回路, 这是因为 $\{b, c\}, \{c, f\}, \{f, e\}$ 和 $\{e, b\}$ 都是边, 这条通路在 b 上开始和结束。长度为 5 的通路 a, b, e, d, a, b 不是简单的, 因为它两次包含边 $\{a, b\}$ 。■

有向图里的通路和回路在第 5 章里介绍过。现在对有向多重图来定义这样的通路。

定义 2 设 n 是非负整数且 G 是有向多重图。在 G 中从 u 到 v 的长度为 n 的通路是 G 的边的序列 $e_1, e_2, \dots, e_n$, 使得 $f(e_1) = (x_0, x_1), f(e_2) = (x_1, x_2), \dots, f(e_n) = (x_{n-1}, x_n)$, 其中 $x_0 = u$ 而 $x_n = v$ 。当有向图里没有多重边时, 就用顶点序列 $x_0, x_1, \dots, x_n$ 表示这条通路。把在相同的顶点上开始和结束的长度大于 0 的通路称为回路或圈。若一条通路或回路不重复地包含相同的边, 则把它称为简单的。

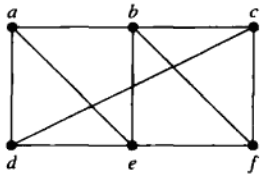


图 6-40 简单图

注意 由定义 2 给出的术语并非经常用来表示其中描述的概念。特别地, 使用路径(walk)、闭合路径(closed walk)、路线(trail)、通路(path)的术语(在定义 1 之后的注解中介绍过)可能用来描述有向图。详见[GrYe06]。

注意, 通路上一条边的终点是这条通路上下一条边的起点。当没有必要区分多重边时, 就用顶点序列 $x_0, x_1, \dots, x_n$ 表示通路 $e_1, e_2, \dots, e_n$, 其中对 $i = 1, 2, \dots, n$ 来说, $f(e_i) = (x_{i-1}, x_i)$ 。这种记法仅仅指出通路所经过的顶点。可以有多条通路经过这个顶点序列。

通路表示许多图模型里的有用信息, 如例 2~4 所示。

例 2 熟人关系图中的通路 在熟人关系图中, 如果相邻两个人彼此认识则将此二人连接起来, 形成一条人链。例如在 6.1 节的图 6-7 中, 有一条连接 Kamini 和 Ching 的 6 个人的链。许多社会学家猜想, 是否可以用只包含 5 个或更少的人的短链来连接世界上几乎每一对人。这意味着世界上所有人的熟人关系图中几乎每对顶点都通过长度不超过 4 的通路来连接。约翰·奎尔(John Guare)的戏剧《六度分离》(*Six Degrees of Separation*)就是基于这个概念。

例 3 合作图中的通路 在合作图中, 如果表示作者的两个顶点 a 和 b 之间有从 a 开始到 b 结束的一系列作者, 使得每条边端点所表示的两个作者写过一篇联名论文, 则 a 和 b 就通过一条通路而连接。在所有数学家的合作图中, 数学家 m 的埃德斯数(在第 5 章补充练习 14 中用关系术语定义过)就是在 m 和代表成果极其丰富的数学家保罗·埃德斯(1996 年去世)的顶点之间的最短通路的长度。换句话说, 一个数学家的埃德斯数就是从保罗·埃德斯开始到这个数学家结束的最短的数学家链的长度, 其中每一对相邻的数学家都联名写过论文。根据“埃德斯数项目”, 2006 年具有各个埃德斯数的数学家的数目如表 6-4 所示。

表 6-4 具有给定埃德斯数的数学家的数目

埃德斯数	人 数	埃德斯数	人 数
0	1	7	11 591
1	504	8	3 146
2	6 593	9	819
3	33 605	10	244
4	83 642	11	68
5	87 760	12	23
6	40 014	13	5

例 4 好莱坞图中的通路 在好莱坞图(参见 6.1 节例 4)中, 当存在连接两个顶点 a 和 b 的演员链, 其中在这个链上每两个相邻的演员都出演过同一部电影时, a 和 b 就被连接。在好莱坞图中演员 c 的培根数定义为连接 c 和著名演员凯文·培根的最短通路的长度。随着新电影(包括凯文·培根的新电影)的不断产生, 演员们的培根数也在不断地发生变化。表 6-5 中显示的是从培根网站得来的 2006 年具有各个培根数的演员的数目。

表 6-5 具有给定培根数的演员数

培根数	人 数	培根数	人 数
0	1	5	8 168
1	1 902	6	810
2	160 463	7	81
3	457 231	8	14
4	111 310		

6.4.3 无向图的连通性

若消息通过一个或多个中间计算机来传递,则计算机网络何时具有每对计算机都可共享信息的性质?当利用图来表示这个计算机网络,用顶点表示计算机而用边表示通信连接时,这个问题就变成:何时在图里任何两个顶点之间都存在通路?

定义 3 若无向图每一对不同的顶点之间都有通路,则该图称为连通的。

因此,在网络里任何两个计算机都可以通信,当且仅当这个网络图是连通的。

例 5 图 6-41 里的图 G_1 是连通的,这是因为在每一对不同的顶点之间都有通路(读者应当验证它)。但是图 6-41 里的图 G_2 不是连通的。例如,在顶点 a 和 d 之间没有通路。

在第 7 章里将用到下述定理。

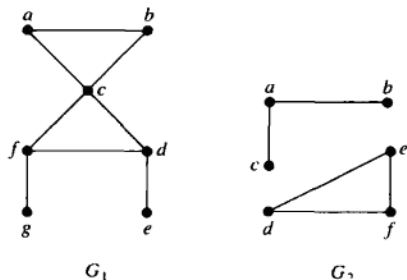
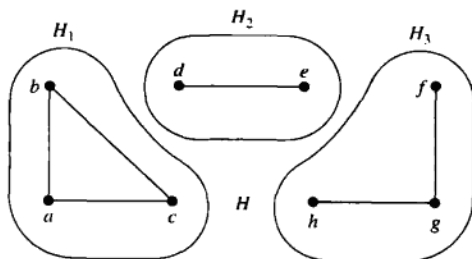
定理 1 在连通无向图的每一对不同顶点之间都存在简单通路。

证 设 u 和 v 是连通无向图 $G=(V, E)$ 的两个不同的顶点。因为 G 是连通的,所以 u 和 v 之间至少有 1 条通路。设 $x_0, x_1, \dots, x_n$ 是长度最短的通路的顶点序列,其中 $x_0=u$ 而 $x_n=v$ 。这条长度最短的通路是简单的。为了看明白这一点,假设它不是简单的。则对满足 $0 \leq i < j$ 的某个 i 和 j 来说,有 $x_i=x_j$ 。这意味着通过删除顶点序列 $x_i, \dots, x_{j-1}$ 所对应的边,就获得了带有顶点序列 $x_0, x_1, \dots, x_{i-1}, x_j, \dots, x_n$ 的从 u 到 v 的更短的通路。□

不连通的图是 2 个或 2 个以上连通子图之并,每一对子图都没有公共的顶点。这些不相交的连通子图称为图的连通分支。

例 6 图 6-42 所示的图 H 的连通分支是什么?

解 图 H 是图 6-42 所示的 3 个不相交连通子图 H_1 、 H_2 和 H_3 之并。这 3 个子图是 H 的连通分支。

图 6-41 图 G_1 和 G_2 图 6-42 图 H 和它的连通分支 H_1 、 H_2 和 H_3

例 7 呼叫图的连通分支 当电话呼叫图(参见 6.1 节例 7)中存在一系列从 x 开始到 y 结束的电话呼叫时,两个顶点 x 和 y 就属于同一个连通分支。当分析 AT&T 网络中特定一天内发生的电话呼叫的呼叫图时,发现这个图具有 53 767 087 个顶点、超过 1 亿 7 千万条边和超过 370 万个连通分支。这些连通分支大多数都很小:大约四分之三是由表示只在彼此之间呼叫的一对电话号码的两个顶点所组成。这个图具有一个包含 44 989 297 个顶点(占总数的 80%)的巨

大的连通分支。另外，这个连通分支中每个顶点都通过一条不超过 20 个顶点的链连接到任何其他顶点。

有时删除一个顶点和它所关联的边，就产生带有比原图更多的连通分支的子图。把这样的顶点称为割点(或节点)。从连通图里删除割点，就产生不连通的子图。同理，把一旦删除就产生带有比原图更多的连通分支的子图的边称为割边或桥。

例 8 求出图 6-43 所示的图 G 的割点和割边。

解 图 G 的割点是 b 、 c 和 e 。删除这些顶点中的一个(和它的邻边)，就使得这个图不再是连通的。割边是 $\{a, b\}$ 和 $\{c, e\}$ 。删除这些边中的一条，就使得 G 不再是连通的。

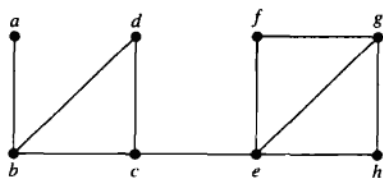


图 6-43 图 G

6.4.4 有向图的连通性

根据是否考虑边的方向，在有向图里有两种连通性概念。

定义 4 若每当 a 和 b 都是一个有向图里的顶点时，就有从 a 到 b 和从 b 到 a 的通路，则该图是强连通的。

为了让一个有向图是强连通的，从这个图里任何一个顶点到任何另外一个顶点必须存在有向边的序列。有向图可以不是强连通的但还是“一整块”。定义 5 准确地说明了这个概念。

定义 5 若在有向图的底图里，任何两个顶点之间都有通路，则该有向图是弱连通的。

有向图是弱连通的，当且仅当在忽略边的方向时，任何两个顶点之间总是存在通路。显然，任何强连通有向图也是弱连通的。

例 9 图 6-44 所示的有向图 G 和 H 是否为强连通的？是否为弱连通的？

解 G 是强连通的，因为在这个有向图里，任何两个顶点之间都存在通路(读者应当验证它)。因此 G 也是弱连通的。图 H 不是强连通的，在这个图里，从 a 到 b 没有有向通路。但是 H 是弱连通的，因为在 H 的底图里，任何两个顶点之间都有通路(读者应当验证它)。

有向图 G 的子图是强连通的而不包含在更大的强连通子图中，即极大强连通子图，可称之为 G 的强连通分支或强分支。

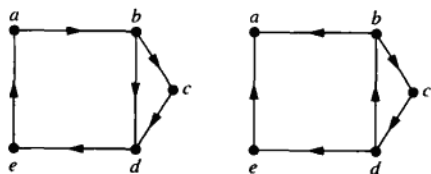


图 6-44 有向图 G 和 H

例 10 图 6-44 中的图 H 有 3 个强连通分支，包括：顶点 a ；顶点 e ；由顶点 b 、 c 和 d 以及边 (b, c) 、 (c, d) 和 (d, b) 所组成的图。

例 11 网络图的强连通分支 6.1 节例 8 介绍的网络图用顶点表示网页并且用有向边表示链接。网络在 1999 年的快照产生了具有 2 亿个顶点和 15 亿条边的网络图(参见 [Br00] 了解详情)。网络图的底无向图不是连通的并且有一个连通分支包含了这个图中大约 90% 的顶点。底无向图的这个连通分支所对应的原来有向图的子图(即具有相同的顶点以及连接这个图中顶点的所有有向边)有一个非常大的强连通分支和许多小的强连通分支。前者称为这个有向图的巨型强连通分支。可以遵循从这个分支里任何其他网页开始的链接来到达这个分支里的某一个网页。已经发现这项研究产生的网络图里的巨型强连通分支有超过 5300 万个顶点。这个无向图的大型连通分支里的其余顶点表示 3 种不同类型的网页：可以从巨型强连通分支里的网页来到达，但是不能通过一系列链接来返回前面这些网页的网页；可以通过一系列链接来返回巨型强连通分支里的网页，但是不能通过巨型强连通分支里网页上的链接来到达的网页；既不能到达巨型强连通分支里的网页，也不能通过一系列链接从巨型强连通分支里网页来到达的网页。在这项研究里，发现其余这三个集合中每个都具有大约 4400 万个顶点(这三个集合都接近同样的规模，这是相当令人惊讶的)。

6.4.5 通路与同构

有多种方式可以利用通路和回路来帮助判定两个图是否同构。例如, 具有特定长度的简单回路的存在性, 就是一种可以用来证明两个图是不同构的有用的不变量。另外, 可以利用通路来构造可能的同构映射。

前面指出过, 简单图的一个有用的同构不变量是长度为 k 的简单回路的存在性, 其中 k 是大于 2 的正整数。例 12 说明如何用这个不变量来证明两个图不同构。

例 12 判定图 6-45 所示的图 G 和 H 是否同构。

解 G 和 H 都具有 6 个顶点和 8 条边。各自具有 4 个 3 度顶点和 2 个 2 度顶点。所以对两个图来说有 3 个不变量(顶点数、边数以及顶点度)都是相同的。但是 H 有长度为 3 的简单回路, 即 v_1, v_2, v_6, v_1 , 而 G 没有长度为 3 的简单回路(G 里的所有简单回路的长度至少为 4)。因为长度为 3 的简单回路的存在性是一个同构不变量, 所以 G 和 H 不同构。■

我们已经说明了如何用某种类型的通路, 即具有特定长度的简单回路, 来证明两个图是不同构的。还可以用通路来求出潜在的同构映射。

例 13 判定图 6-46 所示的图 G 和 H 是否同构。

解 G 和 H 都具有 5 个顶点和 6 条边, 都具有 2 个 3 度顶点和 3 个 2 度顶点, 都具有 1 个长度为 3 的简单回路, 1 个长度为 4 的简单回路, 以及 1 个长度为 5 的简单回路。因为所有这些同构不变量都是相同的, 所以 G 和 H 可能是同构的。为了求出可能的同构, 沿着经过所有顶点并且使得两图里的对应顶点的度都相同的通路前进。例如, G 里的通路 u_1, u_4, u_3, u_2, u_5 和 H 里的通路 v_3, v_2, v_1, v_5, v_4 都经过图里的每一个顶点, 都在 3 度顶点上开始, 都分别经过 2 度顶点和 2 度顶点并且在 2 度顶点上结束。通过在图里沿着这些通路前进, 定义映射 f 满足 $f(u_1) = v_3, f(u_4) = v_2, f(u_3) = v_1, f(u_2) = v_5$ 和 $f(u_5) = v_4$ 。通过说明 f 保持边, 或者通过说明在顶点的适当顺序下 G 和 H 的邻接矩阵是相同的, 读者就可以说明 f 是一个同构, 所以 G 与 H 是同构的。■

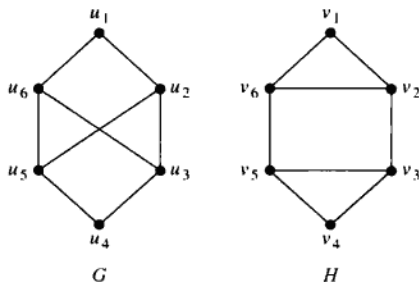


图 6-45 图 G 和 H

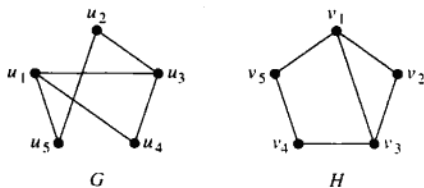


图 6-46 图 G 和 H

6.4.6 计算顶点之间的通路数

在一个图里两个顶点之间的通路的数目, 可以用这个图的邻接矩阵来确定。

定理 2 设 G 是带有相对于顶点顺序 $v_1, v_2, \dots, v_n$ 的邻接矩阵 A 的图(允许带有无向或有向边、带有多重边和环)。从 v_i 到 v_j 的长度为 r 的不同通路的数目等于 A^r 的第 (i, j) 项, 其中 r 是正整数。

证 用数学归纳法证明。设 G 是带有邻接矩阵 A 的图(假设 G 的顶点具有顺序 $v_1, v_2, \dots, v_n$)。从 v_i 到 v_j 长度为 1 的通路数是 A 的第 (i, j) 项, 这是因为该项是从 v_i 到 v_j 的边数。

假设 A^r 的第 (i, j) 项是从 v_i 到 v_j 的长度为 r 的不同通路的个数。这是归纳假设。因为 $A^{r+1} = A^r A$, 所以 A^{r+1} 的第 (i, j) 项等于

$$b_{i1}a_{1j} + b_{i2}a_{2j} + \dots + b_{in}a_{nj}$$

其中 b_{ik} 是 A^r 的第 (i, k) 项。根据归纳假设, b_{ik} 是从 v_i 到 v_k 的长度为 r 的通路数。

从 v_i 到 v_j 的长度为 $r+1$ 的通路, 包括从 v_i 到某个中间顶点 v_k 的长度为 r 的通路以及从 v_k 到 v_j 的边。根据计数的乘法规则, 这样的通路的个数是从 v_i 到 v_k 的长度为 r 的通路数(即 b_{ik})与从 v_k 到 v_j 的边数(即 a_{kj})之积。当对所有可能的中间顶点 v_k 求这些乘积之和时, 根据计数的加法规则,

就得出所需要的结果。

例 14 在图 6-47 所示的简单图 G 里, 从 a 到 d 的长度为 4 的通路有多少条?

解 G 的邻接矩阵(顶点顺序为 a, b, c, d)是

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

因此从 a 到 d 的长度为 4 的通路数是 A^4 的第(1, 4)项。因为

$$A^4 = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix}$$

所以恰好有 8 条从 a 到 d 的、长度为 4 的通路。通过检查这个图, 我们看出 a, b, a, b, d ; a, b, a, c, d ; a, b, d, b, d ; a, b, d, c, d ; a, c, a, b, d ; a, c, a, c, d ; a, c, d, b, d 和 a, c, d, c, d 是 8 条从 a 到 d 的通路。

定理 2 可以用来求出在图的两个顶点之间的最短通路的长度, 还可以用来判定图是否连通(见练习 25)。

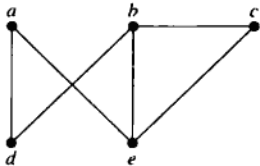
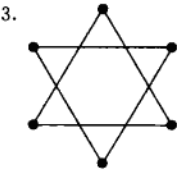
练习

1. 下述的每个顶点列表是否可以构成右图里的通路? 哪些通路是简单的?

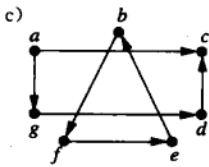
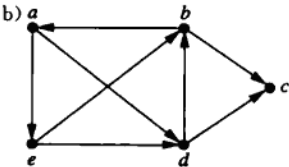
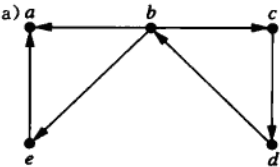
哪些是回路? 那些通路的长度是多少?

- a) a, e, b, c, b
- b) a, e, a, d, b, c, a
- c) e, b, a, d, b, e
- d) c, b, d, a, e, c

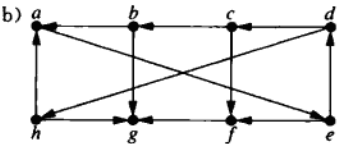
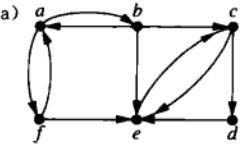
在练习 2~3 里, 判定所给的图是否连通。

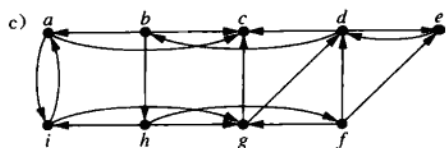


- 4. 熟人关系图的连通分支表示什么?
- 5. 解释一下为什么在数学家的合作图里, 表示一个数学家的顶点与表示保罗·埃德斯的顶点是在同一个连通分支里, 当且仅当这个数学家具有有穷的埃德斯数。
- 6. 判断下列各图是否是强连通的, 如果不是, 再判断是否是弱连通的。



- 7. 电话呼叫图的强连通分支表示什么?
- 8. 求下列每个图的强连通分支。

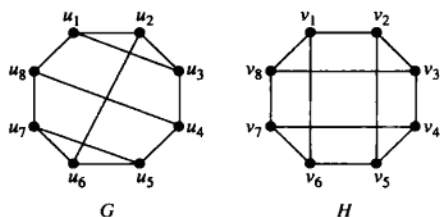




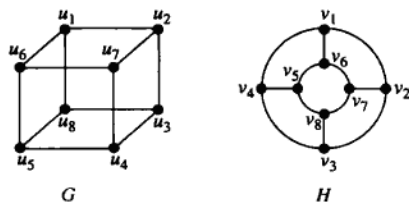
9. 求出 K_4 里两个不同顶点之间长度为 n 的通路数目, 若 n 是

- a) 2 b) 3 c) 4 d) 5

10. 运用通路要么证明这两个图不是同构的, 要么找出这两图之间的一个同构。



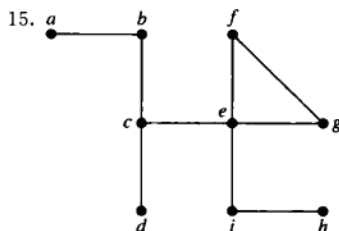
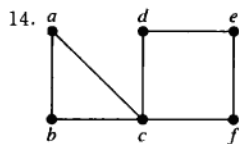
11. 运用通路要么证明这两个图不是同构的, 要么找出这两图之间的一个同构。



12. 对练习 9 里的 n 值来说, 求出 $K_{3,3}$ 里任意两个不相邻顶点之间长度为 n 的通路数目。

13. 设 $G=(V, E)$ 是简单图。设 R 是 V 上的关系, 它是由这样的顶点对 (u, v) 所组成的, 使得存在从 u 到 v 的通路或使得 $u=v$ 。证明: R 是一个等价关系。

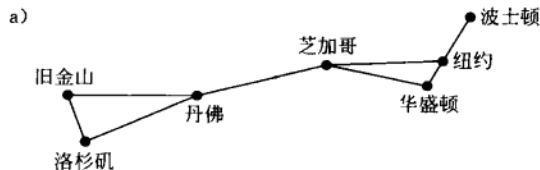
在练习 14~15 里, 求所给图的所有割点。

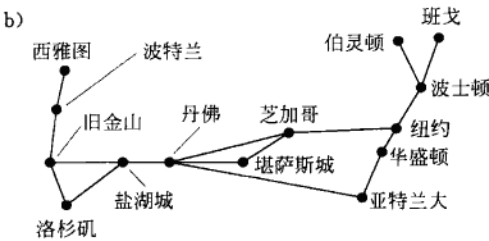


*16. 假设 v 是一条割边的端点。证明: v 是割点当且仅当它不是悬挂点。

*17. 证明: 在至少有 2 个顶点的简单图中, 至少有 2 个顶点不是割点。

18. 若网络里的通信连接故障会导致不能传送某些消息, 则应当提供备份连接。对下面(a)和(b)里所示的通信网络来说, 确定出那些应当有备份的连接。





一个有向图里的顶点基是顶点的一个集合，使得该有向图里该集合之外的任何顶点，都有从该集合里某个顶点开始的一条通路，并且该集合里任何两个顶点之间都没有通路。

19. 在影响图(在 6.1 节例 3 里描述)里顶点基的重要性是什么？找出在该例里的影响图的顶点基。

*20. 证明：若简单图 G 有 k 个连通分支，而且这些分支分别具有 $n_1, n_2, \dots, n_k$ 个顶点，则 G 的边数不超过

$$\sum_{i=1}^k C(n_i, 2)$$

*21. 证明：若带有 n 个顶点的简单图 G 具有超过 $(n-1)(n-2)/2$ 条边，则它是连通的。

22. 当 n 取如下值时，存在多少个不同构的、带有 n 个顶点的连通简单图？

a) 2

b) 3

c) 4

d) 5

23. 用定理 2 求出图 6-40 里的多重图从 a 到 f 的最短通路的长度。

24. 设 P_1 和 P_2 是简单图 G 里顶点 u 和 v 之间的没有相同边的两条简单通路。证明：在 G 里存在简单回路。

25. 解释一下如何用定理 2 判定图是否连通。

26. 证明：简单图 G 是偶图，当且仅当 G 没有包含奇数条边的回路。

6.5 欧拉通路与哈密顿通路

6.5.1 引言

能否从一个顶点出发沿着图的边前进，恰好经过图的每条边一次并且回到这个顶点？同样，能否从一个顶点出发沿着图的边前进，恰好经过图的每个顶点一次并且回到这个顶点？虽然这两个问题有相似之处，但是对于所有的图来说，可以轻而易举地回答第一个关于是否具有欧拉回路的问题，却非常难以解决第二个关于是否具有哈密顿回路的问题。本节将研究这些问题并且讨论这些问题的难点。虽然两个问题在许多不同领域里都有实际应用，但是都来源于古老的智力题。下面将介绍这些古老的智力题以及现代的实际应用。

6.5.2 欧拉通路与欧拉回路

普鲁士的哥尼斯堡镇(现名加里宁格勒，属于俄罗斯共和国)被普雷格尔河支流分成四部分。这四部分包括河两岸、河中心岛以及两条支流之间的部分。在 18 世纪这四部分用七座桥连接起来，如图 6-48 所示。

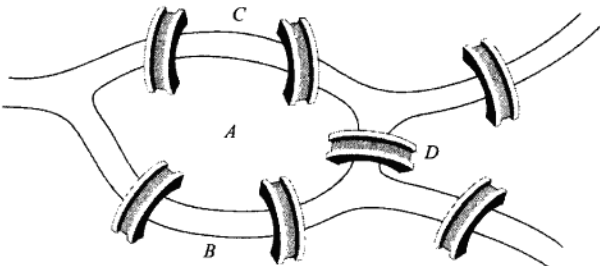


图 6-48 哥尼斯堡七桥

镇上的人们在周日里穿过镇子进行长距离的散步。他们想弄明白是否可能从镇里某个位置出发不重复地经过所有桥并且返回出发点。

瑞士数学家列昂哈德·欧拉解决了这个问题。他的解答发表在1736年，这也许是人们第一次使用图论。欧拉利用多重图来研究这个问题，其中用顶点表示四个部分，用边表示桥，如图6-49所示。

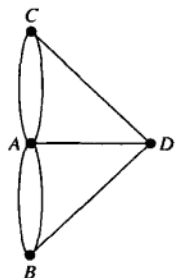


图 6-49 哥尼斯堡镇的多重图模型

不重复地经过每一座桥来旅行的问题可以利用这个模型来重新叙述。问题变成：在这个多重图里是否存在包含每一条边的简单回路？

定义 1 图 G 里的欧拉回路是包含着 G 的每一条边的简单回路。图 G 里的欧拉通路是包含着 G 的每一条边的简单通路。

例 1 和例 2 解释了欧拉回路和欧拉通路的概念。

例 1 在图 6-50 里，哪些无向图具有欧拉回路？在没有欧拉回路的那些图里，哪些具有欧拉通路？

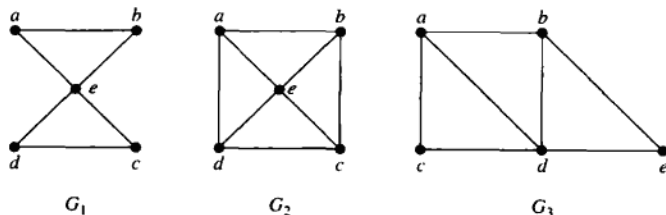


图 6-50 无向图 G_1 , G_2 和 G_3

解 图 G_1 具有欧拉回路，例如 a, e, c, d, e, b, a 。图 G_2 和 G_3 都没有欧拉回路（读者应当验证它）。但是 G_3 具有欧拉通路，即 a, c, d, e, b, d, a, b 。图 G_2 没有欧拉通路（读者应当验证它）。 ■

例 2 在图 6-51 里，哪些有向图具有欧拉回路？在没有欧拉回路的那些图里，哪些具有欧拉通路？

解 图 H_2 具有欧拉回路，例如 $a, g, c, b, g, e, d, f, a$ 。图 H_1 和 H_3 都没有欧拉回路（读者应当验证它）。图 H_3 具有欧拉通路，即 c, a, b, c, d, b ，但是 H_1 没有欧拉通路（读者应当验证它）。 ■

欧拉回路和欧拉通路的充要条件 对判断多重图是否具有欧拉回路和欧拉通路来说，存在着简单的标准。欧拉在解决著名的哥尼斯堡七桥问题时发现了它们。假设在本节里讨论的所有图都具有无穷多个顶点和边。

若一个连通多重图具有欧拉回路，则它有什么性质呢？可以说明的是：每一个顶点都必有偶数条边。为此，首先注意一条欧拉回路从顶点 a 开始，接着是 a 关联的一条边，比方说 $\{a, b\}$ 。边 $\{a, b\}$ 为 $\deg(a)$ 贡献 1 度。这条回路每次经过一个顶点就为该顶点的度贡献 2 度，这是因为这条回路经过关联该顶点的边进入，又经过另一条这样的边离开。最后，这条回路在它开始的地方结束，为 $\deg(a)$ 贡献 1 度。因此 $\deg(a)$ 必为偶数，

这是因为当回路开始时它贡献 1 度，当回路结束时它贡献 1 度，每次经过 a 都贡献 2 度（如果它又经过了 a ）。除 a 外的其余顶点都有偶数度，这是因为每次回路经过一个顶点就为该顶点的度贡献 2 度。由此得出结论若连通图有欧拉回路，则每一个顶点必有偶数度。

欧拉回路的存在性的这个必要条件是否也是充分的？即若在连通多重图里所有顶点都有偶数

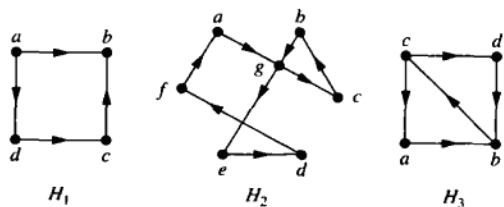


图 6-51 有向图 H_1 , H_2 和 H_3

度, 则是否必有欧拉回路? 这个问题可以通过构造来解决。

假设 G 是连通多重图而且 G 的每一个顶点都有偶数度。构造从 G 的任意顶点 a 开始的简单回路。设 $x_0 = a$ 。首先任意地选择一条关联 a 的边 $\{x_0, x_1\}$ 。通过建立尽量长的简单通路 $\{x_0, x_1\}$, $\{x_1, x_2\}$, $\dots$, $\{x_{n-1}, x_n\}$ 来继续构造。例如, 在图 6-52 的图 G 里, 从 a 开始而且连续地选择边 $\{a, f\}$, $\{f, c\}$, $\{c, b\}$ 和 $\{b, a\}$ 。

这样的通路必然结束, 这是因为图的边数是有穷的。它在 a 上以形如 $\{a, x\}$ 的边开始, 而且在 a 上以形如 $\{y, a\}$ 的边结束。这是因为通路每次经过一个偶数度顶点时, 它只用 1 条边进入这个顶点, 所以至少还剩下 1 条边让通路离开这个顶点。这条通路可能用完了所有的边, 也可能没用完。

若所有的边都已经用完, 则欧拉回路已经构造好了。否则, 考虑通过从 G 里删除已经用过的边和不关联任何剩余的边的顶点这样所得到的子图 H 。当从图 6-52 里的图删除回路 a, f, c, b, a 时, 就得到标记为 H 的子图。

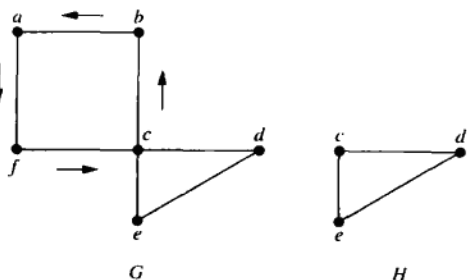


图 6-52 构造 G 里的欧拉回路

因为 G 是连通的, 所以 H 与已经删除的回路至少具有 1 个公共顶点。设 w 是这样的顶点(此例中 c 是这个顶点)。

H 里的每一个顶点都有偶数度。(因为 G 里的所有顶点都有偶数度, 对每个顶点来说, 把与这个顶点关联的边成对地删除, 以便形成 H 。)注意 H 可能是不连通的。像在 G 里做过的那样, 在 w 上开始, 通过尽可能地选择边来构造 H 里的简单回路。这条回路必然在 w 上结束。例如, 在图 6-52 里, c, d, e, c 是 H 里的回路。下一步通过把 H 里的回路与 G 里原来的回路拼接起来形成 G 里的回路(这是可行的, 因为 w 是这个回路的顶点之一)。当在图 6-52 的图里这样做时, 就得到回路 a, f, c, d, e, c, b, a 。

继续进行这个过程, 直到已经用完了所有的边为止(这个过程必然结束, 这是因为图里只有有穷的边数)。这样就产生出欧拉回路。这样的构造说明, 若连通多重图的顶点都有偶数度, 则该图具有欧拉回路。

把这些结果总结成定理 1。

定理 1 连通多重图具有欧拉回路当且仅当它的每个顶点都有偶数度。

现在可以解决哥尼斯堡七桥问题了。因为图 6-49 所示的表示这些桥的多重图具有 4 个奇数度顶点, 所以它没有欧拉回路。从给定点开始、恰好经过每座桥一次并返回开始点是无法实现的。

算法 1 给出了在定理 1 之前的讨论里所给的求欧拉回路的构造过程。(因为这个过程里的回路是任意地选择的, 所以存在一些不确定性。这里不再通过更精确地说明过程的步骤来消除这些不确定性。)

算法 1 构造欧拉回路

procedure Euler(G : 所有顶点有偶数度的连通多重图)

$circuit$:= 在 G 里任选的顶点开始, 连续地加入边所形成的回到该顶点的回路

H := 删除这条回路的边之后的 G

while H 还有边

begin

$subcircuit$:= 在既是 H 里的顶点也是 $circuit$ 的边的端点处开始的 H 里的一条回路

H := 删除 $subcircuit$ 的边和所有孤立点之后的 H

$circuit$:= 在适当顶点上插入 $subcircuit$ 之后的 $circuit$

end{ $circuit$ 是欧拉回路}

下一个例子说明如何利用欧拉通路和欧拉回路来解决一种类型的智力题。

例3 有许多智力题要求用铅笔连续移动，不离开纸面并且不重复地画出图形。利用欧拉回路和欧拉通路来解决这样的智力题。例如，能否用这样的方法画出图 6-53 所示的穆罕默德短弯刀？其中图形是在同一个顶点上开始和结束。

解 可以解决这个问题，因为图 6-53 所示的图 G 具有欧拉回路。它具有这样的回路是因为它的所有顶点都有偶数度。用算法 1 来构造欧拉回路。首先形成回路 $a, b, d, c, b, e, i, f, e, a$ 。通过删除这条回路的边并且删除因此产生的孤立点，就得到子图 H 。然后形成 H 里的回路 $d, g, h, j, i, h, k, g, f, d$ 。形成这条回路之后就用了 G 里的所有边。在适当的地方拼接这条回路和第一条回路，就产生出欧拉回路 $a, b, d, g, h, j, i, h, k, g, f, d, c, b, e, i, f, e, a$ 。这条回路给出了铅笔不离开纸面并且不重复地画出弯刀的方法。

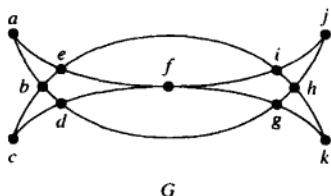


图 6-53 穆罕默德短弯刀

构造欧拉回路的另一个算法称为弗勒里算法，在本节末尾的练习里描述它。

现在说明，连通多重图具有欧拉通路（不是欧拉回路）当且仅当它恰有 2 个奇数度顶点。首先假设连通多重图有从 a 到 b 的欧拉通路，但不是欧拉回路。通路的第一条边为 a 的度贡献 1 度。通路每次经过 a 就为 a 的度贡献 2 度。通路的最后一条边为 b 的度贡献 1 度。通路每次经过 b 就为 b 的度贡献 2 度。所以 a 和 b 的度都是奇数。每一个其他顶点都具有偶数度，这是因为每当通路经过一个顶点时，就为这个顶点的度贡献 2 度。

现在反过来考虑。假设这个图恰有 2 个奇数度顶点，比方说 a 和 b 。考虑由原来的图和边 $\{a, b\}$ 所组成的更大的图。这个更大的图的每一个顶点都有偶数度，所以具有欧拉回路。删除新边就产生原图的欧拉通路。定理 2 总结了这些结果。

定理 2 连通多重图具有欧拉通路但无欧拉回路当且仅当它恰有 2 个奇数度顶点。

例 4 图 6-54 所示的哪些图具有欧拉通路？

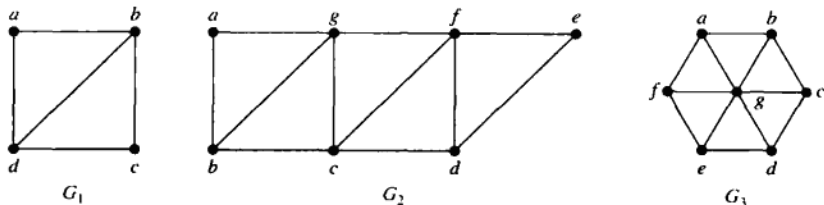


图 6-54 三个无向图

解 G_1 恰有 2 个奇数度顶点，即 b 和 d 。因此它具有必须用 b 和 d 作为端点的欧拉通路。一条这样的欧拉通路是 d, a, b, c, d, b 。同理， G_2 恰有 2 个奇数度顶点，即 b 和 d 。因此它具有必须用 b 和 d 作为端点的欧拉通路。一条这样的欧拉通路是 $b, a, g, f, e, d, c, g, b, c, f, d$ 。 G_3 没有欧拉通路，因为它具有 6 个奇数度顶点。

回到 18 世纪的哥尼斯堡，是否有可能在镇里某点开始，旅行经过所有的桥，在镇里其他某点结束？通过判定表示哥尼斯堡七桥的多重图是否具有欧拉通路，就可以回答这个问题。因为这个多重图有四个奇数度顶点，没有欧拉通路，所以这样的旅行是不可能的。

有向图里欧拉通路和欧拉回路的充要条件，在本节末尾的练习里讨论。

可以用欧拉通路和欧拉回路解决许多实际问题。例如，许多应用都要求一条通路或回路，它要恰好一次地经过一个街区里的每条街道、一个交通网里的每条道路、一个高压输电网里的每个连接或者一个通信网络里的每个链接。求出适当的图模型里的欧拉通路或欧拉回路就可以解决这样的问题。例如，如果一个邮递员可以求出表示他所负责投递的街道的图中的欧拉通路，则这条通路就产生恰好一次经过每条街道的投递路线。这个问题称为中国邮递员问题。

以纪念在1962年提出这个问题的中国科学家管梅谷。参看[MiRo91]了解关于不存在欧拉通路时中国邮递员问题的解的更多信息。

应用欧拉通路和欧拉回路的其他领域有：电路布线、网络组播和分子生物学，在分子生物学中欧拉通路用于DNA测序。

6.5.3 哈密顿通路与哈密顿回路

包含多重图每一条边恰好一次的通路和回路的存在性的充要条件已经得出。那么包含图的每一个顶点恰好一次的简单通路和回路的存在性的充要条件是否也能得出呢？

定义2 在图 $G=(V, E)$ 里，若 $V=\{x_0, x_1, \dots, x_{n-1}, x_n\}$ 并且对 $0 \leq i < j \leq n$ 来说有 $x_i \neq x_j$ ，则通路 $x_0, x_1, \dots, x_{n-1}, x_n$ 称为哈密顿通路。在图 $G=(V, E)$ 里，若 $x_0, x_1, \dots, x_{n-1}, x_n$ 是哈密顿通路，则 $x_0, x_1, \dots, x_{n-1}, x_n, x_0$ (其中 $n \geq 1$) 称为哈密顿回路。

这个术语来自爱尔兰数学家威廉·罗万·哈密顿爵士在1857年发明的智力题。哈密顿的智力题用到了木质十二面体(如图6-55a所示，十二面体有12个正五边形表面)，十二面体每个顶点上的钉子，以及细线。十二面体的20个顶点用世界上的不同城市作标记。智力题要求从一个城市开始，沿十二面体的边旅行，访问其他19个城市每个恰好一次，回到第一个城市结束。旅行经过的回路用钉子和细线来标记。

因为作者不可能向每位读者提供带钉子和细线的木质十二面体，所以考虑一个等价的问题：图6-55b中的图是否具有恰好经过每个顶点一次的回路？它就是对原题的解，因为该图同构于包含十二面体顶点和边的图。图6-56是哈密顿智力题的解。

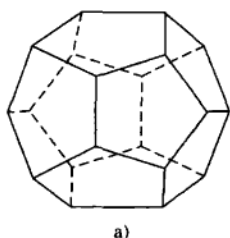


图 6-55 哈密顿的“周游世界”智力题

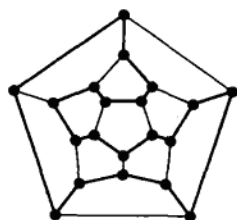
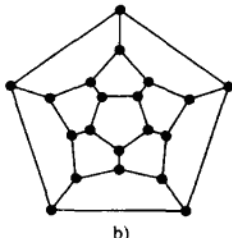


图 6-56 “周游世界”智力题的解

例5 在图6-57里，哪些简单图具有哈密顿回路？或者没有哈密顿回路但是有哈密顿通路？

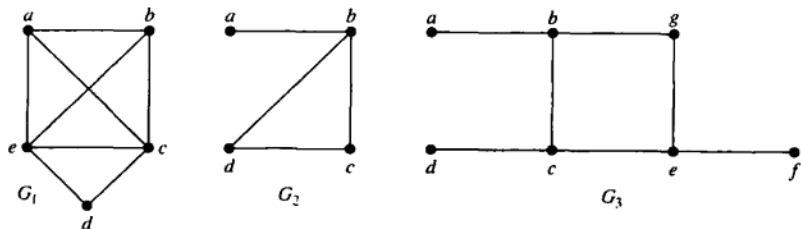


图 6-57 三个简单图

解 G_1 有哈密顿回路： a, b, c, d, e, a 。 G_2 没有哈密顿回路(从包含每一个顶点的任何回路必然两次包含边 $\{a, b\}$ 可以看出这一点)，但是 G_2 确实有哈密顿通路，即 a, b, c, d 。 G_3 既无哈密顿回路也无哈密顿通路，这是因为包含所有顶点的任何通路都必然多次包含边 $\{a, b\}$ ， $\{e, f\}$ 和 $\{c, d\}$ 其中之一。

是否存在简单方式来判定一个图有无哈密顿回路或哈密顿通路？初看起来，似乎应当有判定这一点的简单方式，因为存在简单方式来回答一个图有无欧拉回路这样的相似问题。令人吃惊的

是, 没有已知的简单的充要条件来判定哈密顿回路的存在性。不过, 已经有许多定理对哈密顿回路的存在性给出了充分条件。另外, 某些性质可以用来证明一个图没有哈密顿回路。例如, 带有 1 度顶点的图没有哈密顿回路, 因为在哈密顿回路里每个顶点都关联着回路里的两条边。另外, 若图中有 2 度顶点, 则关联这个顶点的两条边属于任意一条哈密顿回路。此外注意, 当构造哈密顿回路而且该回路经过某一个顶点时, 除了回路所用到的两条边以外, 这个顶点所关联的其他所有边不用再考虑。而且, 哈密顿回路里不能包含更小的回路。

例 6 证明图 6-58 里的图都没有哈密顿回路。

解 G 里没有哈密顿回路, 因为 G 有 1 度顶点, 即 e 。现在考虑 H 。因为顶点 a, b, d 和 e 的度都为 2, 所以这些顶点关联的每一条边都必然属于任意一条哈密顿回路。现在容易看出 H 里不存在哈密顿回路, 因为任何这样的哈密顿回路都不得不包含 4 条关联 c 的边, 这是不可能的。

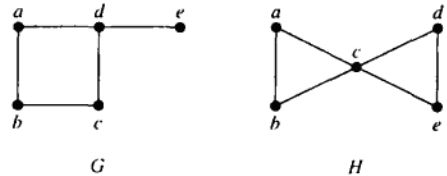


图 6-58 两个没有哈密顿回路的图

例 7 证明: 每当 $n \geq 3$ 时, K_n 就有哈密顿回路。

解 从 K_n 里的任意一个顶点开始来形成哈密顿回路。以所选择的任意顺序来访问顶点, 只要求通路在同一个顶点开始和结束, 而且对其他每个顶点恰好访问一次。这样做是可能的, 因为在 K_n 里任意两个顶点之间都有边。

虽然还未发现任何有用的关于哈密顿回路存在性的充要条件, 但很多充分条件已经找到了。注意到一个图的边越多, 这个图就越可能有哈密顿回路。另外, 加入边(而不是顶点)到已经有哈密顿回路的图中就产生有相同哈密顿回路的图。因此, 当加入边到图中时, 特别是当确保给每个顶点都加入边时, 这个图存在哈密顿回路的可能性就更大了。这里叙述充分条件中最重要的两个。这些条件是加布里尔·A·狄拉克(Gabriel A. Dirac)在 1952 年和奥伊斯坦·奥尔(Oystein Ore)在 1960 年发现的。

定理 3 狄拉克定理 如果 G 是带 n 个顶点的连通简单图, 其中 $n \geq 3$, 并且 G 中每个顶点的度都至少为 $n/2$, 则 G 有哈密顿回路。

定理 4 奥尔定理 如果 G 是带 n 个顶点的连通简单图, 其中 $n \geq 3$, 并且对于 G 中每一对不相邻的顶点 u 和 v 来说, 都有 $\deg(u) + \deg(v) \geq n$, 则 G 有哈密顿回路。

本节末的练习 27 粗略介绍了奥尔定理的证明。狄拉克定理可以作为奥尔定理的推论来证明, 因为狄拉克定理的条件蕴含了奥尔定理的条件。

奥尔定理和狄拉克定理都给出了连通简单图有哈密顿回路的充分条件。但是, 这些定理没有给出哈密顿回路存在性的必要条件。例如, 图 C_5 有哈密顿回路, 但既不满足奥尔定理的假设也不满足狄拉克定理的假设, 读者可以验证这一点。

已知的最好的求一个图里的哈密顿回路或判定这样的回路不存在的算法具有指数的最坏情形时间复杂性(相对于图的顶点数来说)。找到具有多项式最坏情形时间复杂性的解法算法是 NP 复杂的, 它的发现将具有重大意义, 因为这意味着其他许多难题都会迎刃而解。

可以用哈密顿通路和哈密顿回路来解决实际问题。例如, 许多应用都要求一条通路或回路, 它要恰好一次地访问一个城市里的每个路口、一个设备网格里的每个管道交汇处或者一个通信网络里的每个节点。求出适当的图模型里的哈密顿通路或哈密顿回路就可以解决这样的问题。著名的旅行商问题要求一个旅行商为了访问一组城市所应当选取的最短路线。这个问题可归结为求完全图的哈密顿回路, 使得这个回路的边的权的总和尽可能小。

现在给出哈密顿回路在编码上的一种不太明显的应用。

例 8 格雷码 旋转的指针的位置可以表示成数字的形式。一种方式是把圆周等分成 2^n 段弧并且用长度为 n 的位串给每段弧赋值。图 6-59 里显示出了用长度为 3 的位串来这样做的两种方式。

用 n 个接触点来确定指针位置的数字表示。每个接触点用来读出位置的数字表示中的一位。图 6-60 针对图 6-59 里的两种赋值对此做了解释。

当指针靠近两段弧的边界时，在读出指针位置时可能发生错误。这可能引起读出的位串里有一个大的错误。例如，在图 6-59a 的编码方案里，若在确定指针位置的过程里发生了一个小的错误，则读出的位串是 100 而不是 011。所有三位都是错的！为了把在确定指针位置的过程里的错误的影响降到最低，用位串对 2^n 段弧的赋值，应当使相邻的弧所表示的位串只相差一位。在图 6-59b 里的编码方案的情况恰好就是这样的。在确定指针位置的过程里的一个错误误给出的位串为 010 而不为 011。只有一位是错的。

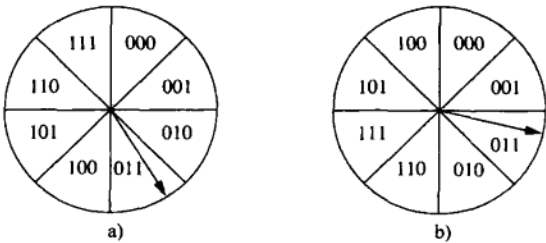


图 6-59 把指针位置转换成数字形式

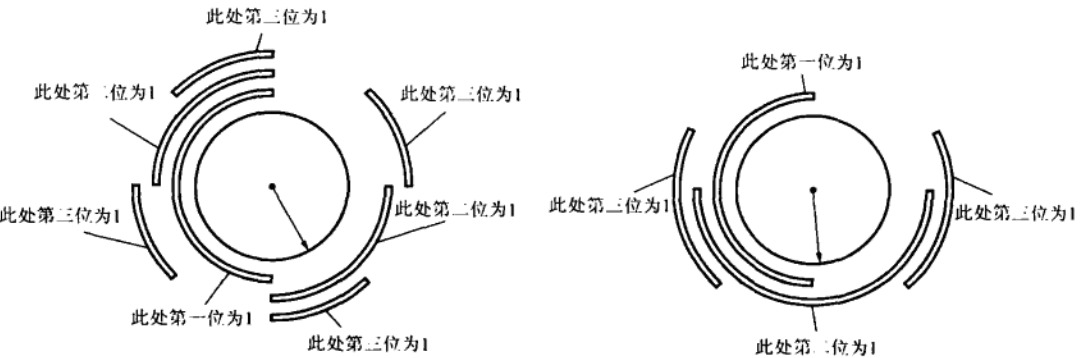


图 6-60 指针位置的数字表示

格雷码是圆周的弧的一种标记，使得相邻的弧具有恰好相差一位的位串标记。在图 6-59b 里的赋值是一个格雷码。可以这样找出格雷码：以下述的方式列出所有长度为 n 的位串，使得每一个串与前一个位串恰好相差一位，而且最后一个串与第一个位串恰好相差一位。可以用 n 立方体 Q_n 来为这个问题建模。解决这个问题所需要的是 Q_n 里的一条哈密顿回路。这样的哈密顿回路容易求出。例如， Q_3 的一条哈密顿回路显示在图 6-61 里。这条哈密顿回路所产生的前后恰好相差一位的位串序列是 000, 001, 011, 010, 110, 111, 101, 100。

格雷码是以弗兰克·格雷的名字来命名的，20 世纪 40 年代，格雷在贝尔实验室研究如何把数字信号传送过程中错误的影响降到最低时发明了它们。

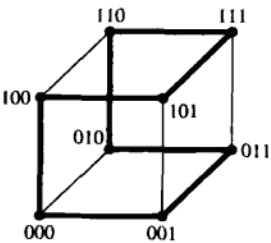
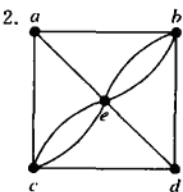
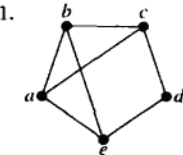
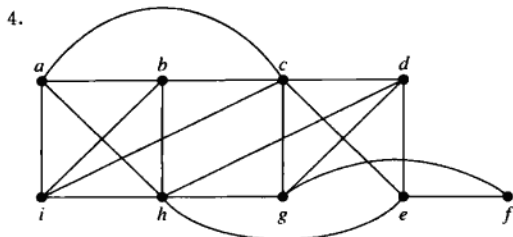
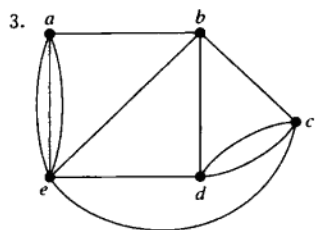


图 6-61 Q_3 的哈密顿回路

练习

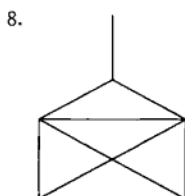
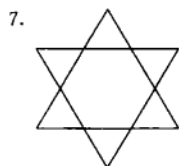
在练习 1~4 里，判定给定的图是否具有欧拉回路。当存在欧拉回路时构造出这样的回路。如果不存在欧拉回路，就确定这个图是否具有欧拉通路，若存在欧拉通路则构造出这样的通路。





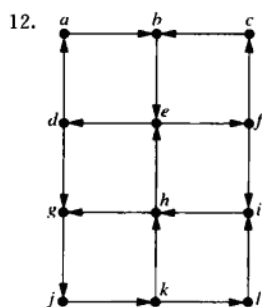
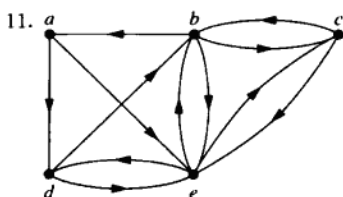
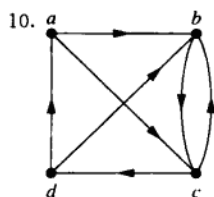
5. 在加里宁格勒(哥尼斯堡现名), 除了18世纪的七座桥之外, 还有另外两座桥。这些新桥分别连接区域B和C以及区域B和D。是否有人能够经过加里宁格勒的九座桥恰好一次并且回到出发点?
6. 何时可以画出一个城市里街道的中心线而不重复经过街道(假设所有街道都是双向街道)?

在练习7~8里, 判定是否可以用一支铅笔连续移动, 不离开纸面并且不重复地画出所示的图形。



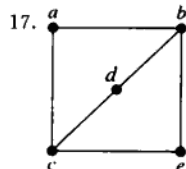
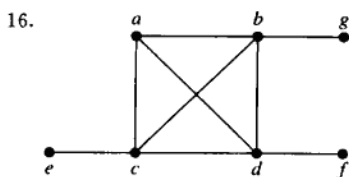
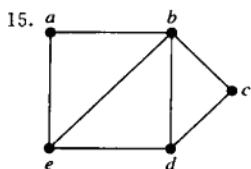
- *9. 证明: 不带有孤立点的有向多重图具有欧拉通路而没有欧拉回路, 当且仅当该图是弱连通的并且存在两个顶点, 一个顶点的入度比出度大1而另外一个顶点的出度比入度大1, 其余每个顶点的入度与出度都相等。

在练习10~12里, 确定所示的有向图是否具有欧拉回路。若存在欧拉回路, 则构造出一条欧拉回路。如果不存在欧拉回路, 就确定这个有向图是否具有欧拉通路。若存在欧拉通路, 则构造出一条欧拉通路。



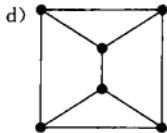
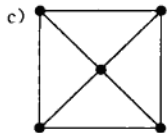
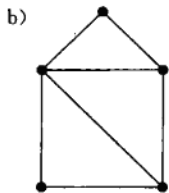
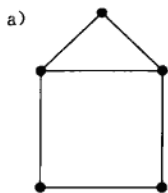
13. 设计一个算法, 构造有向图里的欧拉通路。
14. 当不重复任何部分地画出练习1~4里的每个图时, 求出铅笔必须离开纸面的最少次数。

在练习15~17里, 确定所给的图是否具有哈密顿回路。若有哈密顿回路, 则求出这样一条回路。若没有哈密顿回路, 则论证为什么这样的回路不存在。



18. 对哪些 m 和 n 值来说, 完全偶图 $K_{m,n}$ 具有哈密顿回路?

19. 对于下列每个图确定：(i)能否用狄拉克定理来证明这个图有哈密顿回路；(ii)能否用奥尔定理来证明这个图有哈密顿回路；(iii)这个图是否具有哈密顿回路。



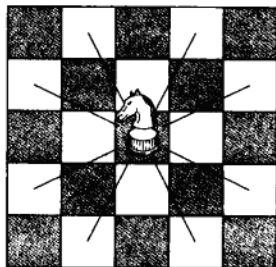
- *20. 证明：每当 n 是正整数时，就存在 n 阶格雷码，或者等价地证明： $n > 1$ 的 n 立方体 Q_n 总是具有哈密顿回路。[提示：用数学归纳法。证明如何从 $n-1$ 阶格雷码产生 n 阶格雷码。]

例 6.10 构造欧拉回路的弗留利算法是从连通多重图的任意一个顶点开始，连续地选择边来形成一条回路。一旦选择了一条边，就删除这条边。连续地选择边，使得每条边从上一条边结束的地方开始，而且使得这条边不是一条割边，除非别无选择。

- *21. 用伪代码表达弗留利算法。
*22. 给出变化的弗留利算法来产生欧拉通路。
23. 证明：带有奇数个顶点的偶图没有哈密顿回路。

马在国际象棋中是这样一种棋子，它的移动可以是水平两格加垂直一格，或者是水平一格加垂直两格。即在 (x, y) 格子的马可以移动到 8 个格子 $(x \pm 2, y \pm 1)$, $(x \pm 1, y \pm 2)$ 中的任何一个，只要这些格子是在棋盘上，如右图所示。

例 6.11 马的周游是马的合法移动的序列，马在某个格子上开始而且访问每个格子恰好一次。若存在一种合法移动，它让马从周游的最后一个格子回到周游开始的地方，则马的周游称为重迭的。可以用图来为马的周游建模，棋盘上每个格子都用一个顶点来表示，若马可以在两个顶点所表示的格子之间合法地移动，则用一条边连接这两个顶点。



24. 画出表示马在 3×4 棋盘上的合法移动的图。
*25. 证明：存在马在 3×4 棋盘上的周游。
*26. 证明：不存在马在 4×4 棋盘上的周游。
27. 本练习粗略介绍奥尔定理的证明。假设 G 是带有 n 个顶点的简单图， $n \geq 3$ ，并且每当 x 和 y 是 G 中不相邻的顶点时， $\deg(x) + \deg(y) \geq n$ 。奥尔定理称在这些条件下 G 有哈密顿回路。
a) 证明：如果 G 没有哈密顿回路，则存在另一个带有与 G 相同顶点的图 H ，可以这样来构造 H ：加入边到 G 使得再加入一条边就产生 H 中的哈密顿回路。[提示：依次在 G 的每个顶点处加入不产生哈密顿回路的尽可能多的边。]
b) 证明：在 H 中存在哈密顿通路。
c) 设 $v_1, v_2, \dots, v_n$ 是 H 中的哈密顿通路。证明： $\deg(v_1) + \deg(v_n) \geq n$ 并且至多存在 $\deg(v_1)$ 个顶点不与 v_n 相邻(包括 v_n 在内)。
d) 设 S 是与哈密顿通路上 v_1 相邻的每个顶点前面的顶点的集合。证明 S 包含 $\deg(v_1)$ 个顶点并且 $v_n \notin S$ 。
e) 证明： S 包含与 v_n 相邻的顶点 v_k 。这蕴含着存在连接 v_1 与 v_{k+1} 和 v_k 与 v_n 的边。
f) 证明：(e) 蕴含着 $v_1, v_2, \dots, v_{k-1}, v_k, v_n, v_{n-1}, \dots, v_{k+1}, v_1$ 是 G 中的哈密顿回路。从这个矛盾得出奥尔定理成立。

6.6 最短通路问题

6.6.1 引言

许多问题可以用边上赋权的图来建模。作为说明，考虑一下航线系统如何建模。如果用顶点

表示城市，用边表示航班。给边赋上城市之间的距离，就可以为涉及距离的问题建模；给边赋上飞行时间，就可以为涉及飞行时间的问题建模；给边赋上票价，就可以为涉及票价的问题建模。图 6-62 显示了给一个图的边赋权的三种不同方式，分别表示距离、飞行时间和票价。

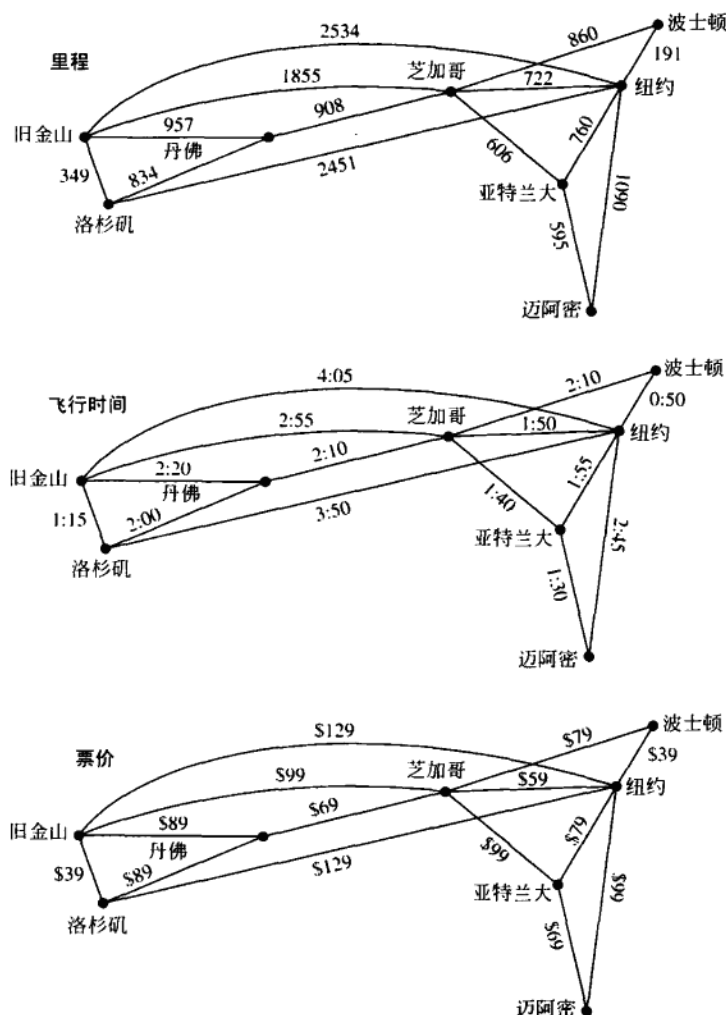


图 6-62 为航线系统建模的带权图

给每条边赋上一个数的图称为带权图。带权图用来为计算机网络建模。通信成本(比如租用电话线的月租费)、计算机在这些线路上的响应时间或计算机之间的距离等都可以用带权图来研究。图 6-63 显示三个带权图，它们表示给计算机网络的图的边赋权的三种方式，分别对应于距离、响应时间和成本。

与带权图有关的几种类型的问题出现得很多。确定网络里两个顶点之间长度最短的通路就是一个这样的问题。说得更具体些，设带权图里一条通路的长度是这条通路上各边的权的总和。(读者应当注意，对术语长度的这种用法，与表示不带权的图的通路里边数的的长度的用法，这两者是不同的。)问题是：什么是最短通路，即什么是在两个给定顶点之间长度最短的通路？例如，在图

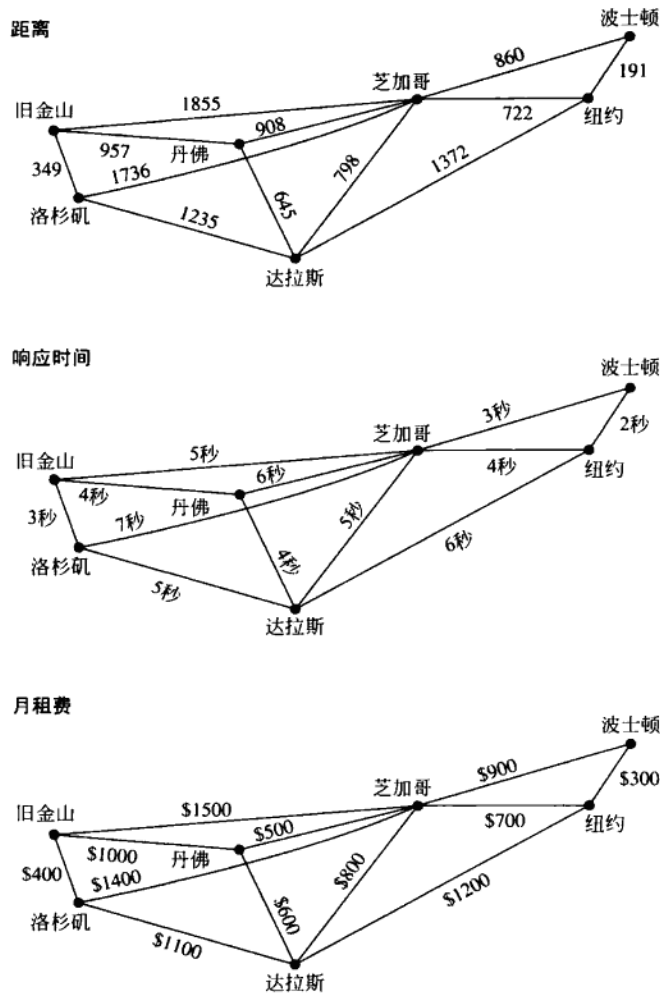


图 6-63 为计算机网络建模的带权图

6-62 所示带权图表示的航线系统里，在波士顿与洛杉矶之间以空中距离计算的最短通路是什么？在波士顿与洛杉矶之间什么样的航班组合的总飞行时间（即在空中的总时间，不包括航班之间的时间）最短？在这两个城市之间的最低费用是多少？在图 6-63 所示的计算机网络里，连接旧金山的计算机与纽约的计算机所需要的最便宜的一组电话线是什么？哪一组电话线给出旧金山与纽约之间的通信的最短响应时间？哪一组电话线有最短的总距离？

与带权图有关的另外一个重要问题是：求访问完全图每个顶点恰好一次的、总长度最短的回路。这就是著名的旅行商问题，它求一位推销员应当以什么样的顺序来访问其路程上的每个城市恰好一次，使得他旅行的总距离最短。本节后面将讨论旅行商问题。

6.6.2 最短通路算法

求带权图里两个顶点之间的最短通路有几个不同的算法。下面将给出荷兰数学家 E·迪克斯特拉在 1959 年所发现的一个解决无向带权图中最短通路问题的算法，其中所有的

权都是正数。可以很容易地将它修改后来解决有向图里的最短通路问题。

在给出这个算法的形式化表示之前将给出一个启发性的例子。

例1 在图6-64所示的带权图里, a 和 z 之间的最短通路的长度是多少?

解 虽然通过观察就容易求出最短通路, 但是需要想出一些有助于理解迪克斯特拉算法的办法。要解决的问题就是: 求从 a 到各个相继的顶点的最短通路, 直到到达 z 为止。

从 a 开始、(直到到达终点为止) 不包含除 a 以外的顶点的唯一通路是 a, b 和 a, d 。因为 a, b 和 a, d 的长度分别为 4 和 2, 所以 d 是与 a 最靠近的顶点。

可以通过查看(直到到达终点为止)只经过 a 和 d 的所有通路, 来求出下一个最靠近 a 的顶点。到 b 的最短通路仍然是 a, b , 长度为 4, 而到 e 的最短通路是 a, d, e , 长度为 5。所以, 下一个与 a 最靠近的顶点是 b 。

为了找出第三个与 a 最靠近的顶点, 只需要检查(直到到达终点为止)只经过了 a, d 和 b 的那些通路。存在长度为 7 到 c 的通路, 即 a, b, c , 以及长度为 6 到 z 的通路, 即 a, d, e, z 。所以, z 是下一个与 a 最靠近的顶点, 而且到 z 的最短通路的长度为 6。

例1说明了在迪克斯特拉算法里使用的一般原理。注意通过检查就可能求出从 a 到 z 的最短通路。不过, 无论对人还是对计算机来说, 检查边数很多的图都是不切实际的。

现在将考虑一般问题: 在无向连通简单带权图里, 求出 a 与 z 之间的最短通路的长度。迪克斯特拉算法如下进行: 求出从 a 到第一个顶点的最短通路的长度, 从 a 到第二个顶点的最短通路的长度, 依此类推, 直到求出从 a 到 z 的最短通路的长度为止。

这个算法依赖于一系列的迭代。通过在每次迭代里添加一个顶点来构造出特殊顶点的集合。在每次迭代里完成一个标记过程。在这个标记过程里, 用只包含特殊顶点的从 a 到 w 的最短通路的长度来标记 w 。添加到特殊顶点集里的顶点是尚在集合之外的那些顶点中带有最小标记的顶点。

现在给出迪克斯特拉算法的细节。它首先用 0 标记 a 而用 ∞ 标记其余的顶点。用记号 $L_0(a) = 0$ 和 $L_0(v) = \infty$ 表示在没有发生任何迭代之前的这些标记(下标 0 表示“第 0 次”迭代)。这些标记是从 a 到这些顶点的最短通路的长度, 其中这些通路只包含顶点 a 。(因为不存在从 a 到其他顶点的这种通路, 所以 ∞ 是 a 与这样的顶点之间的最短通路的长度。)

迪克斯特拉算法是通过形成特殊顶点的集合来进行的。设 S_k 表示在标记过程 k 次迭代之后的特殊顶点集。首先让 $S_0 = \emptyset$ 。集合 S_k 是通过把不属于 S_{k-1} 的带最小标记的顶点 u 添加到 S_{k-1} 里形成的。一旦把 u 添加到 S_k 里, 就更新所有不属于 S_k 的顶点的标记, 使得顶点 v 在第 k 个阶段的标记 $L_k(v)$ 是只包含 S_k 里顶点(即已有的特殊顶点再加上 u)的、从 a 到 v 的最短通路的长度。

设 v 是不属于 S_k 的一个顶点。为了更新 v 的标记, 注意 $L_k(v)$ 是只包含 S_k 里顶点的从 a 到 v 的最短通路的长度。当利用下面的观察结果时, 就可以有效地完成这个更新: 只包含 S_k 里顶点的从 a 到 v 的最短通路, 要么是只包含 S_{k-1} 里顶点(即不包括 u 在内的特殊顶点)的从 a 到 v 的最短通路, 要么是在第 $k-1$ 阶段加上边 (u, v) 的从 a 到 u 的最短通路。换句话说,

$$L_k(a, v) = \min\{L_{k-1}(a, v), L_{k-1}(a, u) + w(u, v)\}$$

这个过程这样迭代: 相继添加顶点到特殊顶点集里, 直到添加 z 为止。当把 z 添加到特殊顶点集里时, 它的标记就是从 a 到 z 的最短通路的长度。在算法 1 里给出迪克斯特拉算法。随后将证明这个算法的正确性。

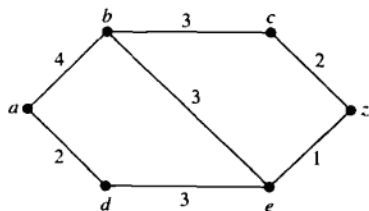


图 6-64 一个带权的简单图

算法1 迪克斯特拉算法**procedure** *Dijkstra*(G : 所有权都为正数的带权连通简单图){ G 带有顶点 $a=v_0, v_1, \dots, v_n=z$ 和权 $w(v_i, v_j)$, 其中若 $\{v_i, v_j\}$ 不是 G 里的边, 则 $w(v_i, v_j)=\infty$ }**for** $i := 1$ **to** n $L(v_i) := \infty$ $L(a) := 0$ $S := \emptyset$ {现在初始化标记, 使得 a 的标记为0而所有其余标记为 ∞ , S 是空集合}**while** $z \notin S$ **begin** $u :=$ 不属于 S 的 $L(u)$ 最小的一个顶点 $S := S \cup \{u\}$ **for** 所有不属于 S 的顶点 v **if** $L(u) + w(u, v) < L(v)$ **then** $L(v) := L(u) + w(u, v)$ {这样就给 S 里添加带最小标记的顶点, 并且更新不在 S 里的顶点的标记}**end**{ $L(z)$ =从 a 到 z 的最短通路的长度}

下面的例子说明迪克斯特拉算法是如何工作的。随后我们将证明这个算法总是产生带权图里两个顶点之间最短通路的长度。

例2 用迪克斯特拉算法求图 6-65a 所示的带权图里顶点 a 与 z 之间最短通路的长度。

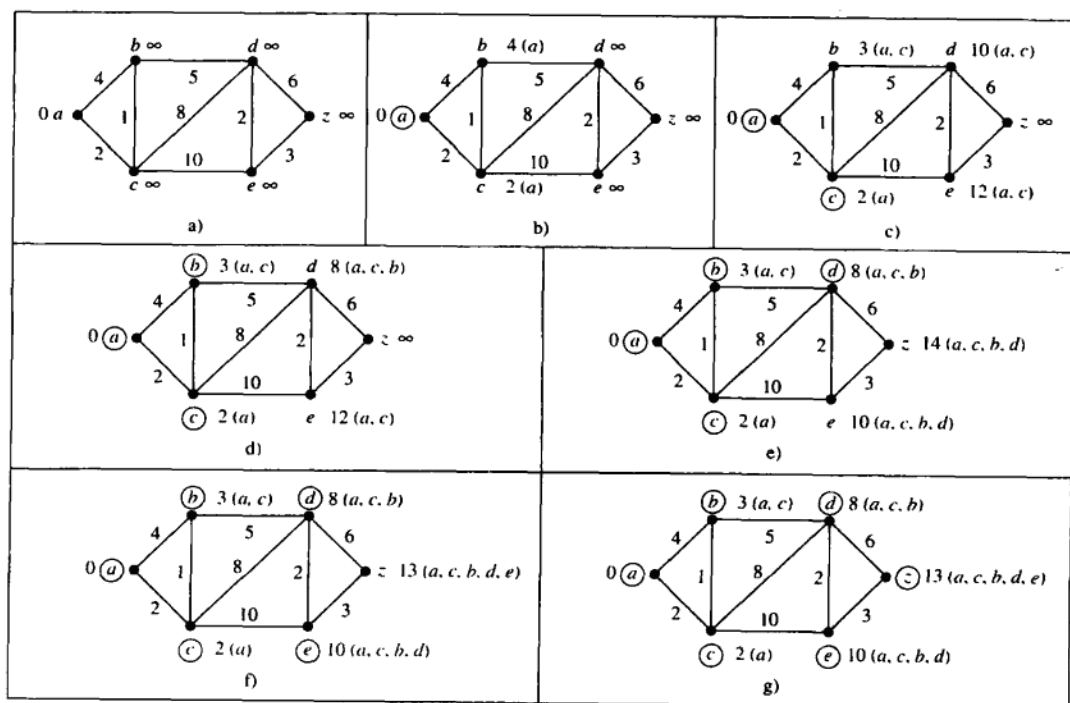


图 6-65 用迪克斯特拉算法求从 a 到 z 的最短通路

解 图 6-65 里显示了迪克斯特拉算法求 a 与 z 之间最短通路所用的步骤。在算法的每次迭代里, 用圆圈圈起集合 S_i 里的顶点。对每次迭代都标明了只包含 S_i 里顶点的从 a 到每个顶点的最短

通路。当圆圈圈到 z 时, 算法终止。找到从 a 到 z 的最短通路是 a, c, b, d, e, z , 长度为 13。■

注意 在执行迪克斯特拉算法时, 为了更便于在每步里跟踪顶点的标记, 有时可以用一个表来代替, 而不再对每步都重新画出这个图。

下一步, 用归纳论证来证明迪克斯特拉算法产生无向连通带权图里两个顶点 a 与 z 之间最短通路的长度。用下列断言作为归纳假设: 在第 k 次迭代里

(i) S 里的顶点 $v(v \neq 0)$ 的标记是从 a 到这个顶点的最短通路的长度。

(ii) 不在 S 里的顶点的标记是(这个顶点自身除外)只包含 S 里顶点的、从 a 到这个顶点的最短通路的长度。

当 $k=0$ 时, 在没有执行任何迭代之前, $S=\{a\}$, 所以从 a 到除 a 外的顶点的最短通路的长度是 ∞ 。设 v 是在第 $k+1$ 次迭代里添加到 S 里的顶点, 使得 v 是在第 k 次迭代结束时带最小标记的不在 S 里的顶点(在该顶点不唯一的情形里, 可以采用带最小标记的任意顶点)。

根据归纳假设, 可以看出在第 $k+1$ 次迭代之前, S 里的顶点都用从 a 出发的最短通路的长度来标记。另外, 必须用从 a 到 v 的最短通路的长度来标记 v 。假如情况不是这样, 那么在第 k 次迭代结束时, 就可能存在包含不在 S 里的顶点的、长度小于 $L_k(v)$ 的通路(因为 $L_k(v)$ 是在第 k 次迭代之后、只包含 S 里顶点的、从 a 到 v 的最短通路的长度)。设 u 是在这样的通路里不属于 S 的第一个顶点。则存在一条只包含 S 里顶点的、从 a 到 u 的长度小于 $L_k(v)$ 的通路。这与对 v 的选择相矛盾。因此, 在第 $k+1$ 次迭代结束时(i)成立。

设 u 是在第 $k+1$ 次迭代之后不属于 S 的一个顶点。只包含 S 里顶点的从 a 到 u 的最短通路要么包含 v 、要么不包含 v 。若它不包含 v , 则根据归纳假设, 它的长度是 $L_k(u)$ 。若它确实包含 v , 则它必然是这样组成的: 一条只包含 S 里除 v 之外的顶点的、从 a 到 v 的、具有最短可能长度的通路, 后面接着从 v 到 u 的边。在这种情形里它的长度是 $L_k(v) + w(v, u)$ 。这样就证明了(ii)为真, 因为 $L_{k+1}(u) = \min\{L_k(u), L_k(v) + w(v, u)\}$ 。

已经证明了定理 1。

定理 1 迪克斯特拉算法求出连通简单无向带权图里两个顶点之间最短通路的长度。

现在可以估计迪克斯特拉算法的计算复杂性(就加法和比较而言)。这个算法使用的迭代次数不超过 $n-1$ 次, 因为在每次迭代里添加一个顶点到特殊顶点集里。若可以估计每次迭代所使用的运算次数, 则大功告成了。可以用不超过 $n-1$ 次比较来找出不在 S_k 里的带最小标记的顶点。于是我们使用一次加法和一次比较来更新不在 S_k 中的每一个顶点的标记, 所以在每次迭代里的运算不超过 $2(n-1)$ 次, 因为在每次迭代里要更新的标记不超过 $n-1$ 个。因为迭代次数不超过 $n-1$ 次, 每次迭代的运算次数不超过 $2(n-1)$ 次, 所以有下面的定理。

定理 2 迪克斯特拉算法使用 $O(n^2)$ 次运算(加法和比较)来求出连通简单无向带权图里两个顶点之间最短通路的长度。

6.6.3 旅行商问题

 现在讨论与带权图有关的一个重要问题。考虑下面的问题: 一位旅行商想要访问 n 个城市中每个城市恰好一次, 并且返回到出发点。例如, 假定这个旅行商想要访问底特律、托莱多、萨吉诺、格兰特·拉皮滋以及卡拉玛祖(见图 6-66)。他应当以什么顺序访问这些城市以便旅行总距离最短? 为了解决这个问题, 可以假定旅行商从底特律出发(因为这个城市必须是回路的一部分), 并且检查他访问其余四个城市然后返回底特律的所有可能方式(从别处出发将产生相同的回路)。这样的回路总共有 24 条, 但是因为往返路程距离相同, 所以在求最短总距离时, 只需要考虑 12 条不同的回路即可。列出这 12 条不同回路和每条回路旅行的最短总距离。从表 6-6 可以看出, 回路底特律-托莱多-卡拉玛祖-格兰特·拉皮滋-萨吉诺, 对应 458 英里的最短总距离。

上面描述了旅行商问题的一个实例。旅行商问题求带权完全无向图里访问每个顶点恰好一次并且返回出发点的总权数最小的回路。这等价于求完全图里总权数最小的哈密顿回路, 因为在问

路里访问每个顶点恰好一次。

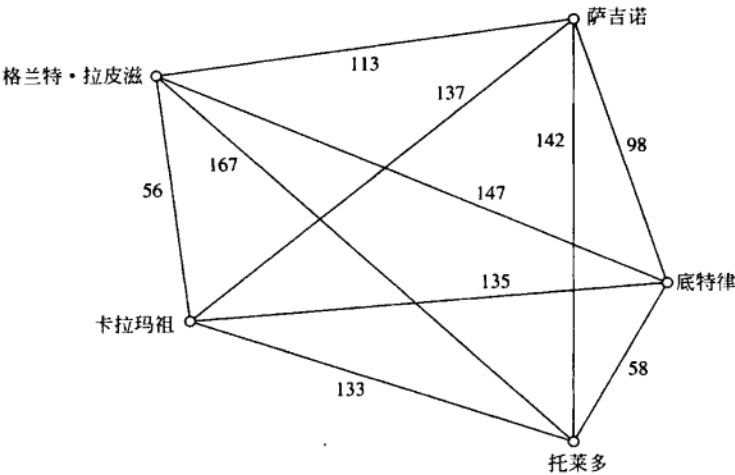


图 6-66 说明五个城市之间距离的图

表 6-6

路 线	总距离(英里)
底特律-托莱多-格兰特·拉皮滋-萨吉诺-卡拉玛祖-底特律	610
底特律-托莱多-格兰特·拉皮滋-卡拉玛祖-萨吉诺-底特律	516
底特律-托莱多-卡拉玛祖-萨吉诺-格兰特·拉皮滋-底特律	588
底特律-托莱多-卡拉玛祖-格兰特·拉皮滋-萨吉诺-底特律	458
底特律-托莱多-萨吉诺-卡拉玛祖-格兰特·拉皮滋-底特律	540
底特律-托莱多-萨吉诺-格兰特·拉皮滋-卡拉玛祖-底特律	504
底特律-萨吉诺-托莱多-格兰特·拉皮滋-卡拉玛祖-底特律	598
底特律-萨吉诺-托莱多-卡拉玛祖-格兰特·拉皮滋-底特律	576
底特律-萨吉诺-卡拉玛祖-托莱多-格兰特·拉皮滋-底特律	682
底特律-萨吉诺-格兰特·拉皮滋-托莱多-卡拉玛祖-底特律	646
底特律-格兰特·拉皮滋-萨吉诺-托莱多-卡拉玛祖-底特律	670
底特律-格兰特·拉皮滋-托莱多-萨吉诺-卡拉玛祖-底特律	728

最直截了当的求解旅行商问题实例的方式是检查所有可能的哈密顿回路，并且挑出总权数最小的一条回路。若在图中有 n 个城市，则为了求解这个问题，得检查多少条回路？一旦选定了出发点，需要检查的不同的哈密顿回路就有 $(n-1)!$ 条，因为第二个顶点有 $n-1$ 种选择，第三个顶点有 $n-2$ 种选择，依此类推。因为可以用相反顺序来经过一条哈密顿回路，所以只需要检查 $(n-1)!/2$ 条回路来求出答案。注意 $(n-1)!/2$ 增长得极快。当只有几十个顶点时，试图用这种方式来解决旅行商问题就是不切实际的。例如，假如有 25 个顶点，那么就不得不考虑总共 $24!/2$ (约为 3.1×10^{23}) 条不同的哈密顿回路。假定检查每条哈密顿回路只花费 1 纳秒 (10^{-9} 秒)，那么就需要大约一千万年才能求出这个图里长度最短的一条哈密顿回路。

因为旅行商问题在实践和理论上都具有重要意义，所以已经投入了巨大的努力来设计解决它的有效算法。不过，还没有已知的解决这个问题的多项式最坏情形时间复杂性的算法。进一步地，假如这种算法找到了，那么许多其他难题(比如在第 1 章里讨论过的确定 n 个变元的命题公式是否重言式)也可以迎刃而解。这个结果是从 NP 完全性理论得出的(关于这个理论的更多信息请参考 [GaJo79])。

当有许多需要访问的顶点时,解决旅行商问题的实际方法是使用近似算法。近似算法是这样的算法,它们不必产生问题的精确解,取而代之的是保证产生接近精确解的解。即它们可能产生带总权数 W' 的哈密顿回路,使得 $W \leq W' \leq cW$, 其中 W 是精确解的总长度,而 c 是一个常数。例如,存在多项式最坏情形时间复杂性算法使得 $c=3/2$ 。对于一般带权图和每个正实数 k 来说,总是产生至多 k 倍于最优解的解的算法还是未知。假如这样的算法存在,那就可能证明 P 类等于 NP 类,这是关于算法复杂性的最著名的未解决问题。

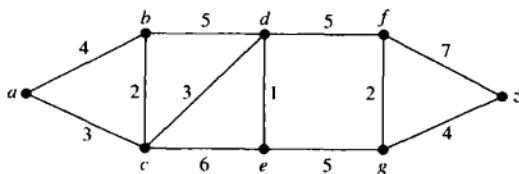
在实际中,已经研究出这样的算法,它们可以只用几分钟的机时,就解决多达 1000 个顶点的旅行商问题,误差在精确解的 2% 之内。关于旅行商问题的更多信息,包括历史、应用和算法等,见《离散数学的应用》(*Applications of Discrete Mathematics*) [MiRo91] 中关于这个主题的那一章。

练习

1. 对下列关于地铁系统的每个问题,描述一个可以用来解决这个问题的带权图模型。

- 在两站之间旅行所需要的最短时间是什么?
- 从一站到达另外一站所经过的最短距离是什么?
- 若把各站之间的票价求和就得出总票价,则在两站之间的最低票价是什么?

2. 求下面的带权图里 a 与 z 之间的最短通路的长度。



3. 在练习 2 的带权图中, a 与 z 之间的最短通路是什么?

4. 在图 6-63 所示的通信网络里,求下列每对城市的计算机中心之间的(距离)最短路线。

- 波士顿与洛杉矶
- 纽约与旧金山
- 达拉斯与旧金山
- 丹佛与纽约

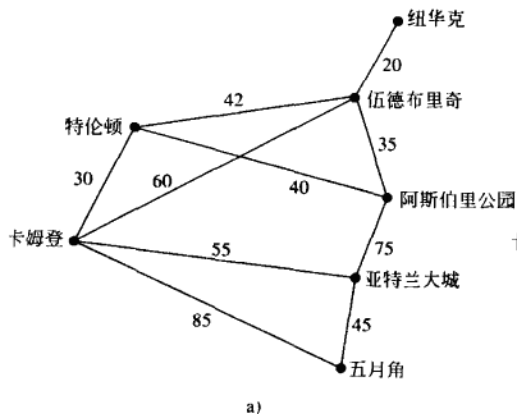
5. 利用图 6-63 里给出的租费,求在练习 4 中成对的计算机中心之间月租费最便宜的路线。

6. 推广求带权简单有向连通图里两个顶点之间最短通路的迪克斯特拉算法,以便求出在顶点 a 与图中其余每个顶点之间的最短通路的长度。

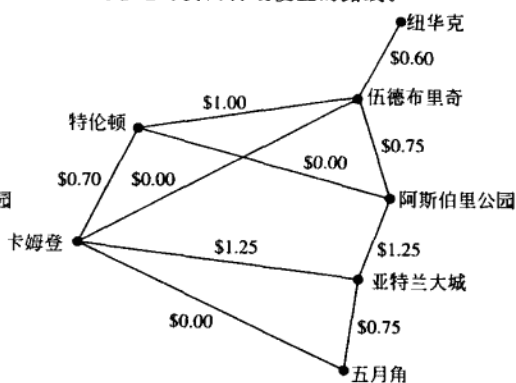
7. 在下图里的带权图说明新泽西的一些主要道路。图 a 说明这些道路上的城市之间的距离;图 b 说明通行费。

a) 利用这些道路,求在纽华克与卡姆登之间,以及在纽华克与五月角之间的距离最短的路线。

b) 利用给出的道路图,求在本题部分(a)里成对城市之间就总通行费而言最便宜的路线。



a)



b)

8. 哪些应用必须求出带权图里两个顶点之间的最长简单通路的长度?

可以用弗洛伊德算法来求出带权连通简单图里所有顶点对之间最短通路的长度。不过,不能用这个算法来构造出最短通路(在下面,把无穷权赋给任何一对不被图中的边所连接的顶点上)。

算法 2 弗洛伊德算法

procedure *Floyd*(*G*: 带权简单图)

{*G* 有顶点 $v_1, v_2, \dots, v_n$ 和权 $w(v_i, v_j)$, 其中若 (v_i, v_j) 不是边, 则 $w(v_i, v_j) = \infty$ }

for $i := 1$ **to** n

for $j := 1$ **to** n

$d(v_i, v_j) := w(v_i, v_j)$

for $i := 1$ **to** n

for $j := 1$ **to** n

for $k := 1$ **to** n

if $d(v_j, v_i) + d(v_i, v_k) < d(v_j, v_k)$

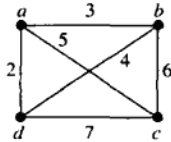
then $d(v_j, v_k) := d(v_j, v_i) + d(v_i, v_k)$

{ $d(v_i, v_j)$ 是在 v_i 与 v_j 之间的最短通路的长度}

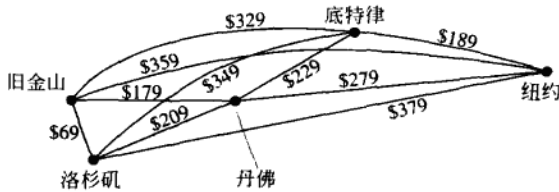
9. 用弗洛伊德算法求图 6-65a 中带权图里所有顶点对之间的距离。

*10. 给出弗洛伊德算法为了确定在带有 n 个顶点的带权简单图里、所有顶点对之间的最短距离而使用的运算(比较和加法)次数的大 O 估计。

11. 通过求出所有哈密顿回路的总权数并且确定出总权数最小的回路, 来解决下图的旅行商问题。



12. 求访问下图里每个城市的机票总价最低的路线, 其中边上的权是在这两个城市之间的航班所提供的最低票价。



13. 构造一个带权无向图, 使得对于访问某些顶点超过一次的回路来说, 访问每个顶点至少一次的回路的总权数是最小的。[提示: 存在带三个顶点的例子。]

6.7 可平面图

6.7.1 引言

考虑把三座房屋与三种设施每种都连接起来的问题, 如图 6-67 所示。是否有可能这样来连接这些房屋与设施, 使得在这样的连接里不发生交叉? 这个问题可以用完全偶图 $K_{3,3}$ 来建模。原来的问题可以重新叙述为: 能否在平面里画出 $K_{3,3}$, 使得没有两条边发生交叉?

本节将研究能否在平面里让边不交叉地画出一个图的问题。特别是, 将回答这个房屋与设施的问题。

图的表示方式有许多种。何时有可能至少求出一种方式来在平面里表示这个图而让边没有任何交叉?

定义 1 若可以在平面里画出一个图而边没有任何交叉(其中边的交叉是表示边的直线或弧线在它们的公共端点以外的地方相交), 则这个图是可平面的。这种画法称为这个图的平面表示。

即使通常带交叉地画出了一个图, 这个图也仍然可能是可平面的, 这是因为有可能以不同的方式来不带交叉地画出这个图。

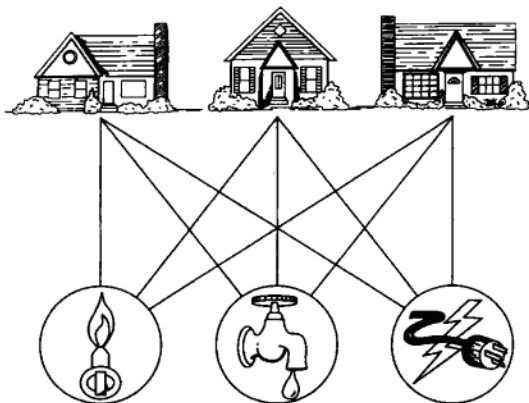
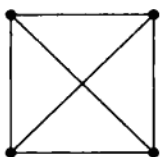
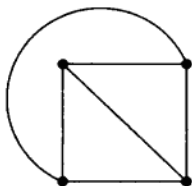


图 6-67 三座房屋与三种设施

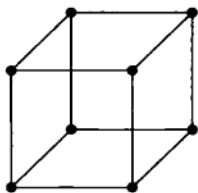
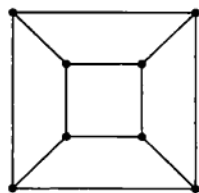
例 1 K_4 (如图 6-68 所示, 有两条边交叉) 是可平面吗?

解 K_4 是可平面的, 因为可以不带交叉地画出它, 如图 6-69 所示。

图 6-68 图 K_4 图 6-69 不带交叉的图 K_4

例 2 图 6-70 所示的 Q_3 是可平面吗?

解 Q_3 是可平面的, 因为可以画出它而没有任何边交叉, 如图 6-71 所示。

图 6-70 图 Q_3 图 6-71 Q_3 的一种平面表示

可以通过显示一种平面表示来证明一个图是可平面的。更难的是证明一个图是非可平面的。下面给出一个例子说明如何以一种特别的方法来做到这一点。

例 3 图 6-72 所示的 $K_{3,3}$ 是可平面吗?

解 在平面里画出 $K_{3,3}$ 而没有边交叉, 任何这样的尝试都注定是失败的。现在说明这是为什么。在 $K_{3,3}$ 的任何平面表示里, 顶点 v_1 和 v_2 都必须同时与 v_4 和 v_5 连接。这四条边所形成的封闭曲线把平面分割成两个区域 R_1 和 R_2 , 如图 6-73a 所示。顶点 v_3 属于 R_1 或 R_2 。当 v_3 属于闭曲线的内部 R_2 时, 在 v_3 和 v_4 之间以及在 v_3 和 v_5 之间的边, 把 R_2 分割成两个区域 R_{21} 和 R_{22} , 如图 6-73b 所示。

下一步, 已经没有办法来不交叉地放置最后一个顶点 v_6 。因为若 v_6 属于 R_1 , 则不能不带交叉地画出 v_6 和 v_3 之间的边。若 v_6 属于 R_{21} , 则不能不带交叉地画出 v_2 和 v_6 之间的边。若 v_6 属于

R_{22} , 则不能不带交叉地画出 v_1 和 v_6 之间的边。

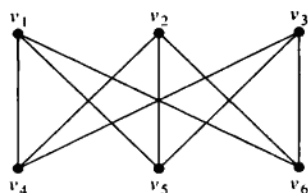


图 6-72 图 $K_{3,3}$

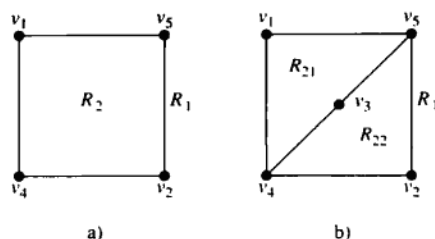


图 6-73 证明 $K_{3,3}$ 是非可平面的

当 v_3 属于 R_1 时, 可以使用类似的论证。请读者来完成这个论证。所以 $K_{3,3}$ 是非可平面的。■

例 3 解决了在本节开头所描述的设施与房屋的问题。不能在平面里连接这三座房屋与三种设施而不发生交叉。可以用类似的论证来证明 K_5 是非可平面的(见本节末的练习 6)。

图的平面性在电子电路的设计中有重要作用。可以用图来为电路建立模型, 用顶点表示电路的器件并且用边表示器件之间的连接。如果表示一个电路的图是可平面的, 则可以把这个电路无连接交叉地印刷在单个电路板上。当这个图不是可平面的时, 就必须求助于成本更高的选项。例如, 可以把表示电路的图的顶点划分成可平面子图。然后使用多层来构造这个电路。每当连接交叉时就可以用绝缘线来构造电路。在这种情况下, 重要的是以尽可能少的交叉来画出这个图。

6.7.2 欧拉公式

一个图的可平面表示把平面分割成一些面, 包括一个无界的面。例如, 图 6-74 所示的图的可平面表示把平面分割成 6 个面并加以标记。欧拉证明过一个图的所有可平面表示都把平面分割成相同数目的面。他是这样证明的: 求出可平面图的面数、顶点数以及边数之间的关系。

定理 1 欧拉公式 设 G 是带 e 条边和 v 个顶点的连通可平面简单图。设 r 是 G 的可平面表示里的面数。则 $r = e - v + 2$ 。

证 首先规定 G 的可平面表示。将要这样证明定理: 构造一系列子图 $G_1, G_2, \dots, G_r = G$, 相继地在每个阶段添加一条边。用下面的归纳定义来这样做。任意地选择一条 G 的边来获得 G_1 。这样通过 G_{n-1} 获得 G_n : 任意地添加一条与 G_{n-1} 里顶点相关联的边, 若与这条边关联的另一个顶点还不在于 G_{n-1} 里, 则添加这个顶点。这样的构造是可能的, 因为 G 是连通的。在添加 e 条边之后就获得 G 。设 r_n, e_n 和 v_n 分别表示 G 的可平面表示所引导出的 G_n 的可平面表示的面数、边数和顶点数。

现在通过归纳来进行证明。对 G_1 来说, 关系 $r_1 = e_1 - v_1 + 2$ 为真, 因为 $e_1 = 1, v_1 = 2$, 而 $r_1 = 1$ 。这种情形如图 6-75 所示。

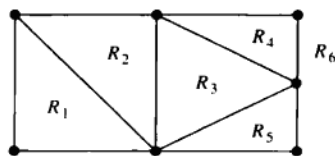


图 6-74 图的可平面表示的面

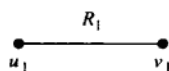


图 6-75 欧拉公式的证明中的基础情形

现在假定 $r_n = e_n - v_n + 2$ 。设 $\{a_{n+1}, b_{n+1}\}$ 是为了获得 G_{n+1} 而添加到 G_n 上的边。有两种情形需要考虑。在第一种情形里, a_{n+1} 和 b_{n+1} 都已经在 G_n 里了。这两个顶点必然是在一个公共面 R 的边界上, 否则就不可能把边 $\{a_{n+1}, b_{n+1}\}$ 添加到 G_n 里而没有两条边交叉(并且 G_{n+1} 是可平面的)。这条新边的添加把 R 分割成两个面。所以, 在这种情形里, $r_{n+1} = r_n + 1, e_{n+1} = e_n + 1$, 而且 $v_{n+1} = v_n$ 。因此, 关系到面数、边数、顶点数的公式两边都恰好增加 1, 所以这个公式仍然为真。换句话

说, $r_{n+1} = e_{n+1} - v_{n+1} + 2$. 在图 6-76a 里说明这种情形。

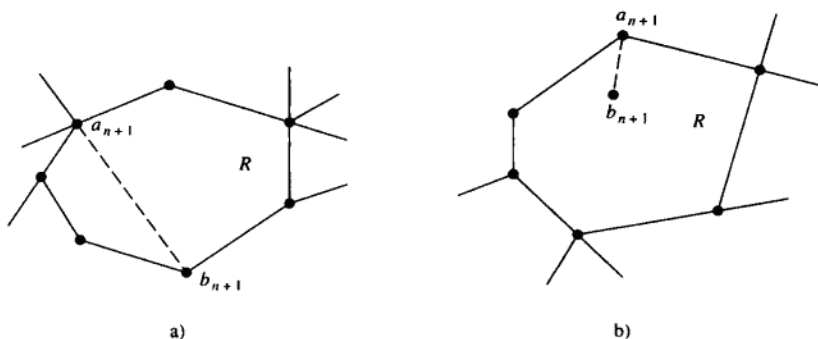


图 6-76 添加一条边到 G_n 来产生 G_{n+1}

在第二种情形里, 新边的两个顶点之一还不在于 G_n 里。假定 a_{n+1} 在 G_n 里但是 b_{n+1} 不在 G_n 里。添加这条新边不产生任何新的面, 因为 b_{n+1} 必然是在边界上有 a_{n+1} 的一个面里。所以, $r_{n+1} = r_n$ 。另外, $e_{n+1} = e_n + 1$ 且 $v_{n+1} = v_n + 1$ 。关系到面数、边数、顶点数的公式两边都保持相等, 所以这个公式仍然为真。换句话说, $r_{n+1} = e_{n+1} - v_{n+1} + 2$ 。在图 6-76b 里说明这种情形。

已经完成了归纳论证。因此对所有 n 来说, 都有 $r_n = e_n - v_n + 2$ 。因为原图是在添加了 e 条边之后所获得的图 G , 所以这个定理为真。□

例 4 解释欧拉公式。

例 4 假定连通可平面简单图有 20 个顶点, 每个顶点度都为 3。这个可平面图表示把平面分割成多少个面?

解 这个图有 20 个顶点, 每个顶点的度都为 3, 所以 $v=20$ 。因为这些顶点的度之和 $3v = 3 \cdot 20 = 60$ 等于边数的两倍 $2e$, 所以有 $2e=60$ 或 $e=30$ 。所以, 根据欧拉公式, 面数是

$$r = e - v + 2 = 30 - 20 + 2 = 12$$

可以用欧拉公式来建立可平面图所必须满足的一些不等式。在下面的推论里给出一个这样的不等式。

推论 1 若 G 是带 e 条边和 v 个顶点的连通可平面简单图, 其中 $v \geq 3$, 则 $e \leq 3v - 6$ 。

在证明推论 1 之前先用它证明下面这个有用的结论。

推论 2 若 G 是连通可平面简单图, 则 G 有度数不超过 5 的顶点。

证 如果 G 有 1 个或 2 个顶点, 则结果为真。如果 G 至少有 3 个顶点, 则根据推论 1 知道 $e \leq 3v - 6$, 所以 $2e \leq 6v - 12$ 。假如每个顶点的度数至少是 6, 则由于 $2e = \sum_{v \in V} \deg(v)$ (根据握手定理), 所以就有 $2e \geq 6v$ 。但是这与 $2e \leq 6v - 12$ 相矛盾。所以必定存在度数不超过 5 的顶点。□

推论 1 的证明是基于面的次数的概念, 它定义为这个面的边界上的边数。当一条边在边界上出现两次(所以当描画边界时就描画它两次)时, 它贡献的次数是 2。图 6-77 里显示出面的次数。

现在可以给出推论 1 的证明了。

证 画在平面里的连通可平面简单图把平面分割成面, 比如说 r 个面。每个面的次数至少为 3。(因为这里所讨论的图都是简单图, 不允许带有可能产生 2 次面的多重边, 或者可能产生 1 次面的环。)特别地, 注意无界的面次数

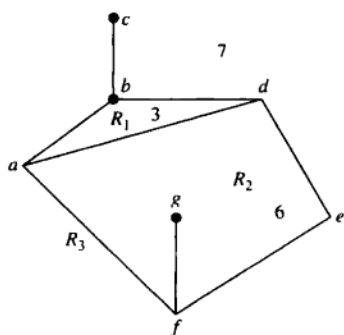


图 6-77 面的次数

至少为3,因为在图里至少有三个顶点。

注意各面的次数之和恰好是图中边数的两倍,因为每条边都在面的边界上出现两次(或者在两个不同面里,或者两次在相同面里)。因为每个面都有大于或等于3的次数,所以

$$2e = \sum_{\text{所有区域} R} \deg(R) \geq 3r$$

因此,

$$(2/3)e \geq r$$

利用 $r = e - v + 2$ (欧拉公式), 就得到

$$e - v + 2 \leq (2/3)e$$

所以 $e/3 \leq v - 2$ 。这样就证明了 $e \leq 3v - 6$ 。 □

可以用这个推论来证明 K_5 是非可平面的。

例5 用推论1证明: K_5 是非可平面的。

解 图 K_5 有5个顶点和10条边。不过,对这个图来说,不满足不等式 $e \leq 3v - 6$, 因为 $e = 10$ 和 $3v - 6 = 9$ 。因此, K_5 不是可平面的。 ■

前面已经证明了 $K_{3,3}$ 不是可平面的。不过,注意这个图有6个顶点和9条边。这意味着满足不等式 $e = 9 \leq 12 = 3 \cdot 6 - 6$ 。所以,满足不等式 $e \leq 3v - 6$ 的事实并不意味着一个图是可平面的。不过,可以利用下面定理1的推论来证明 $K_{3,3}$ 不是可平面的。

推论3 若连通可平面简单图有 e 条边和 v 个顶点, $v \geq 3$ 并且没有长度为3的回路, 则 $e \leq 2v - 4$ 。

推论3的证明类似于推论1的证明,不同之处在于,在这种情形里,没有长度为3的回路的事实蕴含着面的次数必然至少为4。把这个证明的细节留给读者(见本节末的练习8)。

例6 用推论3证明: $K_{3,3}$ 是非可平面的。

解 因为 $K_{3,3}$ 没有长度为3的回路(容易看出这一点,因为它是偶图),所以可以使用推论3。 $K_{3,3}$ 有6个顶点和9条边。因为 $e = 9$ 和 $2v - 4 = 8$, 所以推论3证明 $K_{3,3}$ 是非可平面的。 ■

6.7.3 库拉图斯基定理

我们已经看到 $K_{3,3}$ 和 K_5 都不是可平面的。显然,若一个图包含这两个图作为子图,则它不是可平面的。另外,所有非可平面图必然包含一个子图,它是可以利用某些允许的操作从 $K_{3,3}$ 或 K_5 来获得的。

若一个图是可平面的,则通过删除一条边 $\{u, v\}$ 并且添加一个新顶点 w 和两条边 $\{u, w\}$ 与 $\{w, v\}$, 所获得的任何图也是可平面的。这样的操作称为初等细分。若可以从相同的图通过一系列初等细分来获得图 $G_1 = (V_1, E_1)$ 和图 $G_2 = (V_2, E_2)$, 则它们称为是同胚的。

例7 证明: 图6-78所示的图 G_1, G_2, G_3 是同胚的。

解 因为这三个图都可以从图 G_1 通过初等细分得到,所以它们是同胚的。 G_1 可以从它自身出发,通过一个空的初等细分序列而得到。要从 G_1 得到 G_2 , 采用如下初等细分序列:

- 1) 删掉边 $\{a, c\}$, 增加顶点 f , 然后添加边 $\{a, f\}$ 和 $\{f, c\}$ 。
- 2) 删掉边 $\{b, c\}$, 增加顶点 g , 然后添加边 $\{b, g\}$ 和 $\{g, c\}$ 。
- 3) 删掉边 $\{b, g\}$, 增加顶点 h , 然后添加边 $\{g, h\}$ 和 $\{h, b\}$ 。

把找出由 G_1 到 G_2 的初等细分序列的任务留给读者。 ■

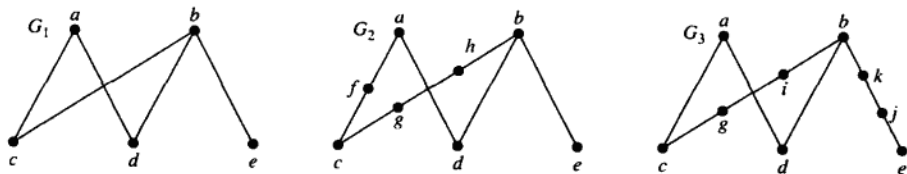


图 6-78 同胚图

波兰数学家库拉图斯基在1930年证明了下面的定理，这个定理用图同胚的概念刻画了可平面图。

定理2 一个图是非可平面的当且仅当它包含一个同胚于 $K_{3,3}$ 或 K_5 的子图。

显然，一个包含着同胚于 $K_{3,3}$ 或 K_5 的子图的图是非可平面的。不过，相反方向的命题——即每个非可平面图都包含一个同胚于 $K_{3,3}$ 或 K_5 的子图——证明起来是复杂的，因而不在这里给出。下面的例子说明如何使用库拉图斯基定理。

例8 确定图6-79所示的图 G 是否是可平面的。

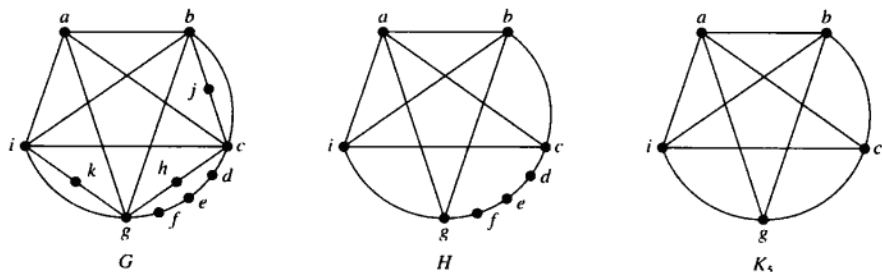


图 6-79 无向图 G 、同胚于 K_5 的子图 H 以及 K_5

解 G 有同胚于 K_5 的子图 H 。 H 是这样获得的：删除 h, j 和 k 以及所有与这些顶点关联的边。 H 是同胚于 K_5 的，因为从 K_5 (带有顶点 a, b, c, g 和 i) 通过一系列初等细分，添加顶点 d, e 和 f 就可以获得 H (读者应当构造出这样的一系列初等细分)。因此， G 是非可平面的。

例9 在图6-80a中所示的彼得森图是可平面吗？(丹麦数学家朱利乌斯·彼得森在1891年介绍过这个图；它常用来说明关于图的各种性质的理论。)

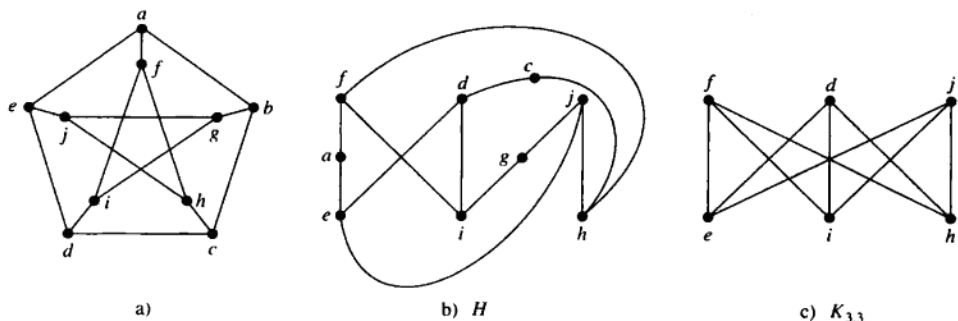


图 6-80 a) 彼得森图；b) 同胚于 $K_{3,3}$ 的子图 H ；c) $K_{3,3}$

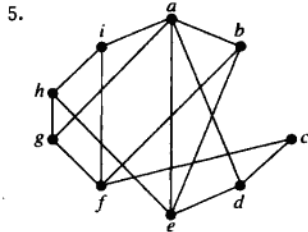
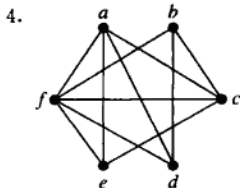
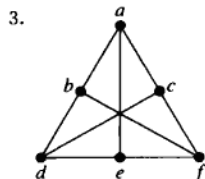
解 彼得森图的子图 H 是这样获得的：删除 b 和以 b 作为端点的三条边，如图6-78b所示，它是同胚于带有顶点集合 $\{f, d, j\}$ 和 $\{e, i, h\}$ 的 $K_{3,3}$ 的，这是因为可以通过一系列初等细分来获得它，删除 $\{d, h\}$ 并且添加 $\{c, h\}$ 和 $\{c, d\}$ ，删除 $\{e, f\}$ 并且添加 $\{a, e\}$ 和 $\{a, f\}$ ，删除 $\{i, j\}$ 并且添加 $\{g, i\}$ 和 $\{g, j\}$ 。因此，彼得森图不是可平面的。

练习

1. 五座房屋能否不带连接交叉地与两种设施相连接？
2. 不带任何交叉地画出下面的可平面图。



在练习 3~5 里, 确定所给的图是否可平面。若是可平面的, 则画出它使得没有边交叉。



6. 用类似于在例 3 里给出的论证来证明: K_5 是非可平面的。

7. 假定一个连通可平面图有 6 个顶点, 每个顶点的度都为 4。这个图的可平面表示把平面分割成多少个面?

8. 证明推论 3。

*9. 假定一个带有 e 条边和 v 个顶点的连通可平面简单图不包含长度为 4 或更短的回路。证明: 若 $v \geq 4$ 则 $e \leq (5/3)v - (10/3)$ 。

10. 下面的哪些非可平面图具有这样的性质: 删除任何一个顶点以及与这个顶点关联的所有边就产生一个可平面图?

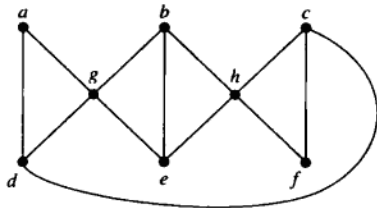
a) K_5

b) K_6

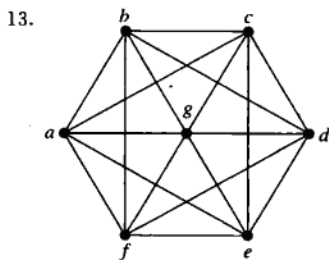
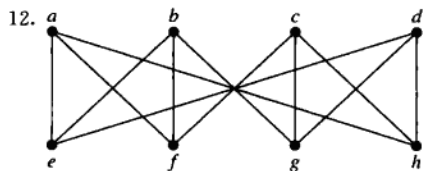
c) $K_{3,3}$

d) $K_{3,4}$

11. 确定下面的图是否同胚于 $K_{3,3}$ 。



在练习 12~13 里, 用库拉图斯基定理来确定所给的图是不是可平面的。



一个简单图的交叉数, 是指当在平面上画出这个图(其中不允许任何三条表示边的弧线在同一个点交叉)时交叉的最少次数。

**14. 求下面每个非可平面图交叉数。

a) K_5

b) K_6

c) K_7

d) $K_{3,4}$

e) $K_{4,4}$

f) $K_{5,5}$

**15. 证明: 若 m 和 n 都是偶正整数, 则 $K_{m,n}$ 的交叉数小于或等于 $mn(m-2)(n-2)/16$ 。[提示: 沿着 x 轴放置 m 个顶点, 使得它们间距相等并且关于原点对称, 再沿着 y 轴放置 n 个顶点, 使得它们间距相等并且关于原点对称。现在连接 x 轴上 m 个顶点中的每一个与 y 轴上 n 个顶点中的每一个, 并且计算交叉数。]

简单图 G 的厚度是指, 以 G 作为它们的并图的 G 的可平面子图的最小个数。

*16. 求练习 14 里的图的厚度。

*17. 在一个环面上画出 $K_{3,3}$, 使得没有边交叉。

6.8 图着色

6.8.1 引言

在图论中,有许多与地图区域(比如世界各部分的地图)着色有关的理论成果。当为一幅地图着色时^①,具有公共边界的两个区域在传统上指定为不同的颜色。一种确保两个相邻的区域永远没有相同的颜色的方法是对每个区域都使用不同的颜色。不过,这种方法效率不高,而且在具有许多区域的地图上,可能难以区分相似的颜色。另一种方法是,应当尽可能地使用少数几种颜色。考虑这样的问题:确定可以用来给一幅地图着色的颜色的最小数目,使得相邻的区域永远没有相同的颜色。例如,对图 6-81 左侧地图来说,四色是足够的,但是三色就不够(读者应当验证这一点)。对图 6-81 右侧地图来说,三色是足够的(但是二色就不够)。

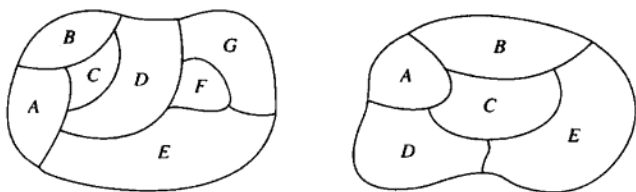


图 6-81 两幅地图

平面里的每幅地图都可以表示成一个图。为了建立这样的对应关系,地图的每个区域都表示成一个顶点。若两个顶点所表示的区域具有公共边界,则用边连接这两个顶点。只相交于一个点的两个区域不算是相邻的。这样所得到的图称为这个地图的对偶图。根据地图的对偶图的构造方式,显然在平面里的任何地图都具有可平面的对偶图。图 6-82 显示对应于图 6-81 所示地图的对偶图。

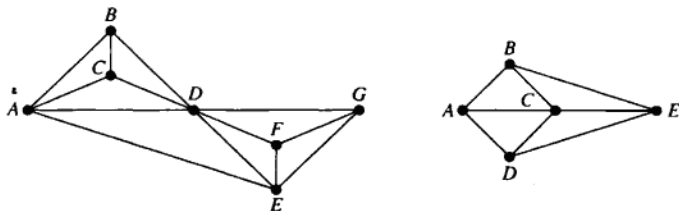


图 6-82 图 6-81 地图的对偶图

给地图的区域着色的问题等价于这样的问题:给对偶图的顶点着色,使得在对偶图里没有两个相邻的顶点具有相同的颜色。给出下面的定义。

定义 1 简单图的着色是对该图的每个顶点都指定一种颜色,使得没有两个相邻的顶点颜色相同。

通过对每个顶点都指定一种不同的颜色,就可以着色一个图。不过,对大多数的图来说,可以找到所用颜色数少于图中顶点数的着色。什么是所需要的最少颜色数?

定义 2 图的色数是着色这个图所需要的最少颜色数。

注意,求平面图의 色数等于是求平面地图着色所需要的最少颜色数,使得没有两个相邻的区域指定为相同的颜色。这个问题已经研究了 100 多年。数学里最著名的定理之一给出了它的答案。

定理 1 四色定理 平面图的色数不超过四。

① 假定地图里所有区域都是连通的。这样就消除了像密歇根这样的地理状况所引起的问题,因为密歇根州是由两个分离的半岛组成的。

 四色定理最早是作为猜想在19世纪50年代提出的。美国数学家肯尼思·阿佩尔和沃尔夫冈·黑肯最终在1976年证明了它。在1976年之前，发表过许多不正确的证明，其中的错误常常难以发现。另外，还尝试过画出需要超过四色的地图来构造反例，而这样做是无效的。(证明五色定理就没有这样困难。)

也许迄今为止在数学里最有名的错误证明就是伦敦的律师和业余的数学家艾尔弗雷德·肯普在1879年所发表的四色定理证明。数学家们一直认为他的证明是正确的，直到1890年珀西·希伍德发现了一处错误，才发现肯普的论证是不完全的。不过，事实证明，肯普的推理路线是阿佩尔和黑肯所给出的成功证明的基础。他们的证明依赖于计算机所完成的对各种情形的仔细分析。他们证明，若四色定理为假，则存在一个反例，它是大约2000种不同类型中的一种，然后他们证明这些类型都没有导致反例。在他们的证明中使用了1000多个小时的机时。计算机在证明过程中起到如此重要的作用，由此引发了广泛的争论。例如，在计算机程序里有没有导致不正确结果的错误？假如论证依赖于或许不可靠的计算机得出，那么它是不是真正的证明？

注意，四色定理只适用于平面图。在例2里将证明非可平面图可以有任意大的色数。

证明一个图的色数为 n 需要做两件事。首先必须证明：用 n 种颜色可以着色这个图。构造出这样的着色就可以完成这件事。其次证明：用少于 n 种颜色不能着色这个图。例1至例4说明如何求出色数。

例1 图6-83所示的图 G 和 H 的色数是什么？

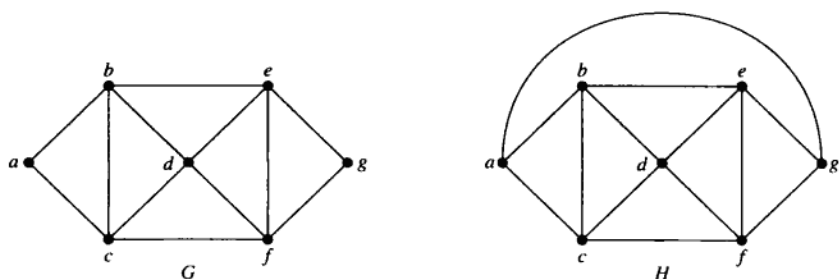


图6-83 简单图 G 和 H

 **解** 图 G 的色数至少为3，因为顶点 a 、 b 和 c 必须为不同的颜色。为了看出是否可以三种颜色来对 G 着色，指定 a 为红， b 为蓝， c 为绿。于是，可以(而且必须)令 d 为红，因为它与 b 和 c 相邻。另外，可以(而且必须)令 e 为绿，因为它只与红色和蓝色顶点相邻，可以(而且必须)令 f 为蓝，因为它只与红色和绿色顶点相邻。最后，可以(而且必须)令 g 为红，因为它只与蓝色和绿色的顶点相邻。这样就产生出恰好使用三种颜色的 G 的着色，如图6-84所示。

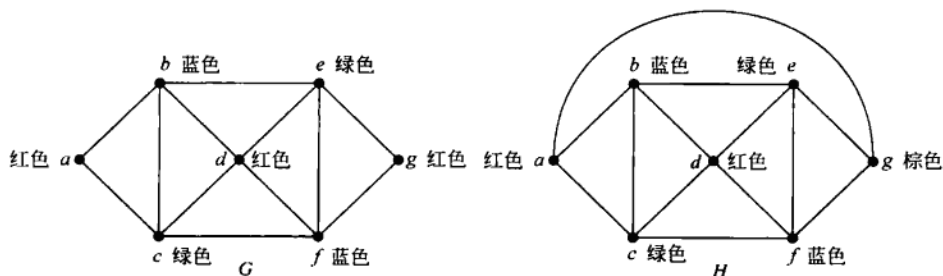


图6-84 图 G 和 H 的着色

图 H 是由图 G 和连接 a 与 g 的一条边所组成的。用三种颜色来着色 H 的任何尝试都必须遵循与着色 G 所用的同样的推理，不同之处是在最后阶段当除了 g 以外的所有顶点都已经着色时，因

为 g 与红色、蓝色和绿色顶点(在 H 里)相邻, 所以需要使用第四种颜色, 比如棕色。因此, H 的色数为四。图 6-84 右图显示 H 的一种着色。

例 2 K_n 的色数是什么?

解 通过给每个顶点指定一种不同的颜色, 用 n 种颜色可以构造 K_n 的着色。使用的颜色能否更少一些? 答案是不能。没有两个顶点可以指定相同的颜色, 因为这个图的每两个顶点都是相邻的。因此, K_n 的色数 $= n$ 。(回忆一下, 当 $n \geq 5$ 时 K_n 不是可平面图, 所以这个结果与四色定理并不矛盾。)在图 6-85 里显示使用五种颜色的 K_5 的着色。

例 3 完全偶图 $K_{m,n}$ 的色数是什么? 其中 m 和 n 都是正整数。

解 需要的颜色数似乎依赖于 m 和 n 。不过, 仅仅需要两种颜色。 m 个顶点为一种颜色, n 个顶点为另外一种颜色。因为边都只能连接 m 个顶点中的一个顶点与 n 个顶点中的一个顶点, 所以没有相邻的顶点具有相同的颜色。在图 6-86 里显示带有两种颜色的 $K_{3,4}$ 的着色。

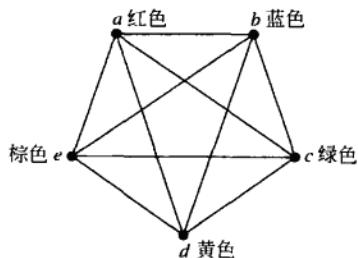


图 6-85 K_5 的着色

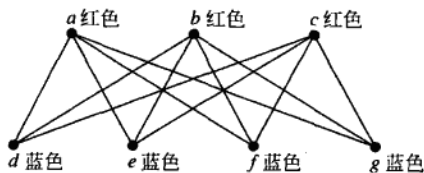


图 6-86 $K_{3,4}$ 的着色

例 4 图 C_n 的色数是什么($n \geq 3$)? (回忆一下, C_n 是带有 n 个顶点的圈图。)

解 首先考虑一些个别情形。首先, 设 $n=6$ 。挑选一个顶点并且把它着色成红色。在图 6-87 所示的 C_6 的平面画法里顺时针前进。必须给到达的下一个顶点指定第二种颜色, 比如蓝色。以顺时针方向继续下去, 可以令第三个顶点为红色, 第四个顶点为蓝色, 第五个顶点为红色。最后, 令第六个顶点为蓝色, 它与第一个顶点是相邻的。因此, C_6 的色数为 2。图 6-87 显示这里构造的着色。

其次, 设 $n=5$ 并且考虑 C_5 。挑选一个顶点并且令它为红色。顺时针前进, 必须给到达的下一个顶点指定第二种颜色, 比如蓝色。以顺时针方向继续下去, 可以令第三个顶点为红色, 第四个顶点为蓝色。第五个顶点既不能为红色也不能为蓝色, 因为它与第四个顶点和第一个顶点都相邻。所以, 对这个顶点就需要第三种颜色。注意, 假如以逆时针方向对顶点着色, 同样需要三种颜色。因此, C_5 的色数是 3。用 3 种颜色对 C_5 着色见图 6-87。

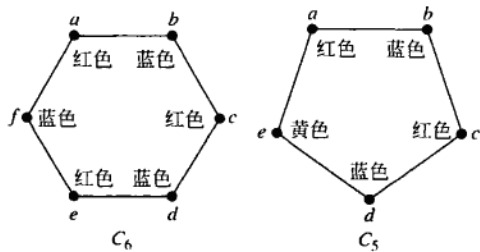


图 6-87 C_5 和 C_6 的着色

在一般情形里, 当 n 是偶数时, 对 C_n 着色需要

两种颜色。为了构造这样的着色, 简单地挑选一个顶点并且令它为红色。然后(利用图的平面表示)以顺时针方向绕图前进, 令第二个顶点为蓝色, 第三个顶点为红色, 依此类推。可以令第 n 个顶点为蓝色, 因为它与它相邻的两个顶点(即第 $n-1$ 个顶点和第一个顶点)都是红色。

当 n 是奇数而且 $n > 1$ 时, C_n 的色数为 3。为了看出这一点, 挑选一个初始顶点。为了只用两种颜色, 当以顺时针方向遍历这个图时, 必须交替使用颜色。不过, 所到达的第 n 个顶点与带不同颜色的两个顶点(第一个顶点和第 $n-1$ 个顶点)相邻。因此, 必须使用第三种颜色。



已知最好的求图的色数的算法(对图的顶点数来说)具有指数的最坏情形时间复杂性。即使求图的色数的近似值也是很难的。已经证明, 假如存在具有多项式最坏情形时间

复杂性的、可以达到 2 倍地近似图的色数的算法(即构造出一个不超过图的色数的两倍的界限),那么也存在具有多项式最坏情形时间复杂性的求图的色数的算法。

6.8.2 图着色的应用

图着色在与调度和分配有关的问题中具有多种应用。(注意,由于不知道图着色的有效算法,所以这并不能得出调度和分配的有效算法。)在这里将给出这样的应用的例子。第一个应用是用来安排期末考试。

例 5 安排期末考试 如何安排一所大学里的期末考试,使得没有学生同时要考两门试?

解 这样的安排问题可以用图模型来解决,用顶点表示科目,若有学生要考两门试,则在表示考试科目的两个顶点之间有边。用不同颜色来表示期末考试的每个时间段。考试的安排就对应于所关联的图的着色。

例如,假定要安排七门期末考试。假定科目从 1 到 7 编号。假定下列各对科目的考试有学生要都参加: 1 和 2, 1 和 3, 1 和 4, 1 和 7, 2 和 3, 2 和 4, 2 和 5, 2 和 7, 3 和 4, 3 和 6, 3 和 7, 4 和 5, 4 和 6, 5 和 6, 5 和 7, 以及 6 和 7。在图 6-88 里显示这组科目所关联的图。一种安排就是由这个图的一种着色来组成的。

因为这个图的色数为 4(读者应当验证这一点),所以需要四个时间段。图 6-89 显示使用了四种颜色的这个图的着色以及所关联的调度。

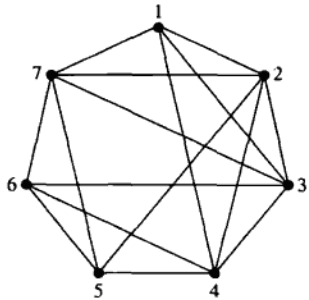


图 6-88 表示期末考试安排的图

现在考虑对电视频道的分配。

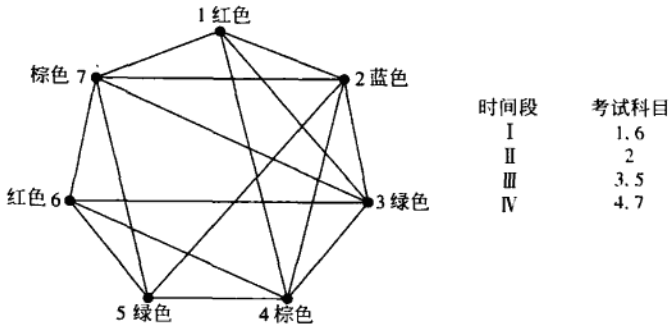


图 6-89 用着色来安排期末考试

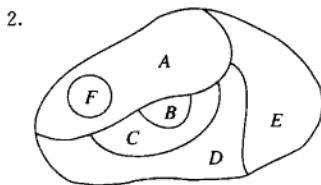
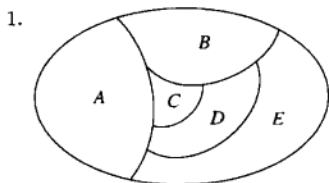
例 6 频率分配 把频道 2 到 13 分配给在北美洲的电视台,要避免 150 英里之内的两家电视台在相同频道上播出。如何用图为频道分配建模?

解 这样构造一个图:给每个电视台指定一个顶点。若两个电视台彼此位于 150 英里以内,则用边连接这两个顶点。频道指派就对应于这个图的着色,其中每种颜色表示一个不同的频道。

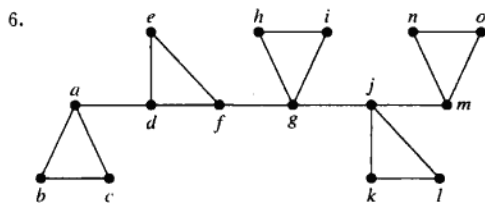
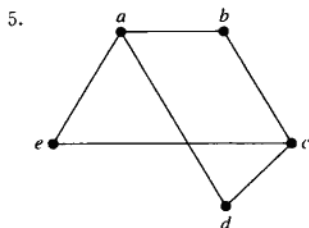
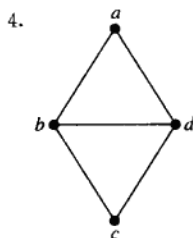
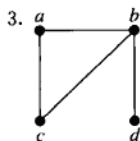
例 7 变址寄存器 在有效的编译器里,当把频繁地使用的变量暂时保存在中央处理单元的变址寄存器里,而不是保存在常规内存里时,可以加速循环的执行。对于给定的循环来说,需要多少个变址寄存器?可以用图着色模型来讨论这个问题。为了建立这个模型,设图的每个顶点表示循环里的一个变量。若在循环执行期间两个顶点所表示的变量必须同时保存在变址寄存器里,则在这两个顶点之间有边。所以,这个图的色数就给出所需要的变址寄存器数,因为当表示变量的顶点在图中相邻时,就必须给这些变量分配不同的寄存器。

练习

在练习 1~2 里, 构造所示地图的对偶图。然后求给这个地图着色使得相邻的两个区域都没有相同的颜色所需要的颜色数。



在练习 3~6 里, 求给定的图的色数。



7. 哪些图具有色数 1?

8. W_n 的色数是什么?

9. 假定除了科目 Math 115 与 CS 473, Math 116 与 CS 473, Math 195 与 CS 101, Math 195 与 CS 102, Math 115 与 Math 116, Math 115 与 Math 185, Math 185 与 Math 195 以外, 其他每种组合里都有学生要参加组合里所有科目的考试, 请使用最少个数的不同时间段, 来为 Math 115, Math 116, Math 185, Math 195, CS 101, CS 102, CS 273 和 CS 473 安排期末考试日程表。

10. 数学系有六个委员会, 都是每月开一次会。假定委员会是 $C_1 = \{\text{阿林豪斯, 布兰德, 沙斯拉夫斯基}\}$, $C_2 = \{\text{布兰德, 李, 罗森}\}$, $C_3 = \{\text{阿林豪斯, 罗森, 沙斯拉夫斯基}\}$, $C_4 = \{\text{李, 罗森, 沙斯拉夫斯基}\}$, $C_5 = \{\text{阿林豪斯, 布兰德}\}$ 和 $C_6 = \{\text{布兰德, 罗森, 沙斯拉夫斯基}\}$, 那么怎样安排才能确保没有人同时参加两个会议。

11. 图的边着色是指对各边指定颜色, 使得关联到相同顶点的边指定不同的颜色。图的边色数是在该图的边着色里可以使用的最少颜色数。

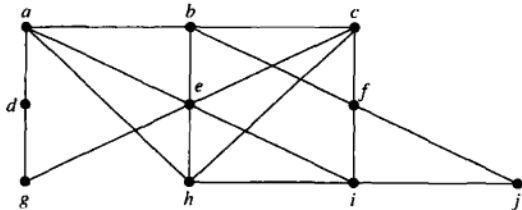
12. 求练习 3~6 里每个图的边色数。

13. 七个变量出现在计算机程序的循环里。这些变量以及必须保存它们的计算步骤是: t : 步骤 1 到 6; u : 步骤 2; v : 步骤 2 到 4; w : 步骤 1, 3 和 5; x : 步骤 1 和 6; y : 步骤 3 到 6; 以及 z : 步骤 4 和 5。在执行期间需要多少个不同的变址寄存器来保存这些变量?

下面的算法可以用来为简单图着色。首先, 以度递减的顺序列出顶点 $v_1, v_2, \dots, v_n$ 使得 $\deg(v_1) \geq \deg(v_2) \geq \dots \geq \deg(v_n)$ 。把颜色 1 指定给 v_1 和在表中不与 v_1 相邻的下一个顶点(若存在一个这样的顶点), 并且继续指定给每一个在表中不与已经指定了颜色 1 的顶点相邻的顶点。然后把颜色 2 指定给表中还没有着色的第一个顶点。继续把颜色 2 指定给那些在表中还没有着色, 而且不与指定了颜色 2 的顶点相邻的顶点。若

还有未着色的顶点, 则指定颜色 3 给表中还没有着色的第一个顶点, 并且用颜色 3 继续对还没有着色, 而且不与指定了颜色 3 的顶点相邻的那些顶点着色。继续这个过程直到所有顶点都着色为止。

13. 用这个算法构造下图的着色。



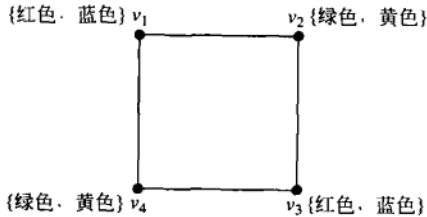
*14. 证明: 这个算法所产生的着色数可能比着色一个图所需的颜色数更多。

如果一个连通图 G 的色数为 k , 但是对于 G 的任意一条边 e , 从 G 中删掉边 e 后得到的新图的色数都是 $k-1$, 则称 G 为着色 k 关键的。

15. 证明: 只要 n 是正的奇数且 $n \geq 3$, 那么 W_n 就是着色 4 关键的。

16. 证明: 如果图 G 为着色 k 关键的, 那么 G 中的各个顶点的度至少是 $k-1$ 。

图 G 的 k 重着色是对 G 的顶点指定含有 k 种不同颜色的集合, 使得相邻的顶点不具有相同的颜色。用 $\chi_k(G)$ 来表示使得 G 有用 n 种颜色的 k 重着色的最小正整数 n 。例如, $\chi_2(C_4) = 4$ 。为了看出这一点, 注意如图所示, 只用四种颜色, 就可以对 C_4 的每个顶点指定两种颜色, 使得两个相邻顶点不具有相同的颜色。另外, 少于四种颜色是不够的, 因为顶点 v_1 和 v_2 每个都必须指定两种颜色, 而且不能对 v_1 和 v_2 指定相同颜色。(关于 k 重着色的更多信息, 见[MiRo91]。)



*17. 设 G 和 H 是图 6-82 所示的图。求

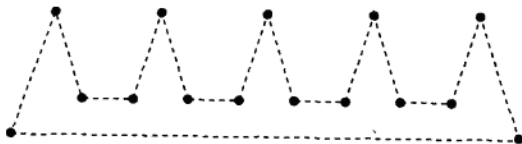
- a) $\chi_2(G)$
- b) $\chi_2(H)$
- c) $\chi_3(G)$
- d) $\chi_3(H)$

18. 移动广播(或蜂窝)电话的频率是按地段分配的。每个地段分配一组该地段里的车辆所使用的频率。在产生干扰问题的地段里不能使用相同频率。解释一下如何用 k 重着色来对一个地区里的每个移动广播地段分配 k 种频率。

著名的艺术馆问题求需要多少名保安才能看全艺术馆的所有部分, 这里艺术馆是一个 n 边形的边界及它所围的内部。为了更精确地描述这个问题, 需要一些术语。简单多边形 P 边界上或 P 内部的点 x 覆盖或看见 P 边界上或 P 内部的点 y , 如果线段 xy 上所有的点都在 P 边界上或 P 内部。如果对于 P 边界上或 P 内部的每一个点 y , 都能够在一个点的集合中找到一个看见 y 的点 x , 就说这个点的集合是简单多边形的看守集。把看守简单多边形 P 所需的最少点数的看守集记为 $G(P)$ 。艺术馆问题求的就是一个函数 $g(n)$, 它是所有 n 顶点的简单多边形 P 的看守集 $G(P)$ (的元素个数) 的最大值(考虑 n 顶点的简单多边形 P 的各种形态——译者注)。也就是说, $g(n)$ 是一个最小的正整数, 使得一个 n 顶点的简单多边形 P 保证可以被 $g(n)$ 个或更少的保安看守。

*19. 证明: $g(5) = 1$, 即所有的五边形都能够被一个点守卫。[提示: 先分出有 0 个、1 个、2 个顶点的内角大于 180 度, 然后再说明在各种情况下一个保安都足够了。]

*20. 证明: $g(n) \geq \lceil n/3 \rceil$ 。[提示: 考虑具有 $3k$ 个顶点和 k 个齿尖的梳子状的多边形, 就像如图所示的 15 边形一样。]



关键术语和结果

术语

无向边：与集合 $\{u, v\}$ 关联的边，其中 u 和 v 都是顶点。

有向边：与有序对 (u, v) 关联的边，其中 u 和 v 都是顶点。

多重边：连接同样一对顶点的不同的边。

环：连接一个顶点与它自身的边。

无向图：一组顶点以及连接这些顶点的一组无向边。

简单图：没有多重边和环的无向图。

多重图：可能包含多重边但不包含环的无向图。

伪图：可能包含多重边和环的无向图。

有向图：一组顶点以及连接这些顶点的一组有向边。

有向多重图：可能包含多重有向边的带有向边的图。

相邻：若两个顶点之间有边则它们是相邻的。

关联：若一个顶点是一条边的端点则那条边关联那个顶点。

$\deg(v)$ (无向图里顶点 v 的度)：与 v 关联的边的数目。

$\deg^-(v)$ (带有向边的图里顶点 v 的入度)：以 v 作为终点的边的数目。

$\deg^+(v)$ (带有向边的图里顶点 v 的出度)：以 v 作为起点的边的数目。

带有向边的图的底图：通过忽略边的方向所获得的无向图。

K_n (n 个顶点的完全图)：带 n 个顶点的无向图，其中每对顶点都用一条边连接。

偶图：顶点集划分成两个子集合 V_1 和 V_2 的图，使得每条边都连接 V_1 里的顶点和 V_2 里的顶点。

$K_{m,n}$ (完全偶图)：顶点集划分成 m 个元素的子集和 n 个元素的子集，使得两个顶点被一条边所连接，当且仅当一个顶点属于第一个子集而另外一个顶点属于第二个子集。

C_n (大小为 n 的圈图)， $n \geq 3$ ：带有 n 个顶点 $v_1, v_2, \dots, v_n$ 和边 $\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$ 的图。

W_n (大小为 n 的轮图)， $n \geq 3$ ：通过向 C_n 添加一个顶点以及从这个顶点到 C_n 里原来每个顶点的一条边所获得的图。

Q_n (n 立方体图)， $n \geq 1$ ：用 2^n 个长度为 n 的位串作为顶点，边连接恰好相差一位的每对位串的图。

孤立点：0度顶点。

悬挂点：1度顶点。

正则图：所有顶点都有相同的度的图。

图 $G=(V, E)$ 的子图：图 (W, F) ，其中 W 是 V 的子图而 F 是 E 的子图。

$G_1 \cup G_2$ (G_1 与 G_2 的并图)：图 $(V_1 \cup V_2, E_1 \cup E_2)$ ，其中 $G_1=(V_1, E_1)$ 而 $G_2=(V_2, E_2)$ 。

邻接矩阵：利用顶点的相邻关系来表示图的矩阵。

关联矩阵：利用边与顶点的关联关系来表示图的矩阵。

同构的简单图：对简单图 $G_1=(V_1, E_1)$ 和简单图 $G_2=(V_2, E_2)$ 来说，若存在从 V_1 到 V_2 的一一对应 f ，使得对所有属于 V_1 的 v_1 和 v_2 来说， $\{f(v_1), f(v_2)\} \in E_2$ 当且仅当 $\{v_1, v_2\} \in E_1$ ，则 G_1 与 G_2 是同构的。

同构不变量：同构的图都有或都没有的性质。

无向图里从 u 到 v 的通路：一条或多条边的序列 $e_1, e_2, \dots, e_n$ ，其中对 $i=0, 1, \dots, n$ 来说， e_i 关联着 $\{x_i, x_{i+1}\}$ ，其中 $x_0=u$ 而 $x_{n+1}=v$ 。

有向图里从 u 到 v 的通路：一条或多条边的序列 $e_1, e_2, \dots, e_n$ ，其中对 $i=0, 1, \dots, n$ 来说， e_i 关联着 (x_i, x_{i+1}) ，其中 $x_0=u$ 而 $x_{n+1}=v$ 。

简单通路：不多次包含一条边的通路。

回路：在相同顶点处开始与结束的通路，通路长度 $n \geq 1$ 。

连通图：在图中每对顶点之间都有通路的无向图。

图 G 的连通分支：图 G 的最大连通子图。

强连通有向图：从每个顶点到每个顶点都存在有向通路的有向图。

有向图 G 的强连通分支： G 的最大强连通子图。

欧拉回路：恰好包含图的每条边一次的回路。

欧拉通路：恰好包含图的每条边一次的通路。

哈密顿通路：简单图 $G=(V, E)$ 里的通路 $x_0, x_1, \dots, x_n$ ，使得 $\{x_0, x_1, \dots, x_n\}=V$ 而且对 $0 \leq i < j \leq n$ 来说有 $x_i \neq x_j$ 。

哈密顿回路：简单图里的回路 $x_0, x_1, \dots, x_n, x_0$ ，使得 $x_0, x_1, \dots, x_n$ 是哈密顿通路。

带权图：为各边指定数字的图。

最短通路问题：确定带权图里的通路以使得这条通路里的边的权之和在所规定的顶点之间的所有通路上是最小值这样的问题。

旅行商问题：求访问图的每个顶点恰好一次的、总长度最短的回路的问题。

可平面图：可以画在平面上而没有边交叉的图。

可平面图的平面表示面：该图的平面表示把平面所分割成的区域。

初等细分：删除无向图的边 $\{u, v\}$ 而且添加新顶点 w 以及边 $\{u, w\}$ 和边 $\{w, v\}$ 。

同胚：若两个无向图是从同一个无向图通过一系列初等细分来获得的，则它们同胚。

图着色：给图的顶点指定颜色，使得相邻的两个顶点没有相同的颜色。

色数：在图的着色里所需要的最少颜色数。

结果

在连通多重图里存在欧拉回路当且仅当每个顶点都有偶数度。

在连通多重图里存在欧拉通路当且仅当至多两个顶点有奇数度。

迪克斯特拉算法：在带权图里求出两个顶点之间最短通路的过程(见 6.6 节)。

欧拉公式： $r=e-v+2$ ，其中 r 、 e 和 v 分别是可平面图的平面表示的面数、边数和顶点数。

库拉图斯基定理：图是不可平面的当且仅当它包含同胚于 $K_{3,3}$ 或 K_5 的子图。(其证明超出本书范围。)

四色定理：每个平面图都可以用不超过四种颜色来着色。(其证明远远超出本书范围!)

复习题

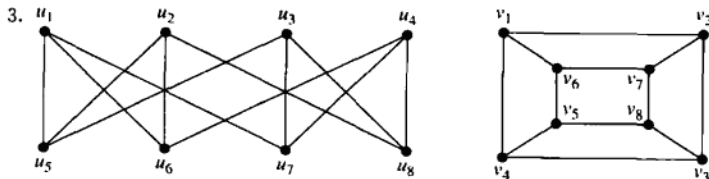
1. a) 定义：简单图，多重图，伪图，有向图，有向多重图。
b) 用例子说明：如何用 a) 里每种类型的图来建模。例如，解释一下如何为计算机网络或飞行航线的不同方面来建模。
2. 给出如何用图建模的至少四个例子。
3. 在无向图里，顶点度数之和与该图里边数之间的关系是什么？解释这个关系为什么成立。
4. 为什么在无向图里奇数度顶点的个数必然是偶数？
5. 在有向图里顶点的入度之和与出度之和之间的关系是什么？解释这个关系为什么成立。
6. 描述下列图族。
 - a) K_n ，在 n 个顶点上的完全图
 - b) $K_{m,n}$ ，在 m 和 n 个顶点上的完全偶图
 - c) C_n ，带 n 个顶点的圈图
 - d) W_n ，大小为 n 的轮图
 - e) Q_n ， n 立方体
7. 在练习 6 的图族里，每个图有多少个顶点和多少条边？
8. a) 什么是偶图？
b) 图 K_n 、 C_n 和 W_n 中哪些是偶图？

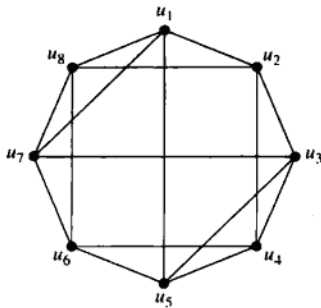
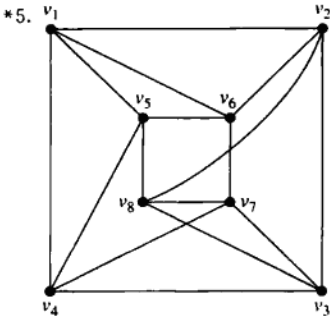
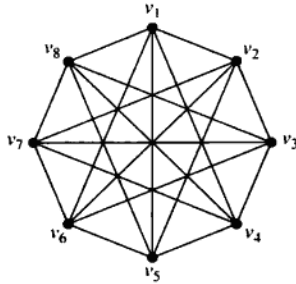
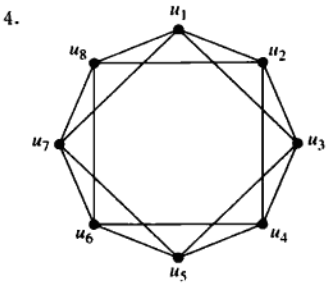
- c) 你如何确定无向图是否为偶图?
9. a) 描述用来表示图的三种不同方法。
b) 画出至少带 5 个顶点和 8 条边的简单图。说明如何用你在 a) 里所描述的方法来表示它。
10. a) 两个简单图是同构的是什么意思?
b) 对于简单图的同构来说, 不变量是什么意思? 给出至少五个这样的不变量的例子。
c) 给出带有相同的顶点数、边数和顶点度但不同构的两个简单图的例子。
d) 是否有一组已知的不变量可以用来有效地确定两个简单图是否同构?
11. a) 图是连通的是什么意思?
b) 什么是图的连通分支?
12. a) 解释一下如何用邻接矩阵来表示图。
b) 如何用邻接矩阵来确定从图 G 的顶点集到图 H 的顶点集的函数是否是同构的?
c) 如何用图的邻接矩阵来确定在图的两个顶点之间长度为 r 的通路数? 其中 r 是正整数。
13. a) 定义无向图里的欧拉回路和欧拉通路。
b) 描述著名的哥尼斯堡七桥问题, 并且解释一下如何利用欧拉回路来重新叙述它。
c) 如何确定无向图是否具有欧拉通路?
d) 如何确定无向图是否具有欧拉回路?
14. a) 定义简单图里的哈密顿回路。
b) 给出一些性质, 这些性质蕴含着简单图没有哈密顿回路。
15. 给出至少两个可以通过求出带权图里最短通路来解决的问题的例子。
16. a) 描述求在带权图两个顶点之间的最短通路的迪克斯特拉算法。
b) 画出至少带 10 个顶点和 20 条边的带权图。用迪克斯特拉算法求出在图中你所选择的两个顶点之间的最短通路。
17. a) 图是可平面的是什么意思?
b) 给出不可平面的简单图的例子。
18. a) 平面图的欧拉公式是什么?
b) 如何用平面图的欧拉公式来证明简单图是不可平面的?
19. 叙述关于图的可平面性的库拉图斯基定理, 并且解释一下它如何刻画了哪些图是可平面的。
20. a) 定义图的色数。
b) 当 n 是正整数时, 图 K_n 的色数是什么?
c) 当 n 是大于 2 的正整数时, 图 C_n 的色数是什么?
d) 当 m 和 n 都是正整数时, 图 $K_{m,n}$ 的色数是什么?
21. 叙述四色定理。有没有不能用四种颜色来着色的图?
22. 解释一下在建模里可以如何使用图的着色。至少举两个不同的例子。

补充练习

1. 一个带 100 个顶点的 50 正则图有多少条边?
2. K_3 有多少种非同构的子图?

在练习 3~5 里, 确定所给的成对的图是否同构。





完全 m 部图 $K_{n_1, n_2, \dots, n_m}$ 的顶点划分成 m 个子集合, 各有 $n_1, n_2, \dots, n_m$ 个元素, 而且顶点相邻当且仅当它们属于这个划分的不同子集合。

6. 画出下列图。

a) $K_{1,2,3}$

b) $K_{2,2,2}$

c) $K_{1,2,2,3}$

*7. 完全 m 部图 $K_{n_1, n_2, \dots, n_m}$ 有多少个顶点和多少条边?

*8. a) 证明或反驳: 在至少有两个顶点的有穷简单图里, 总是存在两个度数相同的顶点。

b) 对有穷多重图做与(a)里同样的事。

设 $G=(V, E)$ 是简单图。顶点集合 V 的子集合 W 的导出子图是图 (W, F) , 其中边集合 F 包含着 E 里的一条边当且仅当这条边的两个端点都属于 W 。

9. 考虑 6.4 节图 6-42 所示的图。求下列顶点的导出子图

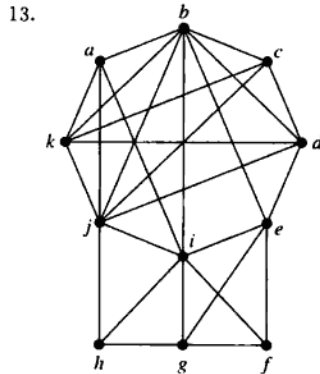
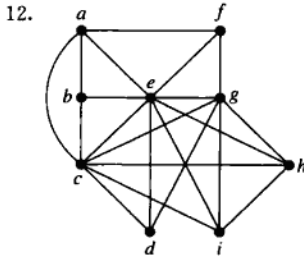
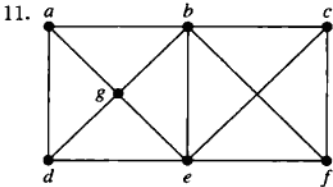
a) $\{a, b, c\}$

b) $\{a, e, g\}$

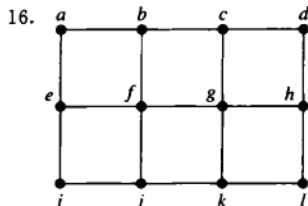
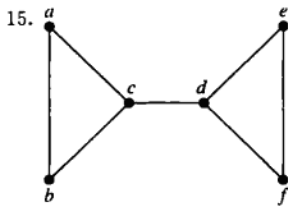
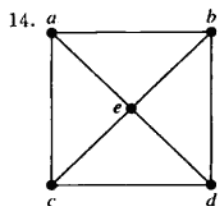
c) $\{b, c, f, g, h\}$

10. 设 n 是正整数。证明: K_n 的顶点集的非空子集的导出子图是完全图。

~~定义~~ 简单无向图里的团是一个完全子图, 它不包含在任何更大的完全子图里。在练习 11~13 里, 求所给的图的所有团。



简单图里顶点的支配集是顶点的一个集合,使得其他每个顶点都与这个集合里至少一个顶点是相邻的。带最少顶点数的支配集称为最小支配集。在练习 14~16 里,求所给的图的最小支配集。

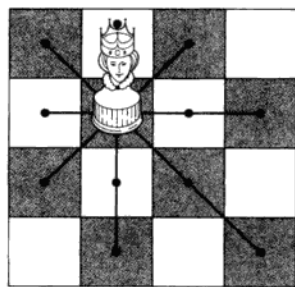


简单图可用来确定在棋盘上控制整个棋盘的最少皇后数。一个 $n \times n$ 的棋盘具有 n^2 个格子。如下图所示,在所给位置里的皇后控制着同行、同列以及包含这个格子的两条斜线上的所有格子。与此对应的简单图具有 n^2 个顶点,每个顶点表示一个格子,而且若一个顶点所表示的格子里的皇后控制着另外一个顶点所表示的格子,则这两个顶点是相邻的。

17. 构造表示 $n \times n$ 棋盘的简单图,用边表示皇后对格子的控制,其中

a) $n=3$ b) $n=4$

18. 解释一下最小支配集的概念如何应用到确定控制 $n \times n$ 棋盘的最少皇后数的问题。



皇后所控制的格子

- **19. 求控制 $n \times n$ 棋盘的最少皇后数,其中

a) $n=3$ b) $n=4$ c) $n=5$

20. 假定 G_1 和 H_1 是同构的而且 G_2 和 H_2 是同构的。证明或反证: $G_1 \cup G_2$ 和 $H_1 \cup H_2$ 是同构的。

21. 证明:下列性质是同构的简单图都有或都没有的不变量。

a) 连通性 b) 哈密顿回路的存在性
c) 欧拉回路的存在性 d) 有交叉数 C
e) 有 n 个孤立顶点 f) 是偶图

22. 如何从 G 的邻接矩阵求 $\bar{G}$ 的邻接矩阵? 其中 G 是简单图。

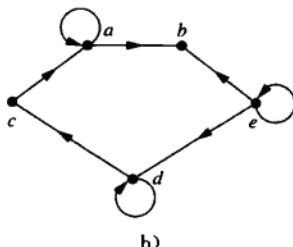
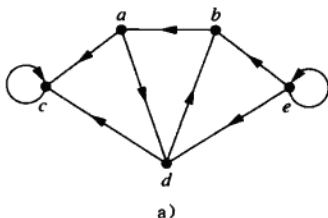
23. 有多少种非同构的、带有 4 个顶点的连通简单偶图?

- *24. 有多少种非同构的简单连通图带有 5 个顶点并且

a) 没有任何顶点的度超过 2?
b) 色数等于 4?
c) 非可平面的?

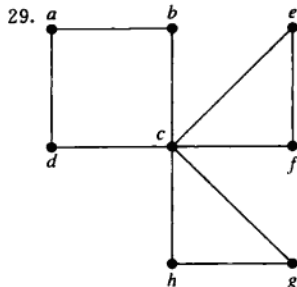
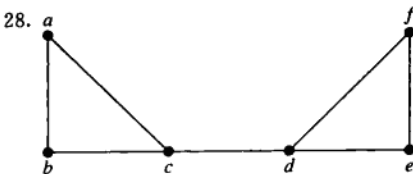
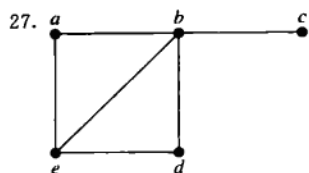
若有向图与它的逆同构,则它是自逆的。

25. 确定下列图是否为自逆的。



26. 证明:若有向图 G 是自逆的而且 H 是同构于 G 的有向图,则 H 也是自逆的。

无向简单图的定向就是指定它的各边的方向,使得所得到的有向图是强连通的。当无向图有定向时,这个图称为可定向的。在练习 27~29 里,确定给定的图是否可定向。



30. 因为在城市中心区交通流量正在增长, 所以交通工程师正在计划把目前所有双向街道都变成单行街道。解释一下如何为这个问题建模。
- *31. 证明: 若一个图具有割边, 则它不是可定向的。
- 竞赛图是简单有向图, 使得若 u 和 v 是图里不同的顶点, 则 (u, v) 和 (v, u) 中恰好有一个是图中的边。
32. 有多少种不同的带 n 个顶点的竞赛图?
33. 在竞赛图里一个顶点的入度与出度之和是什么?
- *34. 证明: 每个竞赛图都有哈密顿通路。
35. 给定鸡群里的两只鸡, 其中一只占优势。这样就定义了这个鸡群的啄食次序。如何用竞赛图来为啄食次序建模?
36. 假定 G 是带 $2k$ 个奇数度顶点的连通多重图。证明: 存在 k 个子图, 它们的并图是 G , 其中每个子图都有欧拉通路并且任何两个子图都没有公共边。[提示: 向该图添加 k 条边, 连接成对的奇数度顶点, 并且利用这个变大了的图的欧拉回路。]
- *37. 设 G 是带有 n 个顶点的简单图。 G 的带宽表示成 $B(G)$, 它是 $\max\{|i-j| \mid a_i \text{ 与 } a_j \text{ 是相邻的}\}$ 在 G 的顶点 $a_1, a_2, \dots, a_n$ 的所有排列上所取的最小值。即带宽是赋给相邻顶点的下标的最大差值在顶点的所有列表上所取的最小值。求下列图的带宽。
- a) K_5 b) $K_{1,3}$ c) $K_{2,3}$ d) $K_{3,3}$ e) Q_3 f) C_5
- *38. 连通简单图的两个不同顶点 v_1 和 v_2 之间的距离是在 v_1 和 v_2 之间的最短通路的长度(边数)。图的半径是从顶点 v 到其他顶点的最大距离在所有顶点 v 上所取的最小值。图的直径是在两个不同顶点之间的最大距离。求下列图的半径和直径。
- a) K_6 b) $K_{4,5}$ c) Q_3 d) C_5
- *39. a) 证明: 若简单图 G 的直径至少为 4, 则它的补图 $\bar{G}$ 的直径不超过 2。
b) 证明: 若简单图 G 的直径至少为 3, 则它的补图 $\bar{G}$ 的直径不超过 3。
- *40. 假定一个多重图有 $2m$ 个奇数度顶点。证明: 任何包含该图中每条边的回路, 必然至少包含 m 条边超过一次。
41. 求 6.6 节图 6-64 中在顶点 a 与 z 之间的次最短通路。
42. 设计一个算法, 它在简单连通带权图里, 求在两个顶点之间的最短通路。
43. 求 6.6 节图 6-65 中在顶点 a 与 z 之间经过顶点 e 的最短通路。
44. 设计一个算法, 它在简单连通带权图里, 求在两个顶点之间经过第三个指定顶点的最短通路。
- *45. 证明: 若 G 是至少带 11 个顶点的简单图, 则或者 G 不是可平面的, 或者 $\bar{G}$ 不是可平面的, $\bar{G}$ 是 G 的补图。
- *46. 若在图中一组顶点的集合里任何两个顶点都不相邻, 则这个顶点集合称为独立的。图的独立数是图中顶点独立集里的最大顶点数。
- *47. 证明: 一个简单图里的顶点数小于或等于这个图的独立数与色数之积。
48. 证明: 一个图的色数小于或等于 $v-i+1$, 其中 v 是这个图的顶点数, i 是这个图的独立数。
49. 假定为了生成带有 n 个顶点的随机简单图, 首先选择满足 $0 \leq p \leq 1$ 的实数 p 。对 $C(n, 2)$ 对不同顶点中的每一对, 都生成一个在 0 与 1 之间的随机数 x 。若 $0 \leq x \leq p$, 则用一条边连接这两个顶点; 否则就不连接

这两个顶点。

- a) 生成带有 m 条边的图的概率是什么? 其中 $0 \leq m \leq C(n, 2)$ 。
 b) 若以概率 p 来包含每一条边, 则在随机生成的带有 n 个顶点的图中, 边数的期望值是什么?
 c) 证明: 若 $p=1/2$, 则以相等的概率生成每一个带 n 个顶点的简单图。

每当向简单图添加更多的边(不添加顶点)时, 都还保持的性质称为单调增的, 每当从简单图删除边(不删除顶点)时, 都还保持的性质称为单调减的。

50. 对下列每个性质来说, 确定它是否为单调增的并且确定它是否为单调减的。

- a) 图 G 是连通的 b) 图 G 不是连通的 c) 图 G 有欧拉回路 d) 图 G 有哈密顿回路
 e) 图 G 是可平面的 f) 图 G 的色数为 4 g) 图 G 的半径为 3 h) 图 G 的直径为 3

51. 证明: 图的性质 P 是单调增的当且仅当图的性质 Q 是单调减的, 其中 Q 是不具有性质 P 的性质。

**52. 假定 P 是简单图的单调增的性质。证明: 带 n 个顶点的随机图有性质 P 的概率是挑选一条边属于图的概率 p 的单调非减函数。

计算机题目

按下述给定的输入与输出写出程序。

1. 给定无向图的各边所关联的顶点对, 确定每个顶点的度。
2. 给定有向图的各边所关联的有序顶点对, 确定每个顶点的入度和出度。
3. 给定简单图的边列表, 确定这个图是否为偶图。
4. 给定图的各边所关联的顶点对, 构造这个图的邻接矩阵。
5. 给定图的邻接矩阵, 列出这个图的各边, 并且给出每条边出现的次数。
6. 给定无向图各边所关联的顶点对, 以及每条边出现的次数, 构造这个图的关联矩阵。
7. 给定无向图的关联矩阵, 列出它的各边, 并且给出每条边出现的次数。
8. 给定正整数 n , 通过产生图的邻接矩阵来生成无向图, 使得以相等的概率来生成所有的简单图。
9. 给定正整数 n , 通过产生图的邻接矩阵来生成有向图, 使得以相等的概率来生成所有的有向图。
10. 给定两个都带不超过 6 个顶点的简单图的边列表, 确定这两个图是否同构。
11. 给定图的邻接矩阵和正整数 n , 求顶点两两之间长度为 n 的通路数(产生对有向图和无向图来说都能工作的程序)。
- *12. 给定简单图的边列表, 确定它是否连通, 若它不连通, 则求连通分支数。
13. 给定多重图的各边所关联的顶点对, 确定它是否有欧拉回路, 若没有欧拉回路, 则确定它是否有欧拉通路。若存在欧拉通路或欧拉回路, 则构造这样的通路或回路。
- *14. 给定有向多重图的各边所关联的有序顶点对, 若存在欧拉通路或欧拉回路, 则构造这样的通路或回路。
- **15. 给定简单图的边列表, 产生一条哈密顿回路, 或者确定该图没有这样的回路。
- **16. 给定简单图的边列表, 产生一条哈密顿通路, 或者确定该图没有这样的通路。
17. 给定带权连通简单图的边及其权的列表, 以及该图中的两个顶点, 用迪克斯特拉算法求这两点间最短通路的长度。另外, 求出这条通路。
18. 给定无向图的边的表, 用 6.8 节练习里所给的算法求这个图的着色。
19. 给定学生及其注册课程的表, 构造期末考试日程表。
20. 给定各对电视台之间的距离, 为这些台分配频率。

计算和研究

利用计算程序或你所编写的程序来做下面的练习。

1. 显示带 4 个顶点的所有简单图。
2. 显示全套的带 6 个顶点的所有非同构的简单图。
3. 显示全套的带 4 个顶点的所有非同构的有向图。
4. 随机地生成 10 个不同的简单图, 每个带 20 个顶点, 使得每个这样的图都是以相等的概率来生成的。
5. 构造一种格雷码, 其中码字都是长度为 6 的位串。
6. 构造马在不同大小的棋盘上的周游路线。

7. 确定你在本组练习的练习 4 里生成的每个图是否为平面图。若你可以做到, 则确定每个非平面图的厚度。
8. 确定你在本组练习的练习 4 里生成的每个图是否连通。若有一个图不连通, 则确定这个图的连通分支数。
9. 随机地生成带 10 个顶点的简单图。当你生成了一个带欧拉回路的图时停止。显示这个图里的一个欧拉回路。
10. 随机地生成带 10 个顶点的简单图。当你生成了一个带哈密顿回路的图时停止。显示这个图里的一个哈密顿回路。
11. 求你在本组练习的练习 4 里所生成的每个图的色数。
- **12. 求旅行推销员访问美国 50 个州的每个首府所能采取的最短路线, 飞机在各城市之间直线通行。
- *13. 对每个不超过 10 的正整数 n 来说, 估计随机生成的带 n 个顶点的简单图连通的概率, 方法是生成一组随机简单图并且确定每个图是否连通。
- **14. 研究这个问题: 确定 $K(7, 7)$ 的交叉数是否为 77, 79 或 81。已知它等于这三个数中的一个。

写作题目

利用本书以外的资料, 就下列问题写作短文。

1. 描述一下在 1900 年以前图论的起源和发展。
2. 讨论一下图论在生态系统研究中的应用。
3. 讨论一下图论在社会学和心理学中的应用。
4. 讨论一下通过研究网络图的性质可以了解到什么?
5. 描述一下给定一个图的顶点和边, 在纸面或屏幕上画出这个图的算法。
6. 一个输入、显示和操纵各种图的软件工具应当具有什么功能? 现有的工具都具有这些功能中的哪些?
7. 描述一下确定两个图是否同构的一些可用算法和这些算法的计算复杂性。目前已知最有效的算法是什么?
8. 描述一下如何用欧拉通路来帮助确定 DNA 序列。
9. 定义一下德布鲁因序列并且讨论它们如何出现在应用里。解释如何用欧拉回路来构造德布鲁因序列。
10. 描述一下中国邮递员问题并且解释如何解决这个问题。
11. 描述一下蕴含着图具有哈密顿回路的一些不同条件。
12. 描述一下用来解决旅行推销员问题的几个不同策略和算法。
13. 描述一下确定一个图是否可平面的几个不同算法。每个算法的计算复杂性是什么?
14. 在建模中, 有时把大规模集成电路(VLSI)图嵌入到一本书中, 让顶点都在书脊上而边都在不同的书页上(在同一页之内边与边互不相交)。定义一下图的书页数(book number, 即上述做法所需要的最少页数——译者注)并且对 $n=3, 4, 5$ 和 6 求包括 K_n 在内的各种图的书页数。
15. 描述一下四色定理的历史。
16. 描述一下在四色定理的证明中计算机所扮演的角色。如何肯定一个依赖计算机的证明是正确的?
17. 就是否产生最少颜色的着色以及复杂性而言, 描述并比较一下给图着色的几个不同算法。
18. 解释一下在各种不同模型里如何使用图的多重着色。
19. 解释一下在带特定性质的图的非构造性存在性证明中如何使用随机图理论。

树

不包含简单回路的连通图称为树。早在 1857 年英国数学家亚瑟·凯莱就用树去计数某些类型的化合物。本章中的例子将说明从那时起，树已经被用来解决各种学科分支里的问题。

树在计算机科学里特别有用，尤其是在算法中。例如，用树来构造求出项在表中位置的有效算法；可以用树构造有效编码以节省数据存储和传输成本，比如赫夫曼编码；可以用树来研究诸如跳棋和象棋这样的博弈，并且可以帮助确定进行这些博弈的取胜策略；可以用树来为通过一系列决策而完成的过程建立模型。构造这些模型可以帮助确定比如排序算法这样的基于一系列决策的算法的计算复杂性。

通过深度优先搜索或宽度优先搜索，可以系统地遍历图的顶点，构造出一棵包括每个顶点的树来。通过深度优先搜索来探索图的顶点，也称为回溯，允许系统地搜索五花八门的问题的解，比如确定在棋盘上如何放置 8 个王后使得这些王后不能互相攻击。

可以给树的边赋权来为许多问题建立模型。例如，用加权树可以开发出构造网络的算法，这些网络含有最便宜的连接不同网络节点的电话线集合。

7.1 概述

第 6 章说明了如何用图来建立模型和解决许多问题。本章将集中讨论称为树的一种特殊类型的图，之所以这样命名是因为这样的图就像是树。例如，家族树是表示族谱图的图。家族树用顶点表示家族成员并且用边表示亲子关系。图 7-1 显示瑞士数学世家伯努利家族的家族树。表示家族树的无向图是树的一个例子。

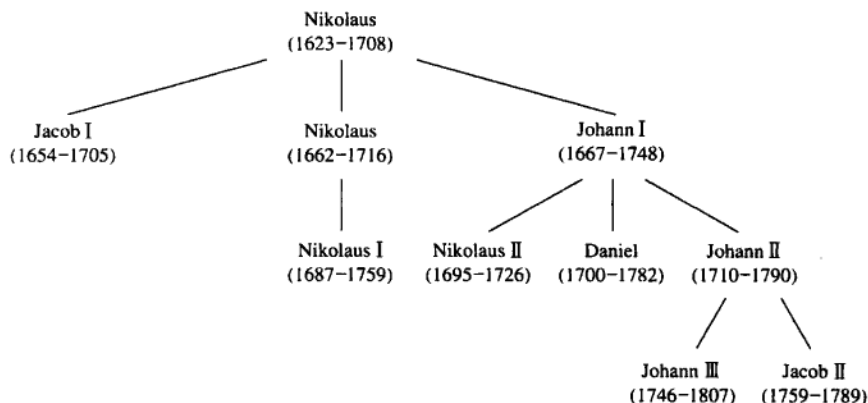


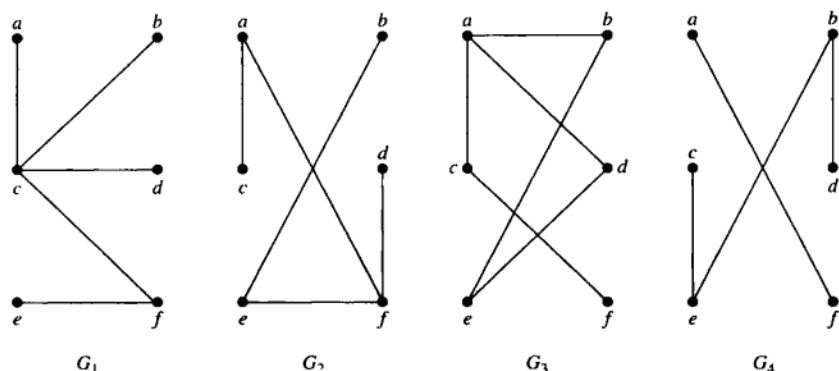
图 7-1 伯努利数学世家

定义 1 树是没有简单回路的连通无向图。

因为树没有简单回路，所以树不含多重边或环。因此任何树都必然是简单图。

例 1 在图 7-2 所示的图中，哪些图是树？

解 G_1 和 G_2 是树，因为都是没有简单回路的连通图。 G_3 不是树，因为 e, b, a, d, e 是这个图中的简单回路。最后， G_4 不是树，因为它不连通。

图 7-2 G_1 和 G_2 是树, G_3 和 G_4 不是树

任何一个不包含简单回路的连通图都是树。不含简单回路但不一定连通的图是什么? 这些图称为森林, 而且具有这样的性质: 它们的每个连通分支都是树。图 7-3 显示一个森林。

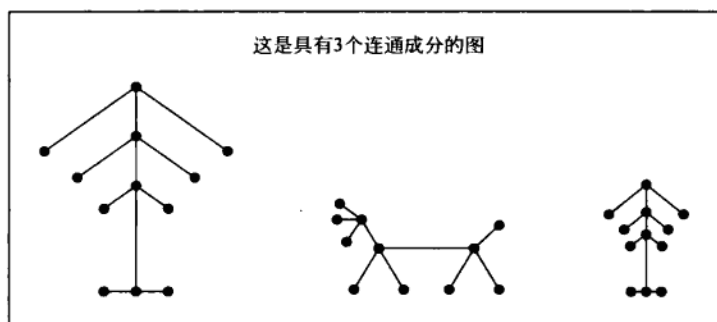


图 7-3 一个森林的例子

通常把树定义成具有在每对顶点之间存在唯一简单通路的性质的无向图。下面的定理说明这个定义与原来的定义等价。

定理 1 一个无向图是树当且仅当在它的每对顶点之间存在唯一简单通路。

证 首先假定 T 是树。则 T 是没有简单回路的连通图。设 x 和 y 是 T 的两个顶点。因为 T 是连通的, 根据 6.4 节定理 1, 在 x 和 y 之间存在一条简单通路。另外, 这条通路必然是唯一的, 因为假如存在第二条这样的通路, 那么从 x 到 y 的第一条通路以及将第二条通路逆转后所得到的从 y 到 x 的通路, 将组合起来形成回路。利用 6.4 节练习 24, 这蕴含着在 T 中存在简单回路。因此, 在树的任何两个顶点之间存在唯一简单通路。

现在假定在图 T 的任何两个顶点之间存在唯一简单通路。则 T 是连通的, 因为在它的任何两个顶点之间存在通路。另外, T 没有简单回路。为了看出这是个真命题, 假定 T 有包含顶点 x 和 y 的简单回路。则在 x 和 y 之间就有两条简单通路, 因为这条简单回路包含一条从 x 到 y 的简单通路和一条从 y 到 x 的简单通路。因此, 在任何两个顶点之间存在唯一简单通路的图是树。□

在树的许多应用里, 指定树的一个特殊顶点作为根。一旦规定了根, 就可以给每条边指定方向如下。因为从根到图的每个顶点存在唯一通路(根据定理 1), 所以指定每条边是离开根的方向。因此, 树与它的根一起产生一个有向图, 称为根树。

定义 2 根树是指定一个顶点作为根并且每条边的方向都离开根棵树。

也可以递归地定义根树。通过选择任何一个顶点来作为根, 就可以把非根树变成根树。注意

对根的不同选择会导致产生不同的根树。例如，图 7-4 显示通过在树 T 里分别指定 a 和 c 作为根所形成的根树。通常在画根树时把根画在图的顶端。指示根树中边的方向的箭头可以省略，因为对根的选择确定了边的方向。

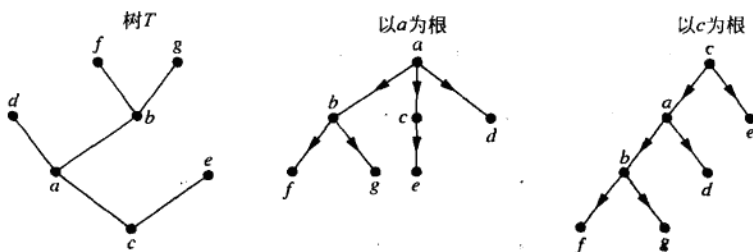


图 7-4 一棵树以及指定两个根所形成的根树

树这个术语起源于植物学和族谱学。假定 T 是根树。若 v 是 T 里非根的顶点，则 v 的父母是使得从 u 到 v 存在有向边的唯一的顶点 u (读者应当证明这样的顶点 u 是唯一的)。当 u 是 v 的父母时， v 称为 u 的子女。具有相同父母的顶点称为兄弟。非根顶点的祖先是根到该顶点的通路上的顶点，不包括该顶点自身但包括根 (即该顶点的父母，该顶点的父母的父母，等等，一直到根为止)。顶点 v 的后代是以 v 作为祖先的顶点。树的顶点若没有子女则称为树叶。有子女的顶点称为内点。根是内点，除非它是图中唯一的顶点，在这种情况下它是树叶。

若 a 是树里的顶点，则以 a 为根的子树是由 a 和 a 的后代以及这些顶点所关联的边所组成的该树的子图。

例 2 在图 7-5 所示的根树里 (有根 a)，求 c 的父母， g 的子女， h 的兄弟， e 的所有祖先， b 的所有后代，所有内点以及所有树叶。什么是根在 g 处的子树？

解 c 的父母是 b 。 g 的子女是 h ， i 和 j 。 h 的兄弟是 i 和 j 。 e 的祖先是 c ， b 和 a 。 b 的后代是 c ， d 和 e 。内点是 a ， b ， c ， g ， h 和 j 。树叶是 d ， e ， f ， i ， k ， l 和 m 。根在 g 处的子树如图 7-6 所示。

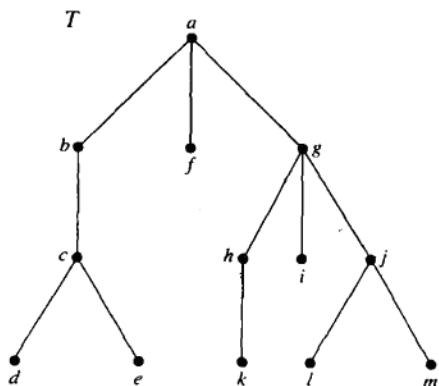


图 7-5 根树 T

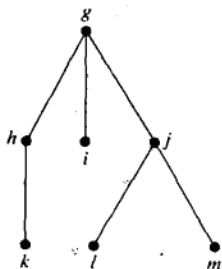


图 7-6 根在 g 处的子树

在许多不同的应用里都使用具有下面性质的根树：它们的所有内点都有同样个数的子女。在本章后面将用这样的树去研究涉及搜索、排序和编码的问题。

定义 3 若根树的每个内点都有不超过 m 个子女，则称它为 m 元树。若该树的每个内点都恰好有 m 个子女，则称它为正则 m 元树。把 $m=2$ 的正则 m 元树称为二叉树。

例 3 图 7-7 里的根树，对某个正整数 m 来说是否为正则 m 元树？

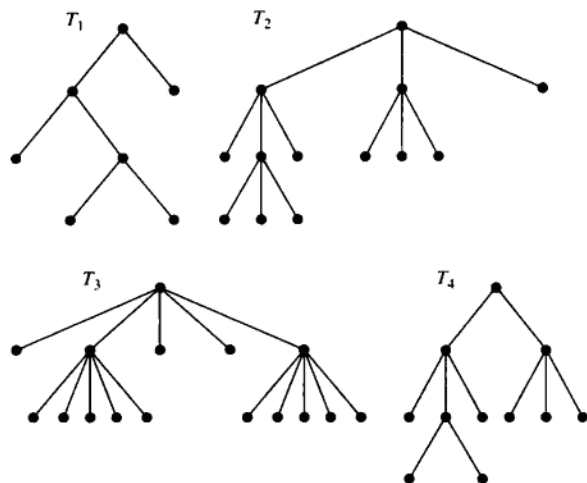


图 7-7 四个根树

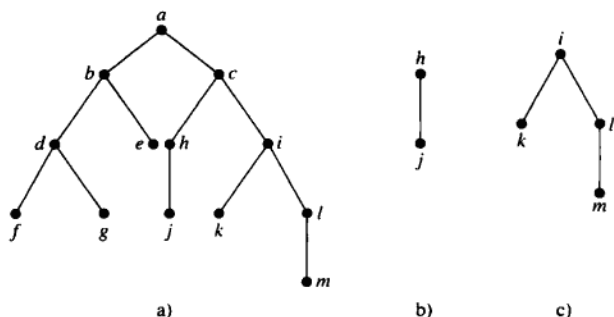
解 T_1 是正则二叉树, 因为它的每个内点都有两个子女。 T_2 是正则三元树, 因为它的每个内点都有三个子女。在 T_3 里每个内点都有五个子女, 所以它是正则五元树。对任何 m 来说, T_4 都不是正则 m 元树, 因为它的某些内点有两个子女而其他内点有三个子女。 ■

有序根树是把每个内点的子女们都排序的根树。画有序根树时, 以从左向右的顺序来显示每个内点的子女们。注意在常规方式下根树的表示将确定它的边的一种顺序。我们将在画图时使用边的这种顺序, 但不明确地指出认为根树是有序的。

在有序二叉树(通常只称为二叉树)中, 若一个内点有两个子女, 则第一个子女称为左子女而第二个子女称为右子女。根处在一个顶点的左子女处的树称为该顶点的左子树, 而根处在一个顶点的右子女处的树称为该顶点的右子树。读者应当注意, 对某些应用来说, 二叉树的每个非根顶点都指定为其父母的右子女或左子女。即使当某些顶点仅有一个子女时也这样做。具体指定方式视需要而定。

可以递归地定义有序根树。

例 4 在图 7-8a 所示二叉树 T 里, 什么是 d 的左子女和右子女(其中顺序是画法所蕴含的)? 什么是 c 的左子树和右子树?

图 7-8 二叉树 T 和顶点 c 的左子树和右子树

解 d 的左子女是 f 而右子女是 g 。在图 7-8b 和图 7-8c 中分别显示 c 的左子树和右子树。 ■
与图的情形恰好一样, 不存在用来描述树、根树、有序根树和二叉树等的标准术语。出现这

种非标准的术语是因为在计算机科学里大量地使用树，而计算机科学还是相对年轻的领域。每当碰到关于树的术语时，读者就应当仔细地核对这些术语所表示的意思。

7.1.1 树作为模型

以树为模型的应用领域非常广泛，比如计算机科学、化学、地理学、植物学和心理学等。下面将描述基于树的各式各样的模型。

例5 饱和碳氢化合物与树 图可以用来表示分子，其中用顶点表示原子，用边表示原子之间的化学键。英国数学家亚瑟·凯莱在1857年发现了树，当时他正在试图列举形如 C_nH_{2n+2} 的化合物的同分异构体，它们都称为饱和碳氢化合物。

在饱和碳氢化合物的图模型里，用4度顶点表示每个碳原子，用1度顶点表示每个氢原子。在形如 C_nH_{2n+2} 的化合物的表示图里有 $3n+2$ 个顶点。在这个图中，边数是顶点度数之和的一半。因此，在这个图中有 $(4n+2n+2)/2=3n+1$ 条边。因为这个图是连通的，而且边数比顶点数少1，所以它必然是树。

带有 n 个4度顶点和 $2n+2$ 个1度顶点的非同构的树，就表示 C_nH_{2n+2} 的不同的同分异构体。例如，当 $n=4$ 时，恰好存在两个这种类型的不同的同分异构体。它们的结构如图7-9所示。这两种同分异构体称为丁烷和异丁烷。

例6 表示组织机构 大的组织机构的结构可以用根树来建模。在这个树里每个顶点表示机构里的一个职务。从一个顶点到另外一个顶点的边的起点所表示的人是终点所表示的人的(直接)上司。图7-10所示就是这样的树。在这个树所表示的组织机构里，硬件开发主任直接为研发经理工作。

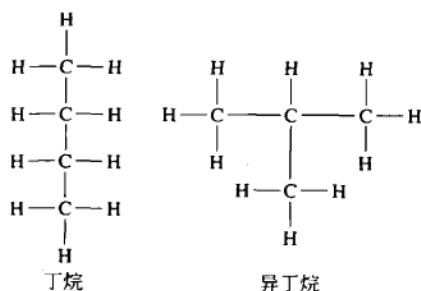


图7-9 丁烷的两种同分异构体

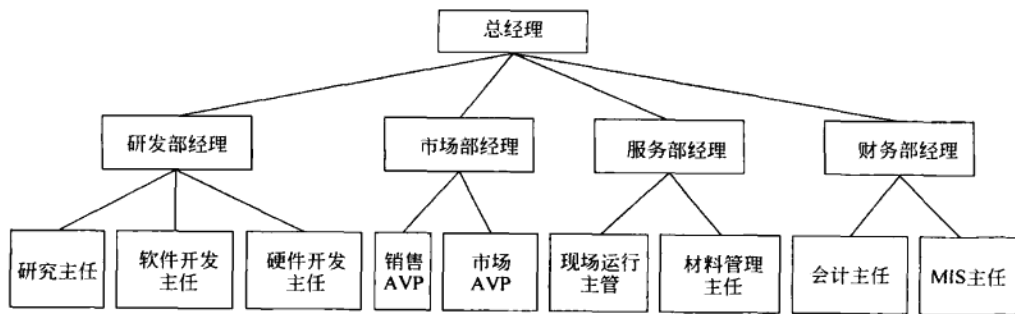


图7-10 一家计算机公司的组织机构图

例7 计算机文件系统 计算机存储器中的文件可以组织成目录。目录可以包含文件和子目录。根目录包含整个文件系统。因此，文件系统可以表示成根树，其中根表示根目录，内点表示子目录，树叶表示文件或空目录。在图7-11中显示一个这样的文件系统。在该系统中，文件 khr 属于目录 rje 。(注意文件的链接，同一个文件会有一个以上的路径名，导致计算机文件系统中存在回路。)

例8 树形连接并行处理系统 在6.2节例16里描述过多种并行处理的互连网络。树形连接网络是把处理器互相连接的另外一种重要方式。这样的网络的表示图是正则二叉树。这样的网络把 $n=2^k-1$ 个处理器互连起来，其中 k 是正整数。一个非根也非树叶的顶点 v 所表示的处理器具有三个双向连接，一个连接通向 v 的父母所表示的处理器，两个连接通向 v 的两个子女所表示的处理器。根所表示的处理器具有两个双向连接，分别通向 v 的两个子女所表示的处理器。树叶所表

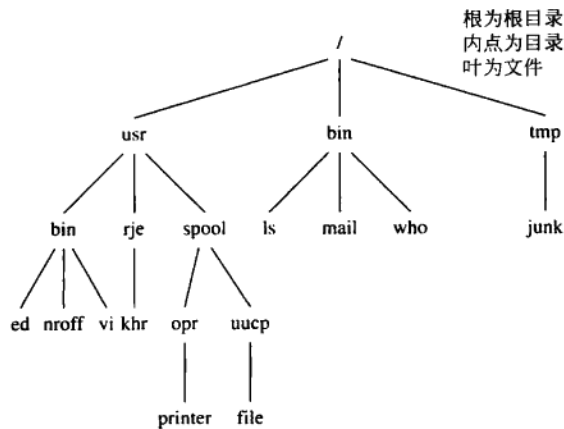


图 7-11 一个计算机文件系统

示的处理器具有一个双向连接，通向 v 的父母。在图 7-12 里显示带 7 个处理器的树形连接网络。

将要说明并行计算是如何使用树形连接网络的。具体地说，将说明图 7-12 里的处理器是如何用三步来完成 8 个数的相加的。在第一步用 P_4 相加 x_1 和 x_2 。用 P_5 相加 x_3 和 x_4 。用 P_6 相加 x_5 和 x_6 。用 P_7 相加 x_7 和 x_8 。在第二步用 P_2 相加 x_1+x_2 和 x_3+x_4 。用 P_3 相加 x_5+x_6 和 x_7+x_8 。最后，在第三步用 P_1 相加 $x_1+x_2+x_3+x_4$ 和 $x_5+x_6+x_7+x_8$ 。这种方法要优于串行地相加 8 个数所需要的七步，串行的步骤是依次把一个数与表中前面各数之和相加。

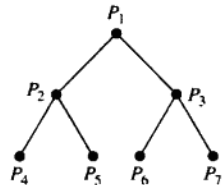


图 7-12 带 7 个处理器的树形连接网络

7.1.2 树的性质

我们常常需要知道树里各种各样的边和顶点的数目之间的联系。

定理 2 带有 n 个顶点的树含有 $n-1$ 条边。

证 将用数学归纳法来证明这个定理。注意对于所有的树来说，在这里可以选择一个树根并且认为这个树有根。

基础步骤：当 $n=1$ 时，有 $n=1$ 个顶点的树没有边。所有对于 $n=1$ 来说定理为真。

归纳步骤：归纳假设说有 k 个顶点的每个树都有 $k-1$ 条边，其中 k 是正整数。假设树 T 有 $k+1$ 个顶点并且 v 是 T 的树叶 (v 必定存在，因为树是有穷的)，设 w 是 v 的父母，从 T 删除顶点 v 和连接 w 和 v 的边，就产生有 k 个顶点的树 T' ，因为所得出的图还是连通的并且没有简单回路。根据归纳假设， T' 有 $k-1$ 条边。所以 T 有 k 条边，因为 T 比 T' 多 1 条边，即连接 v 和 w 的边。这样就完成了归纳步骤。 $\square$

下面的定理说明，带有指定的内点数的 m 元树的顶点数是确定的。与定理 2 一样，将用 n 来表示树中的顶点数。

定理 3 带有 i 个内点的正则 m 元树含有 $n=mi+1$ 个顶点。

证 除根之外的每个顶点都是内点的子女。因为每个内点有 m 个子女，所以在树中除根之外还有 mi 个顶点。因此，该树含有 $n=mi+1$ 个顶点。 $\square$

假定 T 是完全 m 元树。设 i 是该树的内点数而 l 是树叶数。一旦 n ， i 和 l 之中的一个已知，另外的两个量就随之确定了。下面的定理给出如何从已知的一个量求其他两个量的方法。

定理 4 一个正则 m 元树若带有

(i) n 个顶点，则带有 $i=(n-1)/m$ 个内点和 $l=[(m-1)n+1]/m$ 个树叶；

(ii) i 个内点, 则带有 $n=mi+1$ 个顶点和 $l=(m-1)i+1$ 个树叶;

(iii) l 个树叶, 则带有 $n=(ml-1)/(m-1)$ 个顶点和 $i=(l-1)/(m-1)$ 个内点。

证 设 n 表示顶点数, i 表示内点数, l 表示树叶数。利用定理 3 中的等式, 即 $n=mi+1$, 以及等式 $n=l+i$ (这个等式为真, 是因为每一个顶点要么是树叶、要么是内点), 就可以证明本定理的所有三个部分。将在这里证明(i)。(ii)和(iii)的证明留给读者作为练习。

在 $n=mi+1$ 里求解 i 得出 $i=(n-1)/m$ 。然后把 i 的这个表达式代入等式 $n=l+i$, 就证明 $l=n-i=n-(n-1)/m=[(m-1)n+1]/m$ 。 $\square$

例 9 说明如何使用定理 4。

例 9 假定某人寄出一封连环信。要求收到信的每个人再把它寄给另外 4 个人。有一些人这样做了, 但是其他人则没有寄出信。若没有人收到超过一封的信, 而且若读过信但是不寄出它的人数超过 100 个后, 连环信就终止了, 则包括第一个人在内, 有多少人看过信? 有多少人寄出过信?

解 可以用 4 元树表示连环信。内点对应于寄出信的人, 而树叶对应于不寄出信的人。因为有 100 个人不寄出信, 所以在这个根树里, 树叶数是 $l=100$ 。因此, 定理 4 的(iii)说明, 已经看过信的人数是 $n=(4 \cdot 100-1)/(4-1)=133$ 。另外, 内点数是 $133-100=33$, 所以 33 个人寄出过信。 $\blacksquare$

经常需要使用这样的根树, 它们是“平衡的”, 所以在每个顶点处的子树都包含大约相同长度的通路。下面的一些定义将解释清楚这个概念。在根树里顶点 v 的层是从根到这个顶点的唯一通路的长度。根的层定义为 0。根树的高度就是顶点层数的最大值。换句话说, 根树的层数是从根到任意顶点的最长通路的长度。

例 10 求图 7-13 所示的根树里每个顶点的层数。这棵树的高度是什么?

解 根 a 在 0 层上。顶点 b, j 和 k 都在 1 层上。顶点 c, e, f 和 l 都在 2 层上。顶点 d, g, i, m 和 n 都在 3 层上。最后, 顶点 h 在 4 层上。因为任意顶点的最大层数是 4, 所以这棵树的高度为 4。 $\blacksquare$

若一棵高度为 h 的 m 元树的所有顶点都在 h 层或 $h-1$ 层, 则这棵树是平衡的。

例 11 在图 7-14 所示的一些根树里, 哪些根树是平衡的?

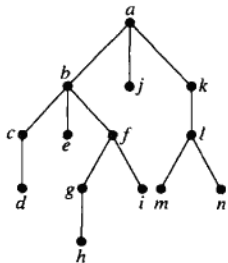


图 7-13 一棵根树

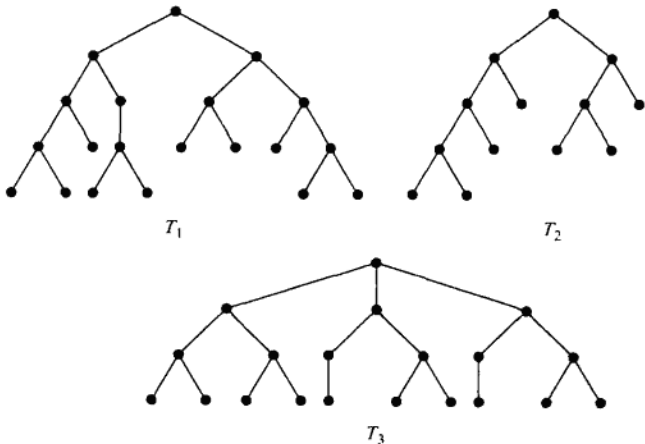


图 7-14 一些根树

解 T_1 是平衡的, 因为它所有的树叶都在 3 层和 4 层上。不过, T_2 不是平衡的, 因为它有树叶在 2 层、3 层和 4 层上。最后, T_3 是平衡的, 因为它所有的树叶都在 3 层上。 $\blacksquare$

定理 5 中的结果建立 m 元树的高度与顶点数之间的联系。

定理 5 在高度为 h 的 m 元树里至多有 m^h 个树叶。

证 本证明对高度使用数学归纳法。首先, 考虑高度为 1 的 m 元树。这些树都是由带有不超过 m 个子树的一个根所组成的, 每个子女都是树叶。因此在高度为 1 的 m 元树里有不超过

$m^1 = m$ 个树叶。这是归纳论证的基础步骤。

现在假定对高度小于 m 的所有 m 元树来说，这个结果都为真；这是归纳假设。设 T 是高度为 h 的 m 元树。 T 的树叶都是通过删除从根到每个在 1 层的顶点的边所获得的 T 的子树的树叶，如图 7-15 所示。

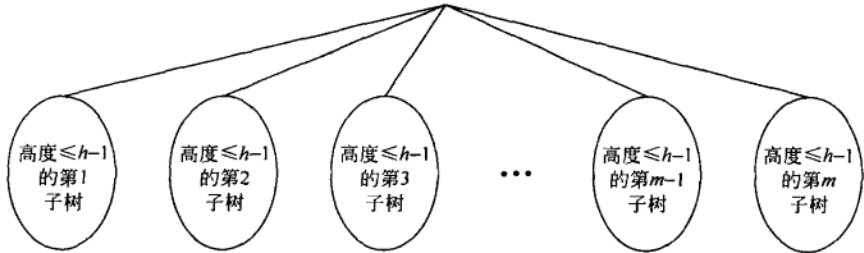


图 7-15 证明的归纳步骤

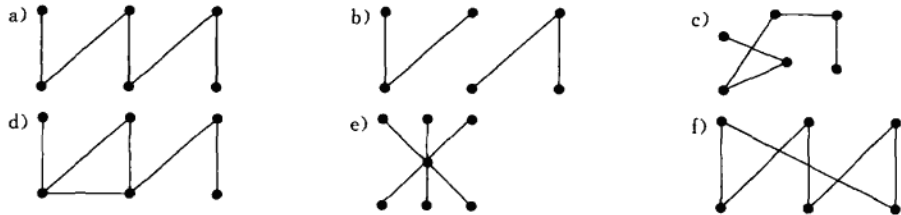
这些子树的高度都小于或等于 $h-1$ 。所以根据归纳假设，这些根树每个都有至多 m^{h-1} 条边。因为最多有 m 个这样的子树，每个子树最多有 m^{h-1} 个树叶，所以在这个根树里最多有 $m \cdot m^{h-1} = m^h$ 个树叶。这样就完成了归纳论证。 □

推论 1 若一个高度为 h 的 m 元树带有 l 个树叶，则 $h \geq \lceil \log_m l \rceil$ 。若这个 m 元树是正则的和平衡的，则 $h = \lceil \log_m l \rceil$ (在这里使用向上取整函数。回忆一下， $\lceil x \rceil$ 是大于或等于 x 的最小整数)。

证 从定理 5 知道 $l \leq m^h$ 。取以 m 为底的对数就证明 $h \geq \log_m l$ 。因为 h 是整数，所以有 $h \geq \lceil \log_m l \rceil$ 。现在假定这个树是平衡的。于是每个树叶都在 h 层或 $h-1$ 层上，而且因为树的高度为 h ，所以在 h 层至少有一个树叶。所以必然有超过 m^{h-1} 个树叶。因为 $l \leq m^h$ ，所以有 $m^{h-1} < l \leq m^h$ 。在这个不等式里取以 m 为底的对数就得出 $h-1 < \log_m l \leq h$ 。因此 $h = \lceil \log_m l \rceil$ 。 □

练习

1. 下面哪些图是树？



2. 回答下列关于左下图所示的根树的问题。

- a) 哪个顶点是根？
- b) 哪些顶点是内点？
- c) 哪些顶点是树叶？
- d) 哪些顶点是 j 的子女？
- e) 哪些顶点是 h 的父母？
- f) 哪些顶点是 o 的兄弟？
- g) 哪些顶点是 m 的祖先？
- h) 哪些顶点是 b 的后代？

3. 练习 2 里的根树对某个正整数 m 来说，是否是正则 m 元树？

4. 练习 2 里的根树的每个顶点的层数是什么？

5. 画出练习 2 里的树以下列顶点为根的子树。

- a) a b) c c) e

6. a) 有多少种非同构的带有 3 个顶点的非根树？

b) 有多少种非同构的带有 3 个顶点的根树(使用有向图的同构)？

*7. a) 有多少种非同构的带有 5 个顶点的非根树？

b) 有多少种非同构的带有 5 个顶点的根树(使用有向图的同构)？

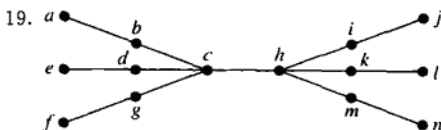
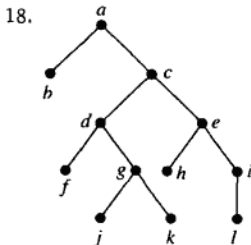
8. 带有 10 000 个顶点的树有多少条边？

9. 带有 1000 个内点的正则二叉树有多少条边?
10. 假定 1000 个人参加象棋巡回赛。若一个选手输掉一盘就遭到淘汰, 而且比赛进行到只有一位参加者还没有输过为止, 则利用这个巡回赛的根树模型, 来确定为了决出冠军必须下多少盘棋(假定没有平局)。
11. 一封连环信。开始时一个人寄出一封信给其他 10 个人。要求每个人寄出此信给其他 10 个人, 而且每封信都包含该连环信前面 6 个人的列表。除非表中不足 6 个名字, 否则每个人都寄一美元给表中第一个人, 从表中删除这个人的名字, 把其他 5 个人的名字向上移动一位, 并且把他或她自己的名字插入到表的末尾。若没有人中断这个连环, 并且每人至多收到一封信, 则这个连环里的一个人最终将收到多少钱?
- *12. 要么画出带有 84 个树叶而且高度为 3 的正则 m 元树, 其中 m 是正整数, 要么证明这样的树不存在。

一个完全 m 元树是其中每个树叶都在同一层上的正则 m 元树。

13. 构造高度为 4 的完全二叉树和高度为 3 的完全 3 元树。
14. 证明: a) 定理 4 的部分(ii)。b) 定理 4 的部分(iii)。
15. 在包含总共 n 个顶点的 t 个树的森林里有多少条边?
16. 下面的饱和碳氢化合物有多少种不同的同分异构体?
a) C_3H_8 b) C_5H_{12} c) C_6H_{14}
17. 设 n 是 2 的幂。证明: 可以用 $n-1$ 个处理器的树形连接网络在 $\log n$ 步里求 n 个数之和。

非根树里顶点的离心度是从这个顶点开始的最长的简单通路的长度。若在树里没有其他顶点比一个顶点的离心度更小, 则这个顶点就称为中心。在练习 18~19 里, 求每一个是所给的树的中心的顶点。



- *20. 证明: 树有一个中心或两个相邻的中心。

根斐波那契树 T_n 是以下面的方式来递归地定义的。 T_1 和 T_2 都是包含单个顶点的根树, 而对 $n=3, 4, \dots$ 来说, 都是从一个根以 T_{n-1} 作为左子树并且以 T_{n-2} 作为右子树来构造出根树 T_n 。

21. 画出前 7 个根斐波那契树。
22. 下面这个使用数学归纳法的“证明”错在什么地方? 命题: 有 n 个顶点的每个树都有长度为 $n-1$ 的路径。
基础步骤: 有 1 个顶点的每个树显然有长度为 0 的路径。归纳步骤: 假设有 n 个顶点的树有长度为 $n-1$ 的路径, 这个路径以 u 作为终点。加入顶点 v 和从 u 到 v 的边。所得出的树有 $n+1$ 个顶点并且有长度为 n 的路径。这样就完成了归纳步骤。

7.2 树的应用

7.2.1 引言

下面将要运用树来讨论三个问题。第一个问题是: 应当如何对列表里的项进行排序, 以便可以容易地找到项的位置? 第二个问题是: 为了在某种类型的一组对象里找出带某种性质的对象, 应当做出一系列什么样的决策? 第三个问题是: 应当如何用位串来有效地编码一组字符?

7.2.2 二叉搜索树

 在列表里搜索一些项, 是计算机科学的一项重要任务。主要目标是实现一个搜索算法, 当项都完全排序时, 这个算法有效地找出项。这个任务可以通过使用二叉搜索树来完成, 二叉搜索树是一种二叉树, 其中任何顶点的每个子节点都指定为右子女或左子女, 没有顶点具有超过一个的右子女或左子女, 而且每个顶点都用一个关键字来标记, 这个关键字是各项中的一个。另外, 这样指定顶点的关键字, 使得顶点的关键字不仅大于它的左子树里的所有顶点的关键

字，而且小于它的右子树里的所有顶点的关键字。

这个递归过程用来形成项的列表的二叉搜索树。从只包含一个顶点(即根)的树开始。指定列表中第一个项作为这个根的关键字。为了添加新的项，首先比较它与已经在树里的顶点的关键字，从根开始，若这个项小于所比较顶点的关键字而且这个顶点有左子女，则向左移动，若这个项大于所比较顶点的关键字而且这个顶点有右子女，则向右移动。当这个项小于所比较顶点的关键字而且这个顶点没有左子女时，就插入以这个项作为关键字的一个新顶点，并把新顶点作为这个顶点的左子女。同理，当这个项大于所比较顶点的关键字而且这个顶点没有右子女时，就插入以这个项作为关键字的一个新顶点，并把新顶点作为这个顶点的右子女。用例 1 来说明这个过程。

例 1 构造下面这些单词的二叉搜索树(用字母顺序): mathematics, physics, geography, zoology, meteorology, geology, psychology 和 chemistry。

解 图 7-16 显示了构造这个二叉搜索树所用的步骤。单词 mathematics 是根的关键字。因为 physics 是在 mathematics 之后(按照字母顺序)，所以给根添加带关键字 physics 的右子女。因为 geography 是在 mathematics 之前，所以给根添加带关键字 geography 的左子女。下一步，给带关键字 physics 的顶点添加右子女，并且给其指定关键字 zoology，因为 zoology 是在 mathematics 之后且在 physics 之后。同理，给带关键字 physics 的顶点添加左子女，并且给其指定关键字 meteorology。给带关键字 geography 的顶点添加右子女，并且给其指定关键字 geology。给带关键字 zoology 的顶点添加左子女，并且给其指定关键字 psychology。给带关键字 geography 的顶点添加左子女，并且给其指定关键字 chemistry(读者应当完成在每步上所需的所有比较)。

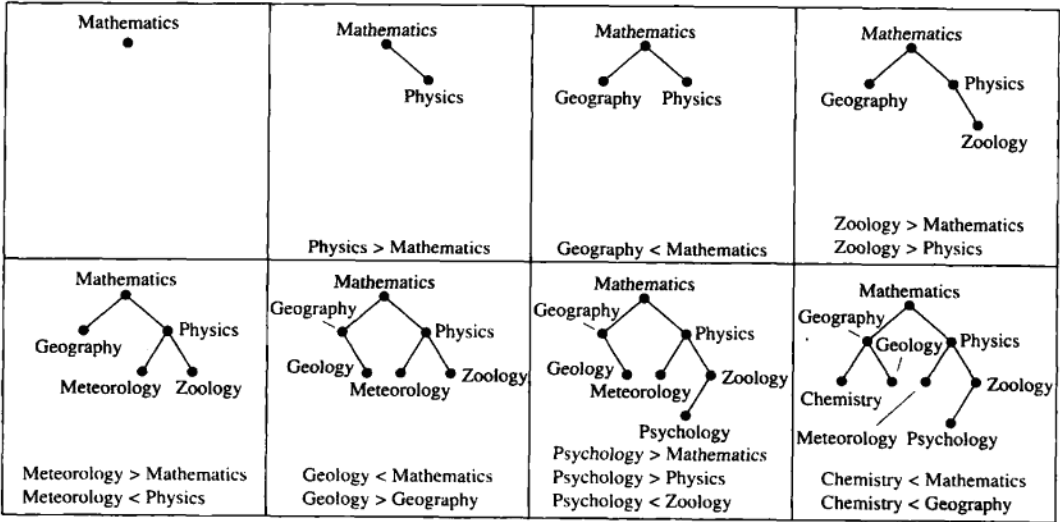


图 7-16 构造二叉搜索树

一旦建立了二叉搜索树，就需要一种在二叉搜索树中定位元素的方法，还需要一种添加新元素的方法。算法 1 是一个插入算法，尽管看上去它只是设计在二叉搜索树添加新节点，但实际上它可以完成上面提到的两个任务。也就是说，算法 1 可以在二叉搜索树定位该元素；如果一个元素 x 不存在，也可以添加该元素 x 。在下面的伪代码中， v 是当前正在查看的节点， $label(v)$ 是该节点的关键字。算法从根开始查看。如果 v 的关键字等于 x ，那么算法就找到了 x 的位置并结束；如果 x 比 v 的关键字小，就向 v 的左孩子节点移动并重复这个过程；如果 x 比 v 的关键字大，就向 v 的右孩子节点移动并重复这个过程。如果在任何一步，要移动到的孩子节点并不存在，那么就on 知道在这棵二叉搜索树中没有 x ，然后就添加一个以 x 为关键字的节点作为这个孩子节点。

算法 1 二叉搜索树算法

```

procedure insertion( $T$ : 二叉搜索树,  $x$ : 项)
 $v := T$  的根
{一个不在  $T$  里出现、具有值  $null$  的顶点}
while  $v \neq null$  并且  $label(v) \neq x$ 
begin
    if  $x < label(v)$  then
        if  $v$  的左子女  $\neq null$  then  $v := v$  的左子女
        else 添加  $new\ vertex$  作为  $v$  的左子女并且设置  $v := null$ 
    else
        if  $v$  的右子女  $\neq null$  then  $v := v$  的右子女
        else 给  $T$  添加  $new\ vertex$  作为  $v$  的右子女并且设置  $v := null$ 
end
if  $T$  的根  $= null$  then 给树添加顶点  $r$  并且用  $x$  标记它
else if  $label(v) \neq x$  then 用  $x$  标记  $new\ vertex$ 
{ $v = x$  的位置}

```

例 2 说明了算法 1 中二叉搜索树的用法。

例 2 运用算法 1 在例 1 的二叉搜索树中插入 *oceanography* 这个词。

解 算法 1 以 v 等于 T 的根节点开始, v 是当前查看的节点。因此 $label(v) = mathematics$ 。

因为 $v \neq null$, 且 $label(v) = mathematics < oceanography$, 接下来就查看根的右孩子节点。右孩子存在, 因此置当前查看的节点 v 等于这个右孩子。这一步, 有 $v \neq null$, 且 $label(v) = physics > oceanography$, 所以要查看 v 的左孩子。左孩子存在, 因此置当前查看的节点 v 等于这个左孩子。在这一步, 有 $v \neq null$, 且 $label(v) = meteorology < oceanography$, 所以试图查看 v 的右孩子。但是, 这个右孩子并不存在, 所以添加一个新的节点来当 v 的右孩子(此时就是关键字为 *oceanography* 的节点), 然后置 $v := null$ 。因为 $v = null$, 现在跳出了 **while** 循环。因为 T 的根不是 $null$ 而 $v = null$, 所以使用算法结束处的 **else if** 语句让新节点以 *oceanography* 为关键字。

现在我们来确定这个过程的计算复杂性。假定有 n 个项的列表的二叉搜索树 T 。可以从 T 这样构造一个满二叉树 U : 在必要时添加无标记的顶点, 以使得每个带关键字的顶点都有两个子女。这个做法在图 7-17 里说明。一旦这样做了, 就容易找出新项的位置, 或者添加新项作为关键字而不添加顶点。

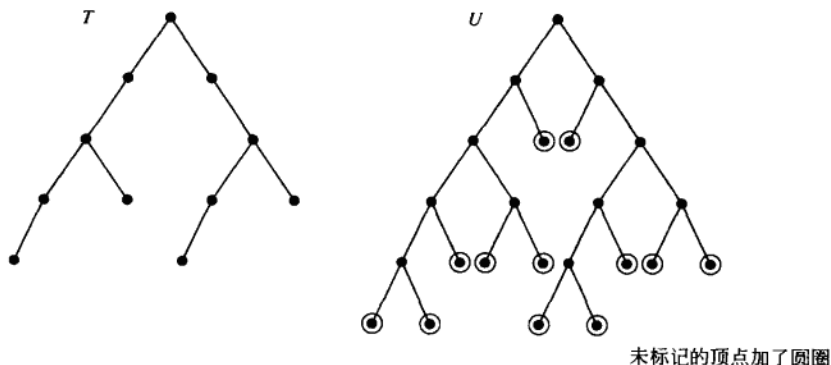


图 7-17 添加无标记顶点使二叉搜索树满

添加一个新项所需要的比较次数,最多等于在 U 里从根到树叶的最长通路的长度。 U 的内点都是 T 的顶点。所以 U 有 n 个内点。现在可以利用7.1节定理4的部分(ii)来得出 U 有 $n+1$ 个树叶。利用7.1节的推论1,可以看出 U 的高度大于或等于 $h = \lceil \log(n+1) \rceil$ 。所以,必须至少执行 $\lceil \log(n+1) \rceil$ 次比较以添加某个项。注意若 U 是平衡的,则它的高度是 $\lceil \log(n+1) \rceil$ (根据7.1节的推论1)。因此,若二叉搜索树是平衡的,则确定一个项的位置或者添加一个项所需要的比较次数不超过 $\lceil \log(n+1) \rceil$ 次。当给二叉搜索树添加一些项时,该树可能变得不平衡。因为平衡的二叉搜索树给出二叉搜索的最优的最坏情形复杂性,所以添加项时重新平衡二叉搜索树的算法已经设计出来。感兴趣的读者可以查阅关于数据结构的参考文献来了解这些算法。

7.2.3 决策树

根树可以用来为这样的问题建立模型,其中一系列决策将生成一个解。例如,二叉搜索树可以用来基于一系列比较来找出项的位置,其中每次比较都说明是否已经找到了项的位置,或者是否应当向右或向左进入子树。其中每个内点都对应着一次决策,这些顶点的子树都对应着该决策的每种可能后果,这样的根树称为决策树。问题的可能的解对应着这个根树的通向树叶的通路。下一个例子说明决策树的应用。

例3 假定有重量相同的7枚硬币和重量较轻的一枚伪币。为了用一架天平确定这8枚硬币中哪个是伪币,需要多少次称重?给出找出这个伪币的算法。

解 在天平上每次称重结果有三种可能性。分别是:两个托盘有相同的重量,第一个托盘较重,或第二个托盘较重。所以,称重序列的决策树是3元树。在决策树里至少有8个树叶,这是因为有8种可能的后果(因为每枚硬币都可能是较轻的伪币),而每种可能的后果必须至少用一个树叶来表示。确定伪币所需要的最大称重次数是决策树的高度。从7.1节的推论1得出决策树的高度至少是 $\lceil \log_3 8 \rceil = 2$ 。因此,至少需要两次称重。

用两次称重来确定伪币,是可行的。说明如何这样做的决策树如图7-18所示。

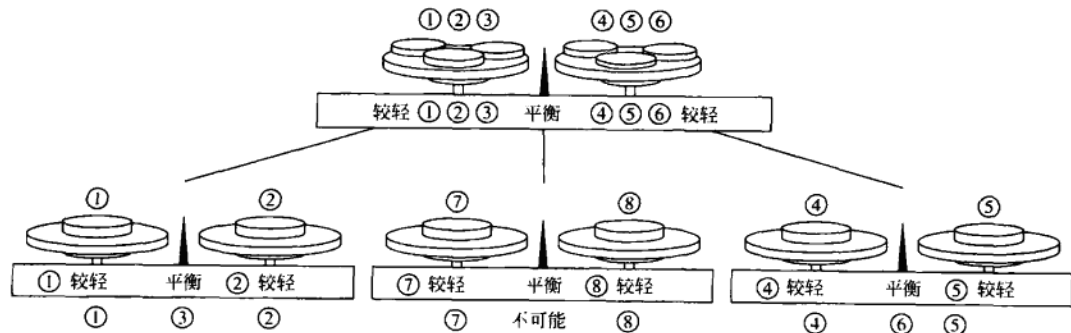


图 7-18 找出伪币位置的决策树。最下面一行为伪币判断结果

排序算法的复杂性 已经开发了许多不同的排序算法。为了确定一个具体的排序算法是否有效,就要确定这个算法的复杂性。用决策树作为模型,可以求出排序算法的最坏情形复杂性的下界。

可以用决策树为排序算法建立模型并且确定对这些算法的最坏情形复杂性的估计。注意给定 n 个元素,这些元素有 $n!$ 种可能的排序,因为这些元素 $n!$ 种排列中的每一个都可以是正确的顺序。本书研究的排序算法以及最常用的排序算法都是基于二叉比较,即一次比较两个元素。每次这样的比较都缩小了可能顺序的集合。因此,基于二叉比较的排序算法可以表示成二叉决策树,其中每个内点表示两个元素的一次比较。每个树叶表示 n 个元素的 $n!$ 种排列中的一种。

例4 在图7-19里显示排序列表 a, b, c 里元素的决策树。

基于二叉比较的排序的复杂性是用二叉比较的次数来度量的。排序有 n 个元素的列表所需要的最多比较次数就给出了这个算法的最坏情形复杂性。所用的最多比较次数等于表示这个排序过

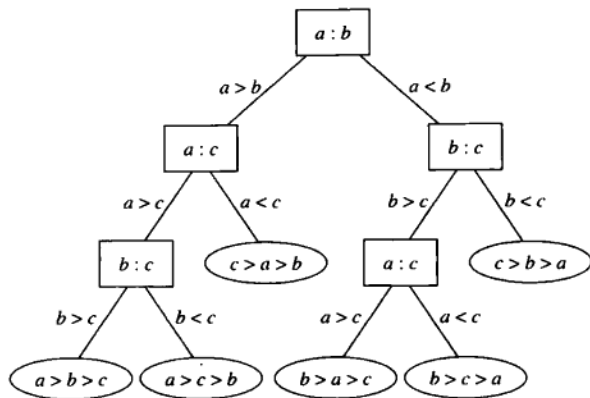


图 7-19 排序三个不同元素的决策树

程的决策树里的最长通路长度。换句话说，所需要的最多比较次数等于这个决策树的高度。因为带 $n!$ 个树叶的二叉树的高度至少是 $\lceil \log n! \rceil$ (利用 7.1 节推论 1)，所以定理 1 说至少需要 $\lceil \log n! \rceil$ 次比较。

定理 1 基于二叉比较的排序算法至少需要 $\lceil \log n! \rceil$ 次比较。

可以用定理 1 来给出基于二叉比较的排序算法所用比较次数的大 O 估计。我们知道 $\lceil \log n! \rceil$ 是 $\Theta(n \log n)$ ，这是算法的计算复杂性经常使用的一个参照函数。推论 1 是这个估计的结果。

推论 1 基于二叉比较的排序算法排序 n 个元素所用的比较次数是 $\Omega(n \log n)$ 。

推论 1 的一个后果是，基于二叉比较的排序算法在最坏情形下使用 $\Theta(n \log n)$ 次比较来排序 n 个元素，其他这类算法都没有更好的最坏情形复杂性，在这个意义下，基于二叉比较的排序算法是最优的。注意，在这个意义下归并排序算法是最优的。

对于排序算法的平均情形复杂性也可以证明类似的结果。基于二叉比较的排序算法所用的平均比较次数是表示这个排序算法的决策树中的平均树叶深度。我们知道，有 N 个顶点的二叉树的平均树叶深度是 $\Omega(\log N)$ 。当令 $N = n!$ 并且注意到因为 $\log n! = \Theta(n \log n)$ ，所以是 $\Omega(\log n!)$ 的函数也是 $\Omega(n \log n)$ 时，就会得出下面的估计。

定理 2 基于二叉比较的排序算法排序 n 个元素所用的平均比较次数是 $\Omega(n \log n)$ 。

7.2.4 前缀码

考虑一下这样的问题：用位串来编码英语字母表里的字母(其中不区分小写和大写字母)。可以用长度为 5 的位串来表示每个字母，因为只有 26 个字母而且有 32 个长度为 5 的位串。当每个字母都用 5 位来编码时，用来编码数据的总位数是 5 乘以文本中的字符数。有没有可能找出这些字母的编码方案，使得在编码数据时使用的位更少？若可能，那么就可以节省存储空间而且缩短传输时间。

考虑用不同长度的位串来编码字母。较短的位串用来编码出现较频繁的字母，较长的位串应当用来编码不经常出现的字母。当用变化的位数来给字母编码时，就必须用某种方法来确定每个字母的位在何处开始和结束。例如，若把 e 编码成 0，把 a 编码成 1，而把 t 编码成 01，则位串 0101 可能对应着 eat 、 tea 、 $eaea$ 或 tt 。

为了保证没有位串对应着多个字母的序列，可以令一个字母的位串永远不出现在另外一个字母的位串的开头部分。具有这个性质的编码称为前缀码。例如，把 e 编码成 0、把 a 编码成 10、而把 t 编码成 11 的编码就是前缀码。从编码着一个单词的字母的唯一位串可以恢复这个单词。例如，串 10110 是 ate 的编码。为了看明白这一点，注意开始的 1 不表示一个字符，但是 10 表示 a (并且它不可能是另外一个字母的位串的开始部分)。然后，下一个 1 不表示一个字符，但是 11 表示 t 。

最后一位 0 表示 e 。

前缀码可以用二叉树来表示, 其中字符是树里树叶的标记。标记树的边, 使得通向左子女的边标记为 0 而通向右子女的边标记为 1。用来编码一个字符的位串是在从根到以这个字符作为标记的树叶的唯一通路上边的标记的序列。例如, 图 7-20 里的树表示把 e 编码成 0, 把 a 编码成 10, 把 t 编码成 110, 把 n 编码成 1110 和把 s 编码成 1111。

表示编码的树可以用来解码位串。例如, 考虑一个用图 7-20 里的编码编成 11111011100 的单词。这个位串可以这样解码: 从根开始, 用位的序列来形成一条到树叶为止的通路。每个 0 位都使得通路向下到达通向通路里上一个顶点的左子女的边, 而每个 1 位都对应到上一个顶点的右子女。所以, 开头的 1111 对应这样的通路: 从根开始, 向右前进四次, 到达以 s 作为标记的树叶, 因为串 1111 是 s 的编码。从第五个位继续进行, 在向右再向左之后, 就到达下一个树叶, 这时访问以 a 作为标记的顶点, 其中把 a 编码成 10。从第七个位开始, 在向右三次然后向左之后, 到达了下一个树叶, 这时访问用 n 标记的顶点, 其中把 n 编码成 1110。最后, 末位 0 通向用 e 标记的树叶。因此, 原来的单词是 *sane*。

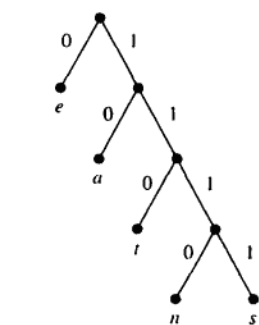


图 7-20 表示前缀码的二叉树

可以从任何二叉树来构造一个前缀码, 其中每个内点的左边都用 0 标记, 而右边都用 1 标记, 树叶都用字符标记。字符都用从根到这个树叶的唯一通路里的边的标记所组成的位串来编码。

赫夫曼编码 现在介绍一种算法, 这种算法用一个字符串中符号的出现频率(即出现概率)作为输入, 并且产生编码这个字符串的一个前缀码作为输出, 在这些符号的所有可能的二叉前缀码中, 这个编码使用最少的位。这个所谓赫夫曼编码的算法是大卫·赫夫曼 1951 年做麻省理工学院的研究生时发表在一篇学期论文中的。(注意, 这个算法假定已知字符串中每个符号出现多少次, 所以可以计算每个符号的出现频率, 方法是用这个符号出现的次数除以这个字符串的长度。)赫夫曼编码是数据压缩中的基本算法, 数据压缩的目的在于减少表示信息所需要的位数。赫夫曼编码广泛用于压缩表示文本的位串, 并且在压缩视频和图像文件方面也起到重要作用。

算法 2 表示赫夫曼编码算法。 给定符号及其频率, 目标是构造一个根二叉树, 其中符号是树叶的标记。算法从一些树的一个森林开始, 每个树含有一个顶点, 每个顶点有一个符号作为标记, 并且这个顶点的权就等于所标记符号的频率。在每一步, 都把具有最小总和权的两个树组合成一个单独的树, 方法是引入一个新的根, 把具有较大的权的树作为左子树, 把具有较小的权的树作为右子树。另外, 把这个树的两个子树的权之和作为这个树的总的权。(虽然可以规定在具有相同的权的树之间进行选择以打破平局的过程, 但是这里将不具体指定这样的过程。)当构造出了一个树, 即森林缩小为单个树时, 算法就停止。

算法 2 赫夫曼编码

procedure *Huffman*(C : 具有频率 w_i 的符号 a_i , $i=1, 2, \dots, n$)

$F := n$ 个根树的森林, 每个根树由单个顶点 a_i 组成并且赋权 w_i

while F 不是树

begin

把 F 中满足 $w(T) \geq w(T')$ 的权最小的根树 T 和 T' 换成具有新树根的一个树,

这个树根以 T 作为左子树并且以 T' 作为右子树。

用 0 标记树根到 T 的新边, 并且用 1 标记树根到 T' 的新边。

把 $w(T) + w(T')$ 作为新树的权。

end

{符号 a_i 的赫夫曼编码是从树根到 a_i 的唯一路径上的边的标记的连接}

例5 说明如何用算法2来编码一组5个符号。

例5 用赫夫曼编码来编码下列符号，这些符号具有下列频率：A: 0.08, B: 0.10, C: 0.12, D: 0.15, E: 0.20, F: 0.35。编码一个字符串所需要的平均位数是多少？

解 图7-21表示了编码这些符号所用的步骤。所产生的编码为：A是111, B是110, C是011, D是010, E是10, F是00。使用这种编码来编码一个符号所用的平均位数是

$$3 \cdot 0.08 + 3 \cdot 0.10 + 3 \cdot 0.12 + 3 \cdot 0.15 + 2 \cdot 0.20 + 2 \cdot 0.35 = 2.45$$

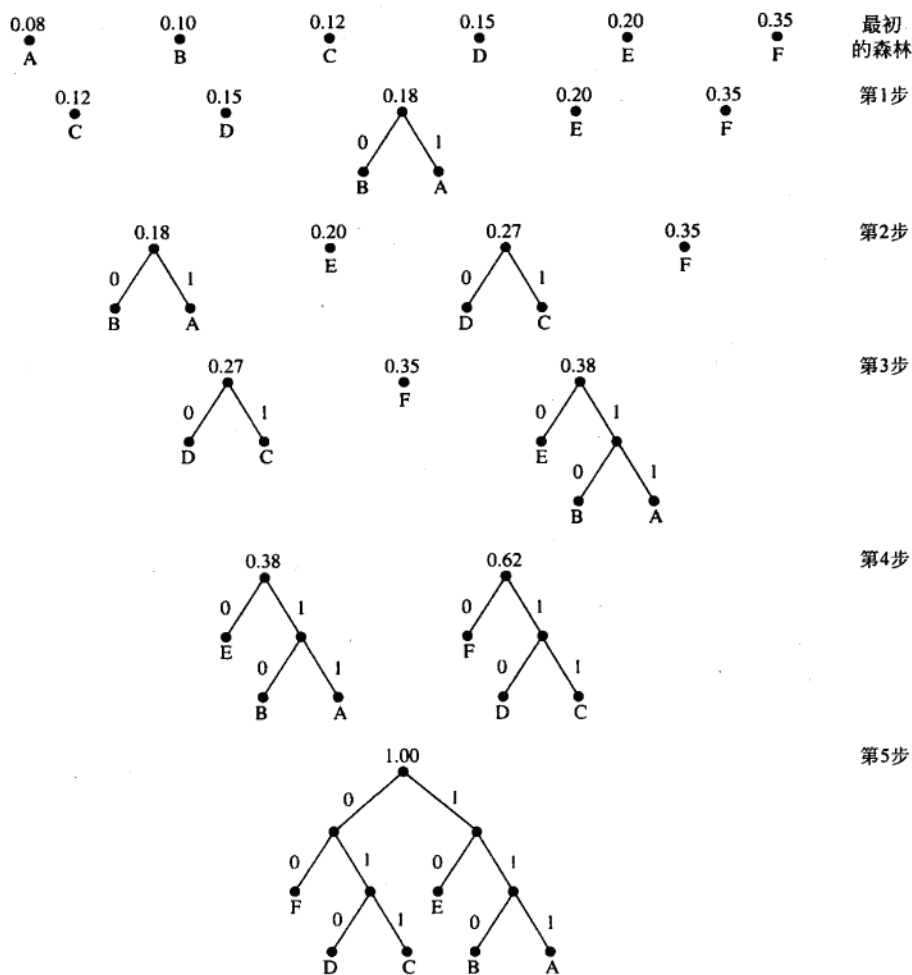


图7-21 例5中符号的赫夫曼编码

注意赫夫曼编码是贪心算法。在每一步替换具有最小权的两棵树，在没有任何二叉前缀码能使用更少的位来编码这些符号的情况下，这样做就导出了最优编码。

赫夫曼编码有许多变种。例如，不编码单个符号，可以编码指定长度的符号块，比如两个符号的块。这样做有可能减少编码这个字符串所需要的位数。也可以用两个以上的符号来编码这个字符串中的原始符号(参看本节末练习15前面的说明)。另外，当事先不知道一个字符串中每个符号的频率时，可以使用一种变种，即所谓的自适应赫夫曼编码(参见[Sa00])，使得在读这个字符串的同时来进行编码。

7.2.5 博弈树

可以用树来分析某些类型的游戏，比如圈叉游戏、轮流取石头游戏、跳棋和象棋。在每一种游戏当中，两个选手轮流进行移动。每个选手知道另一个选手的移动并且游戏里不存在偶然因素。使用博弈树为这样的游戏建立模型，这些树的顶点表示当游戏进行时游戏所处的局面，边表示在这些局面之间合乎规则的移动。由于博弈树常常很大，所以通过用同一个顶点表示所有对称的局面来简化博弈树。但是，如果一个游戏的不同移动序列导致同一个局面，则可以用不同的顶点来表示这个局面。通常的约定是用方框表示偶数层的顶点并且用圆圈表示奇数层的顶点。当游戏处在偶数层顶点所表示的局面时，就轮到第一个选手移动。当游戏处在奇数层顶点所表示的局面时，就轮到第二个选手移动。博弈树所表示的游戏可以永远不结束，比如进入了无穷循环，因而博弈树可以是无穷的，但是对于大多数游戏来说，都存在一些规则导致有穷的博弈树。

博弈树的树叶表示游戏的终局。给每个树叶指定一个值，来表示当游戏在这个树叶所代表的局面里终止时第一个选手的得分。对于非胜即负的游戏，用1来标记圆圈所表示的终结顶点以表示第一个选手获胜，用-1来标记方框所表示的终结顶点以表示第二个选手获胜。对于允许平局的游戏，用0来标记平局所对应的终结顶点。注意，对于非胜即负的游戏，为终结顶点指定值，这个值越高，第一个选手的结局就越好。

例6展示一个非常著名的和经过深入研究的游戏的博弈树。

例6 轮流取石头游戏 轮流取石头游戏是这样的，在游戏的开始，有几堆石头，两个选手轮流进行移动，一个合法的移动包括从其中一堆取走一块或多块石头，而不去移动其余的所有石头。不能进行合法移动的选手告负。（也可以规定取走最后一块石头的选手告负，因为不允许没有石头堆的局面。）图7-22所示的博弈树表示这种形式的给定开局的轮流取石头游戏，其中有3堆石头，分别包含2块、2块和1块石头。用不同堆中石头数的无序表来表示每个局面（堆的顺序无关紧要）。第一个选手的初始移动可以导致3种可能的局面，因为这个选手可以从有2块石头的堆中取走1块石头（留下包含1块、1块和2块石头的3堆），可以从包含2块石头的堆中取走2块石头（留下包含2块和1块石头的2堆），或者从包含1块石头的堆中取走1块石头（留下包含2块石头的2堆）。当只剩下包含1块石头的1堆时，就不可能进行合法移动了，所以这样的局面就是终局。由于轮流取石头游戏是非胜即负的游戏，所以用+1标记表示第一个选手获胜的终结顶点，用-1标记表示第二个选手获胜的终结顶点。

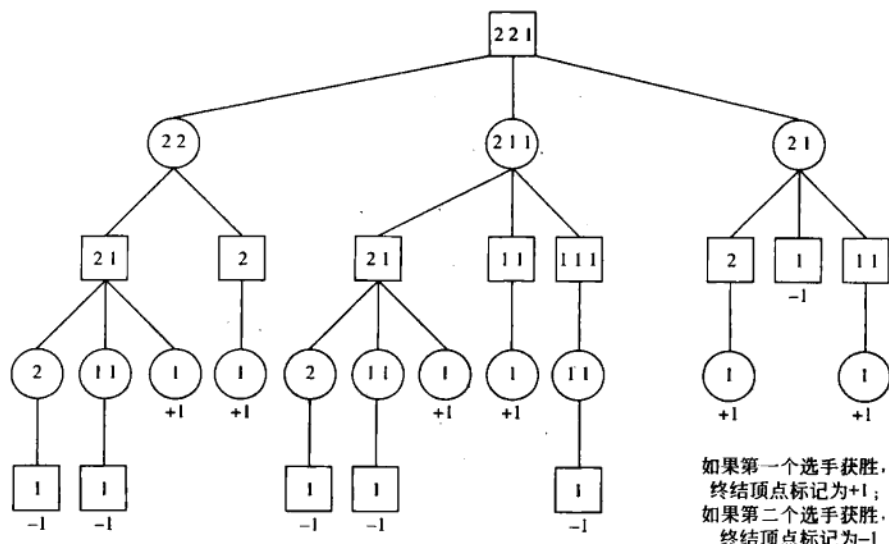


图 7-22 轮流取石头游戏的博弈树

例7 圈叉游戏 圈叉游戏的博弈树非常大，在这里不能画出，尽管计算机能轻而易举地构造出这样的树。图7-23a显示圈叉游戏的博弈树的一部分。因为对称的顶点是等价的，所以只需要考虑图7-23a所示的3种可能的初始移动。还显示这个博弈树的导致图7-23b中终局的一个子树，其中一个能够获胜的选手进行了制胜的移动。

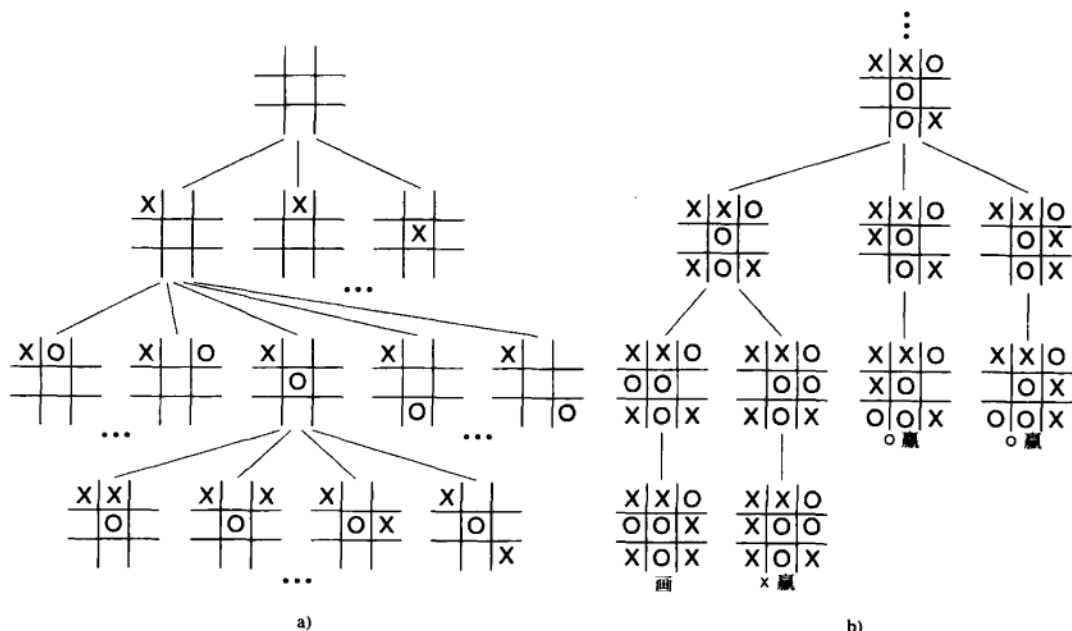


图 7-23 圈叉游戏的部分博弈树

可以用某种方式递归地定义博弈树中所有顶点的值，使得可以确定当两个选手都遵循最优策略时这个游戏的结果。所谓策略，就意味着一组规则，这些规则说明一个选手如何移动来赢得游戏。第一个选手的最优策略就是把这个选手的得分最大化的策略，第二个选手的最优策略就是把这个得分最小化的策略。现在递归地定义顶点的值。

定义1 博弈树中顶点的值递归地定义为：

(i) 一个树叶的值是当游戏在这个树叶所表示的局面里终止时第一个选手的得分。

(ii) 偶数层内点的值是这个内点的子女的最大值，奇数层内点的值是这个内点的子女的最小值。

第一个选手移动到具有最大值的子女所表示的局面并且第二个选手移动到具有最小值的子女所表示的局面，这样的策略称为最小最大策略。当两个选手都遵循最小最大策略时，通过计算树根的值就可以确定谁将赢得游戏，这个值称为树的值。这是定理3的后果。

定理3 如果两个选手都遵循最小最大策略并且从博弈树的某一个顶点所表示的局面开始进行游戏，则这个顶点的值说明第一个选手的得分。

证 将用归纳法来证明这个定理。

基础步骤：如果这个顶点是树叶，则根据定义指定给这个顶点的值就是第一个选手的得分。

归纳步骤：归纳假设一个顶点的子女的值就是第一个选手的得分，假定从这些顶点所表示的每一个局面中开始进行游戏。需要考虑两种情形，即当轮到第一个选手时和当轮到第二个选手时。

当轮到第一个选手时，这个选手遵循最小最大策略并且移动到具有最大值的子女所表示的局面。根据归纳假设，当从这个子女所表示的局面开始游戏并且遵循最小最大策略时，这个值就是

第一个选手的得分。根据偶数层内点的值的定义的递归步骤(作为其子女的最大值),当从这个顶点所表示的局面开始游戏时,这个顶点的值就是这个得分。

当轮到第二个选手时,这个选手遵循最小最大策略并且移动到具有最小值的子女所表示的局面。根据归纳假设,当从这个子女所表示的局面开始游戏并且遵循最小最大策略时,这个值就是第一个选手的得分。根据把奇数层内点的值作为其子女的最小值的递归定义,当从这个顶点所表示的局面开始游戏时,这个顶点的值就是这个得分。□

注意 通过扩展定理3的证明,可以证明对于两个选手来说最小最大策略都是最优策略。

例8 解释最小最大过程如何工作。它说明为例6的博弈树中的内点所指定的值。注意可以缩短所需要的计算,注意到对于非胜即负游戏来说,一旦找到方框顶点具有+1值的一个子女,则方框顶点的值也是+1,因为+1是最大可能的得分。同样,一旦找到圆圈顶点具有-1值的一个子女,则这个值也是这个圆圈顶点的值。

例8 例6构造了具有包含2块、2块和1块的3堆石头的开局的轮流取石头游戏的博弈树。图7-24说明这个博弈树的顶点的值。这些顶点的值是这样计算的:使用树叶的值并且每次向上计算1层。这个图的右边空白处说明究竟使用子女的最大值还是最小值来求出每层内点的值。例如,一旦求出了树根的3个子女的值,则这样求出树根的值:计算 $\max(1, -1, -1) = 1$ 。由于根的值是1,所以得出当两个选手都遵循最小最大策略时第一个选手获胜。■

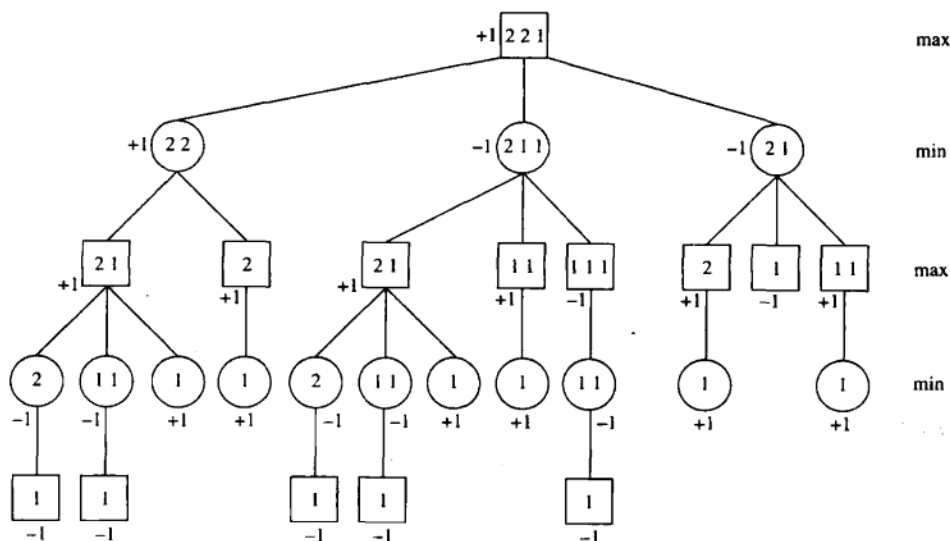


图 7-24 说明轮流取石头游戏中顶点的值

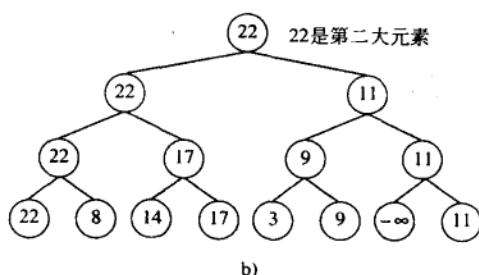
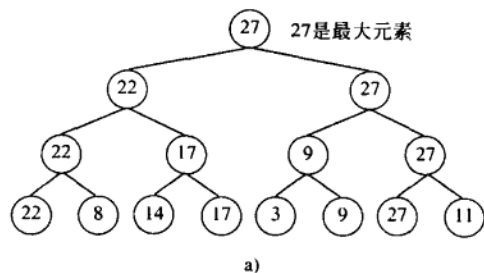
一些著名游戏的博弈树可能非同寻常地大,因为这些游戏有多种移动选择。例如,据估计象棋的博弈树有多达 10^{100} 个顶点!由于博弈树规模的原因,也许不可能直接使用定理3来研究这样的游戏,所以设计了各式各样的方法来帮助确定好的策略以及确定游戏的结果。一种被称作 α - β 剪枝的有用技巧减少了许多计算,它剪掉不能影响祖先顶点的值的那部分博弈树(关于 α - β 剪枝的信息,参考[Gr90])。另一种有用的方法是使用求值函数,当精确地计算博弈树中内点值不可行时,它就估计这些值。例如,在圈叉游戏中,可以使用不含圈O(O用来表示第二个选手的移动)的直行(行、列、对角线)数减去不含叉X(X用来表示第一个选手的移动)的直行数来作为一个局面的求值函数。这个求值函数给出了关于哪个选手在游戏中领先的一些说明。一旦插入求值函数的值,遵循最小最大策略使用的规则就可以计算出游戏的值。科学家研制出来的用

来下棋的计算机程序，比如著名的深蓝程序，都是基于复杂的求值函数的。关于计算机如何下棋的更多信息请参见[Le91]。

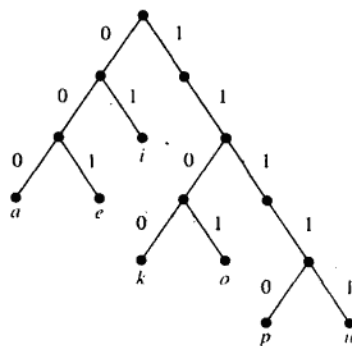
练习

- 用字母顺序建立下面这些单词的二叉搜索树：*banana*, *peach*, *apple*, *pear*, *coconut*, *mango* 和 *papaya*。
- 为了在练习 1 的搜索树里找出下面每个单词的位置或者添加它们，而且每次都重新开始，分别需要多少次比较？
a) *pear* b) *banana* c) *kumquat* d) *orange*
- 用字母顺序构造下面句子里的单词的二叉搜索树：“*The quick brown fox jumps over the lazy dog*”。
- 若一枚伪币与其他硬币质量不等，那么为了在 4 枚硬币中找出这枚伪币，需要用天平称多少次？描述用同样的称重次数来找出这枚伪币的算法。
- *5. 若一枚伪币比其他硬币较轻，那么为了在 12 枚硬币中找出这枚伪币，需要用天平称多少次？描述用同样的称重次数来找出这枚伪币的算法。
- *6. 求排序 4 个元素所需要的最少比较次数并且设计一个能够依此次数实现的算法。

竞赛图排序是通过构造有序二叉树来工作的排序算法。用将成为树叶的顶点来表示待排序的元素。就如同构造表示循环赛比赛胜者的树那样，一次构造这个树的一层。从左向右，比较成对的相邻元素，加入用所比较的两个元素中较大的那个来标记的一个父母顶点。在每一层在顶点的标记之间进行类似的比较，直到到达了用最大元素标记的树根为止。22, 8, 14, 17, 3, 9, 27, 11 的竞赛图排序所构造的树如下图(a)所示。一旦确定了最大元素，具有这个标记的树叶就重新标记为 $-\infty$ ，定义为比每个元素都小。从这个顶点直到树根的路径上所有顶点的标记都重新计算，如下图(b)所示。这样就产生了第二大元素。这个过程继续进行下去，直到整个表都已经排序为止。



- 完成表 22, 8, 14, 17, 3, 9, 27, 11 的竞赛图排序。说明在每个步骤上顶点的标记。
- 用伪码描述竞赛图排序。
- 竞赛图排序求第二大元素、第三大元素……，直到第 $(n-1)$ 大(或第二小)元素所使用的比较次数是多少？
- 下面哪些编码是前缀码？
a) $a: 11, e: 00, t: 10, s: 01$
b) $a: 0, e: 1, t: 01, s: 001$
c) $a: 101, e: 11, t: 001, s: 011, n: 010$
d) $a: 010, e: 11, t: 011, s: 1011, n: 1001, i: 10101$
- 若编码方案是用右面的树来表示，那么什么是 a 、 e 、 i 、 k 、 o 、 p 和 u 的编码？
- 用赫夫曼编码来编码具有给定频率的如下符号： $a: 0.20, b: 0.10, c: 0.15, d: 0.25, e: 0.30$ 。编码一个符号所需要的平均位数是多少？
- 为如下符号和频率构造两个不同的赫夫曼编码： $t: 0.2, u: 0.3, v: 0.2, w: 0.3$ 。
- 为英文字母表的字母构造赫夫曼编码，其中典型英文文本中字母的频率如下表所示。



字母	频率	字母	频率
A	0.0817	N	0.0662
B	0.0145	O	0.0781
C	0.0248	P	0.0156
D	0.0431	Q	0.0009
E	0.1232	R	0.0572
F	0.0209	S	0.0628
G	0.0182	T	0.0905
H	0.0668	U	0.0304
I	0.0689	V	0.0102
J	0.0010	W	0.0264
K	0.0080	X	0.0015
L	0.0397	Y	0.0211
M	0.0277	Z	0.0005

假设 m 是正整数且 $m \geq 2$ 。对于 N 个符号的集合来说，类似于二元赫夫曼编码的构造，可以构造 m 元赫夫曼编码。在初始步骤，把由 $((N-1) \bmod (m-1)) + 1$ 个权最小的单个顶点所组成的树组合成以这些顶点作为树叶的一棵树。在每个后续步骤，把权最小的 m 棵树组合成一棵 m 元树。

15. 使用符号 0、1 和 2，用三元 ($m=3$) 赫夫曼编码来编码具有给定频率的这些字母：A: 0.25, E: 0.30, N: 0.10, R: 0.05, T: 0.12, Z: 0.18。
16. 给定 $n+1$ 个符号 $x_1, x_2, \dots, x_n, x_{n+1}$ ，它们在一个字符串中分别出现 $1, f_1, f_2, \dots, f_n$ 次，其中 f_j 是第 j 个斐波那契数。当在赫夫曼编码算法的每个阶段考虑所有可能的打破平局的选择时，用来编码一个符号的最大位数是多少？
17. 画出轮流取石头游戏的博弈树，假设开局包括分别有 2 块和 3 块石头的两堆石头。在画这棵树的时候，用同一个顶点表示相同移动所导致的对称局面。求出这个博弈树每个顶点的值。如果两个选手都遵循最优策略，则哪个选手获胜？
18. 假设在轮流取石头游戏中修改获胜选手的得分，使得当 n 是到达终局前所做合法移动的步数时得分就是 n 美元。求第一个选手的得分，假设开局包括：
- a) 分别有 1 块和 3 块石头的两堆石头。
 - b) 分别有 2 块和 4 块石头的两堆石头。
 - c) 分别有 1 块、2 块和 3 块石头的三堆石头。
19. 画出圈叉游戏博弈树从下列每个局面开始的子树。确定每个子树的值。

a)

O	X	X
X	O	O
		X

b)

X	O	X
O	X	X
		O

c)

X		O
O	O	
X		X

d)

	O	X
	X	O
X	O	

20. 证明：如果轮流取石头游戏从包含相同数目的 2 堆石头开始，只要这个数目至少是 2，则当两个选手都遵循最优策略时第二个选手获胜。
21. 跳棋博弈树的根有多少个子女？有多少个孙子孙女？
22. 画出圈叉游戏博弈树前两步移动所对应的层。指明正文中所提到的求值函数的值，这个函数给局面指定不含 O 的直行数减去不含 X 的直行数来作为这一层每个顶点的值，并且在求值函数给出这些顶点的正确值的假设下，对这些顶点计算树的值。

7.3 树的遍历

7.3.1 引言

有序根树常常用来保存信息。掌握一些访问有序根树的每个顶点以存取数据的算法是非常必要的。几个重要的算法将在下面描述。有序根树也可以用来表示各种类型的表达式，比如由数字、变量和运算所组成的算术表达式。对用来表示这些表达式的有序根树来说，

它的顶点的一些不同的列表在这些表达式的求值中很有用。

7.3.2 通用地址系统

遍历有序根树的所有顶点的过程,都依赖于子女的顺序。在有序根树里,一个内点的子女从左向右地显示在表示这些有向图的图形里。

下面将描述一种完全地排序有序根树顶点的方法。为了产生这个顺序,必须首先标记所有的顶点。如下递归地完成这件事。

1. 用整数0标记根。然后用1, 2, 3, ..., k 从左向右标记它的 k 个子女(在1层上)。
2. 对在 n 层上带标记 A 的每个顶点 v ,按照从左向右画出它的 k_v 个子女的顺序,用 $A.1$, $A.2$, ..., $A.k$ 标记它的 k_v 个子女。

遵循这个过程,对 $n \geq 1$ 来说,在 n 层上的顶点 v 标记成 $x_1.x_2 \cdots x_n$,其中从根到 v 的唯一通路经过1层的第 x_1 个顶点,以及2层的第 x_2 个顶点,依此类推。这样的标记称为根树的通用地址系统。

可以利用顶点在通用地址系统里的标记的字典顺序将这些顶点完全排序。若存在 $i(0 \leq i \leq n)$ 满足 $x_1 = y_1, x_2 = y_2, \dots, x_{i-1} = y_{i-1}$,并且 $x_i < y_i$;或者若 $n < m$ 并且对 $i = 1, 2, \dots, n$ 来说 $x_i = y_i$,那么标记着 $x_1.x_2 \cdots x_n$ 的顶点就小于标记着 $y_1.y_2 \cdots y_m$ 的顶点。

例1 在如图7-25所示的有序根树的顶点的旁边,显示出了通用地址系统的标记。这些标记的字典顺序是

$0 < 1 < 1.1 < 1.2 < 1.3 < 2 < 3 < 3.1 < 3.1.1 < 3.1.2 < 3.1.2.1 < 3.1.2.2$
 $< 3.1.2.3 < 3.1.2.4 < 3.1.3 < 3.2 < 4 < 4.1 < 5 < 5.1 < 5.1.1 < 5.2 < 5.3$

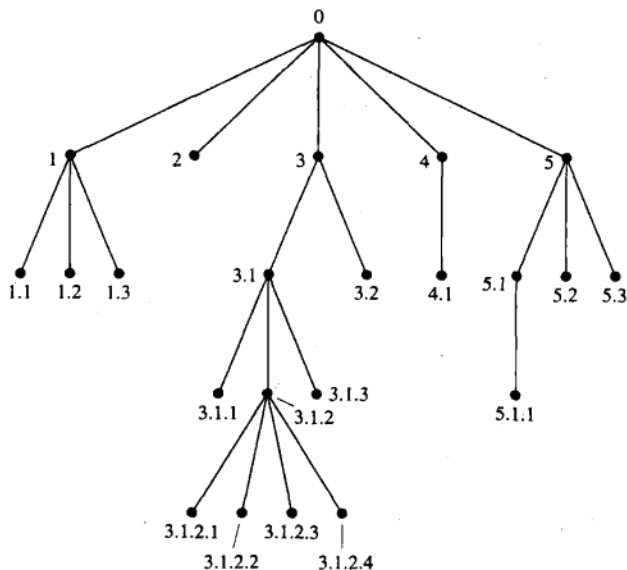


图 7-25 有序根树的通用地址系统

7.3.3 遍历算法

系统地访问有序根树每个顶点的过程都称为遍历算法。下面描述三个最常用的此类算法:前序遍历、中序遍历和后序遍历。这些算法都可以递归地定义。首先定义前序遍历。

定义1 设 T 是带根 r 的有序根树。若 T 只包含 r ,则 r 是 T 的前序遍历。否则,假定 $T_1, T_2, \dots, T_k$ 是 T 里在 r 处从左向右的子树。前序遍历首先访问 r 。它接着以前序来遍历 T_1 ,然后

以前序来遍历 T_2 ，依此类推，直到以前序来遍历了 T_n 为止。

读者应当验证，有序根树的前序遍历给出与利用通用地址系统所得出的顺序相同的顶点顺序。

图 7-26 说明如何执行前序遍历。

例 2 说明前序遍历。

例 2 前序遍历以什么顺序访问图 7-27 所示的有序根树里的顶点？

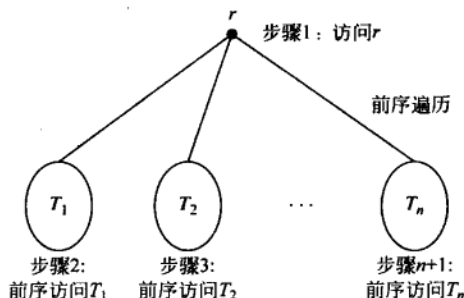


图 7-26 前序遍历

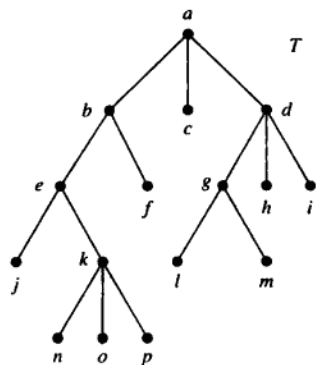


图 7-27 有序根树 T

解 T 的前序遍历的步骤如图 7-28 所示。这样以前序来遍历 T ，首先列出根 a ，接着依次是带根 b 的子树的前序列表，带根 c 的子树(它只有 c)的前序列表和带根 d 的子树的前序列表。

带根 b 的子树的前序列表首先列出 b ，然后以前序列出带根 e 的子树的顶点，然后以前序列出带根 f 的子树(它只有 f)的顶点。带根 d 的子树的前序列表首先列出 d ，接着是带根 g 的子树的前序列表，接着是带根 h 的子树(它只有 h)，接着是带根 i 的子树(它只有 i)。

带根 e 的子树的前序列表首先列出 e ，接着是带根 j 的子树(它只有 j)的前序列表，接着是带根 k 的子树的前序列表。带根 g 的子树的前序列表是 g 接着 l ，接着是 m 。带根 k 的子树的前序列表是 k, n, o, p 。所以， T 的前序遍历是 $a, b, e, j, k, n, o, p, f, c, d, g, l, m, h, i$ 。

现在将定义中序遍历。

定义 2 设 T 是带根 r 的有序根树。若 T 只包含 r ，则 r 是 T 的中序遍历。否则，假定 $T_1, T_2, \dots, T_n$ 是 T 里在 r 处从左向右的子树。中序遍历首先以前序来遍历 T_1 ，然后访问 r 。它接着以中序来遍历 T_2 ，中序遍历 T_3 ，依此类推，直到以中序来遍历了 T_n 为止。

图 7-29 说明如何执行中序遍历。例 3 说明对一棵特定的树，如何执行中序遍历。

例 3 中序遍历以什么顺序访问图 7-27 所示的有序根树 T 里的顶点？

解 T 的中序遍历的步骤显示在图 7-30 里。中序遍历首先是带根 b 的子树的中序遍历，然后是根 a ，带根 c 的子树(它只有 c)的中序列表和带根 d 的子树的中序列表。

带根 b 的子树的中序列表，首先是带根 e 的子树的中序列表，然后是根 b ，以及 f 。带根 d 的子树的中序列表，首先是带根 g 的子树的中序列表，接着是根 d ，接着是 h ，接着是 i 。

带根 e 的子树的中序列表是 j ，接着是根 e ，接着是带根 k 的子树的中序列表。带根 g 的子树的中序列表是 l, g, m 。带根 k 的子树的中序列表是 n, k, o, p 。所以，这个根树的中序遍历是 $j, e, n, k, o, p, b, f, a, c, l, g, m, d, h, i$ 。

现在定义后序遍历。

定义 3 设 T 是带根 r 的有序根树。若 T 只包含 r ，则 r 是 T 的后序遍历。否则，假定 $T_1, T_2, \dots, T_n$ 是 T 里在 r 处从左向右的子树。后序遍历首先以后序来遍历 T_1 ，然后以后序来遍历 $T_2, \dots$ 然后以后序来遍历 T_n ，最后访问 r 。

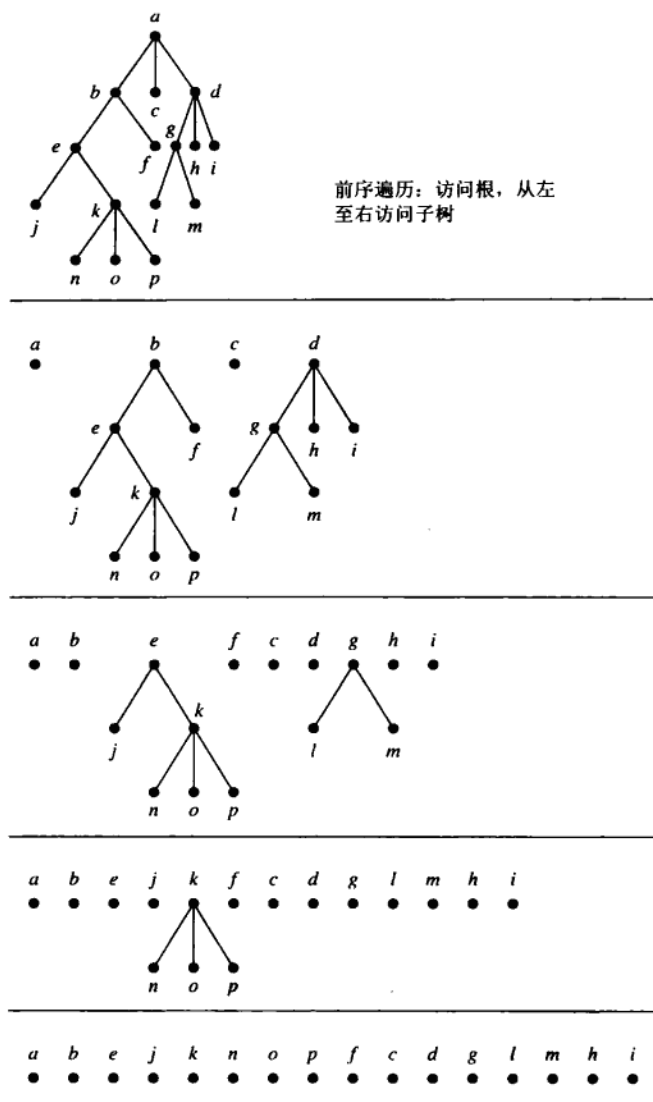
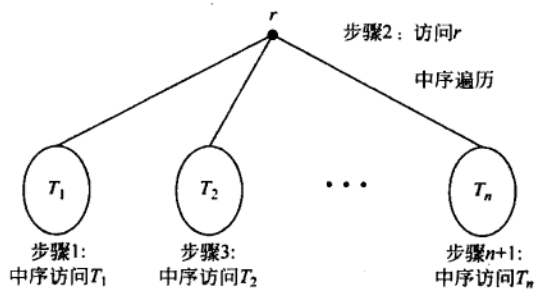
图 7-28 T 的前序遍历

图 7-29 中序遍历

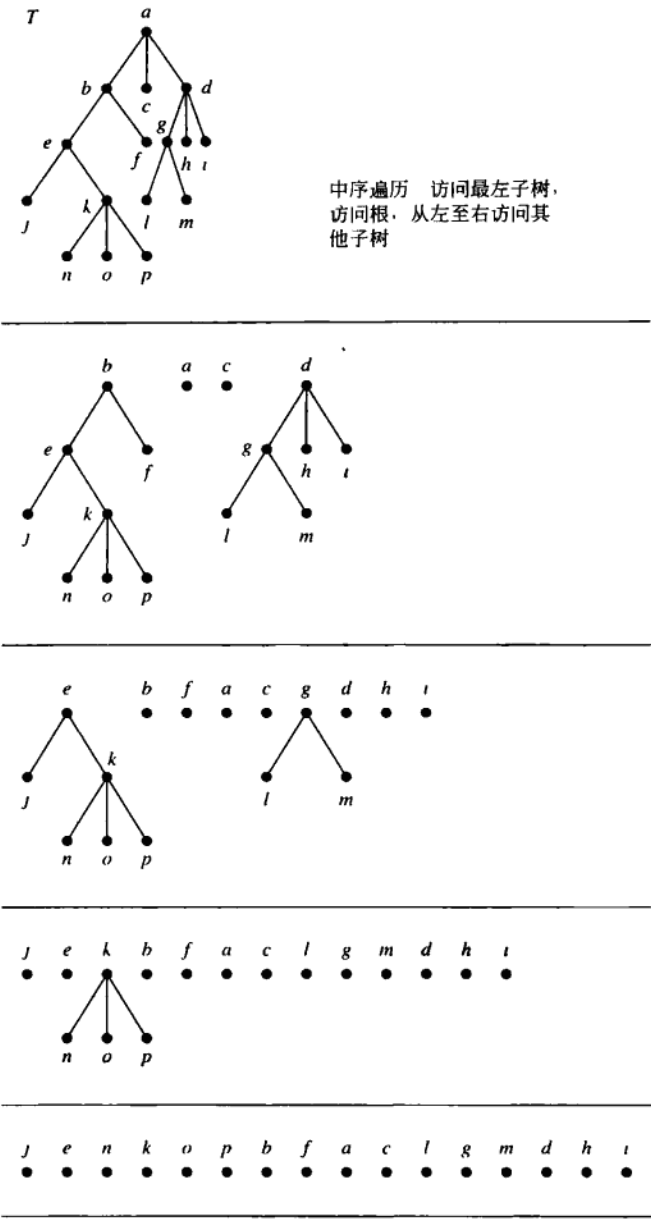


图 7-30 T 的中序遍历

图 7-31 说明后序遍历是如何执行的。例 4 说明后序遍历如何工作。

例 4 后序遍历以什么顺序访问图 7-27 所示有序根树 T 里的顶点?

解 T 的后序遍历的步骤显示在图 7-32 里。后序遍历首先是带根 b 的子树的后序遍历，然后是带根 c 的子树(它只有 c)的后序遍历，带根 d 的子树的后序遍历，接着是根 a 。

带根 b 的子树的后序遍历首先是带根 e 的子树的后序遍历，接着是 f ，接着是根 b 。带根 d 的

子树的后序遍历首先是带根 g 的子树的后序遍历，接着是 h ，接着是 i ，接着是根 d 。

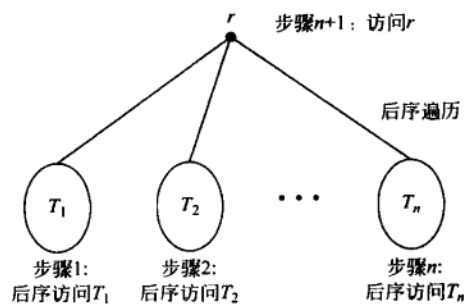


图 7-31 后序遍历

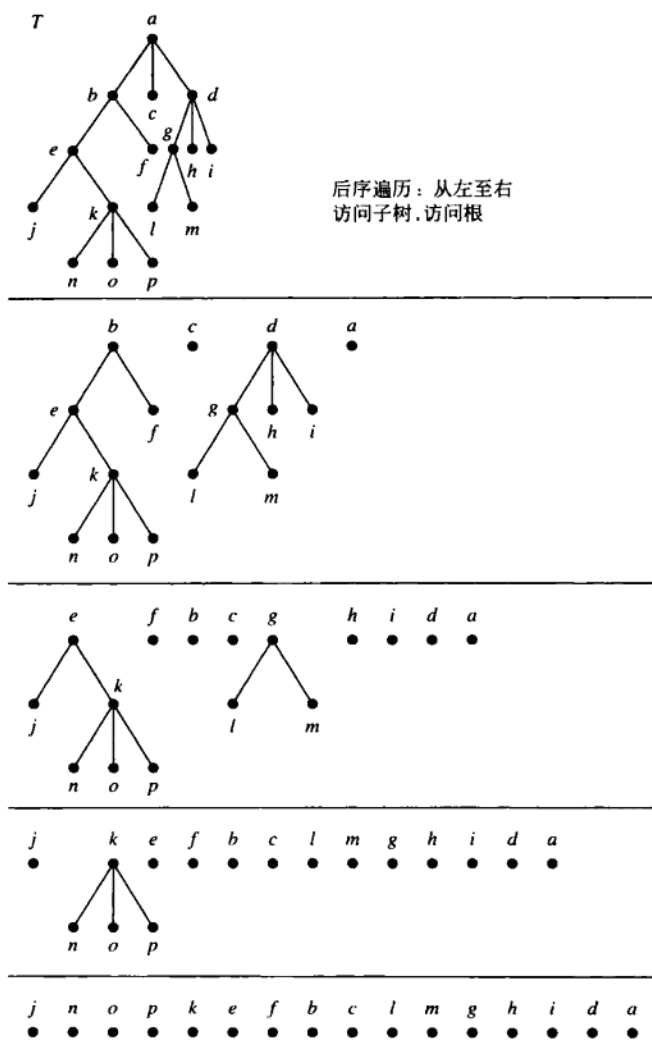


图 7-32 T 的后序遍历

带根 e 的子树的后序遍历是 j ，接着是带根 k 的子树的后序遍历，接着是根 e 。带根 g 的子树的后序遍历是 l, m, g 。带根 k 的子树的后序遍历是 n, o, p, k 。因此，根树 T 的后序遍历是 $j, n, o, p, k, e, f, b, c, l, m, g, h, i, d, a$ 。

有一些简易的方法以前序、中序和后序来列出有序根树的顶点。首先从根开始围绕有序根树画一条曲线，如图 7-33 里的例子所示，沿着边移动。可以这样按照前序来列出顶点：当曲线第一次经过一个顶点时，就列出这个顶点。可以这样按照中序来列出顶点：当曲线第一次经过一个树叶时，就列出这个树叶，当曲线第二次经过一个内点时就列出这个内点。可以这样按照后序来列出顶点：当曲线最后一次经过一个顶点而返回这个顶点的父母时，就列出这个顶点。当在图 7-33 的根树里这样做时，结果是前序遍历给出 $a, b, d, h, e, i, j, c, f, g, k$ ；中序遍历给出 $h, d, b, i, e, j, a, f, c, k, g$ ；后序遍历给出 $h, d, i, j, e, b, f, k, g, c, a$ 。

这些以前序、中序和后序来遍历有序根树的算法，最容易用递归来表示。

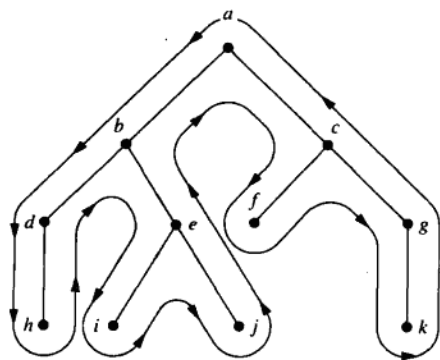


图 7-33 以前序、中序和后序来遍历有序根树的快捷方法

算法 1 前序遍历

```

procedure preorder( $T$ : 有序根树)
 $r := T$  的根
列出  $r$ 
for 从左到右的  $r$  的每个子女  $c$ 
begin
     $T(c) :=$  以  $c$  为根的子树
    preorder( $T(c)$ )
end
    
```

算法 2 中序遍历

```

procedure inorder( $T$ : 有序根树)
 $r := T$  的根
if  $r$  是树叶 then 列出  $r$ 
else
begin
     $l :=$  从左到右的  $r$  的第一个子女
     $T(l) :=$  以  $l$  为根的子树
    inorder( $T(l)$ )
    列出  $r$ 
    for 除  $l$  外从左到右的  $r$  的每个子女  $c$ 
         $T(c) :=$  以  $c$  为根的子树
        inorder( $T(c)$ )
end
    
```

算法 3 后序遍历

procedure *postorder*(*T*: 有序根树)

r := *T* 的根

for 从左到右的 *r* 的每个子女 *c*

begin

T(*c*) := 以 *c* 为根的子树

postorder(*T*(*c*))

end

列出 *r*

注意, 当规定了每个顶点的孩子数时, 有序根树的前序遍历和后序遍历的编码结构相同。也就是说, 当指定树的前序遍历或者后序遍历所生成的顶点列表和每个顶点的孩子数目, 有序根树是唯一确定的(见练习 13)。一般地, 所有有序 m 元树的前序遍历和后序遍历的编码结构相同。然而, 当不规定每个顶点的孩子数时, 有序根树的前序遍历和后序遍历编码结构都不同(见练习 14)。

7.3.4 中缀、前缀和后缀记法

可以用有序树来表示复杂的表达式, 比如复合命题, 集合的组合, 以及算术表达式。例如, 考虑由运算+(加)、-(减)、*(乘)、/(除)、 $\uparrow$ (幂)所组成的算术表达式的表示。我们将用括号来说明运算次序。有序根树可以用来表示这样的表达式, 其中内点表示运算, 树叶表示变量或数字。每个运算都作用在它的左子树和右子树上(以此顺序)。

例 5 表示表达式 $((x+y) \uparrow 2) + ((x-4)/3)$ 的有序根树是什么?

解 这个表达式的二叉树可以自底向上来构造。首先, 构造表达式 $x+y$ 的子树; 然后, 加入这个子树作为表示 $(x+y) \uparrow 2$ 的更大子树的一部分。同样, 构造表达式 $x-4$ 的子树, 然后, 加入这个子树到表示 $(x-4)/3$ 的子树里; 最后, 组合表示 $(x+y) \uparrow 2$ 与 $(x-4)/3$ 的子树来形成表示 $((x+y) \uparrow 2) + ((x-4)/3)$ 的有序根树。这些步骤都显示在图 7-34 里。 ■

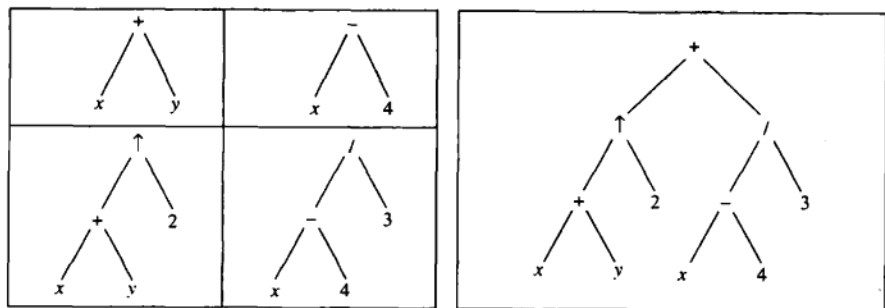
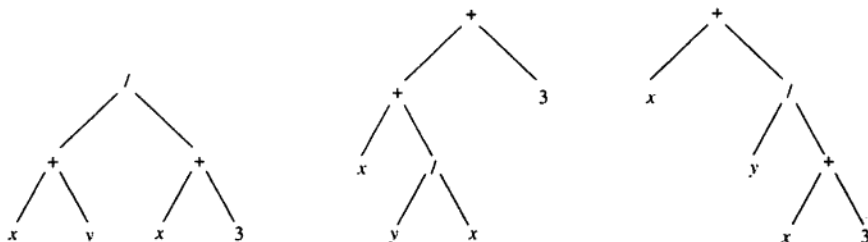


图 7-34 表示 $((x+y) \uparrow 2) + ((x-4)/3)$ 的二叉树

对表示一个表达式的二叉树的中序遍历, 产生原来的表达式, 其中元素和运算都是按原有的次序出现, 例外的是一元运算, 它们紧随运算对象。例如, 图 7-35 里的二叉树分别表示表达式 $(x+y)/(x+3)$, $(x+(y/x))+3$ 和 $x+(y/(x+3))$, 对它们的中序遍历都得出中缀表达式 $x+y/x+3$ 。为了让这样的表达式无二义性, 每当遇到运算时, 就有必要在中序遍历里包含括号。以这种方式获得的带完整括号的表达式称为中缀形式。

当以前序来遍历表达式的根树时, 就获得它的前缀形式。写成前缀形式的表达式称为波兰记法, 它的命名由来自于逻辑学家扬·武卡谢维奇(其实他是乌克兰人而非波兰人)。前缀记法下的表达式(其中每个运算都有规定的运算对象数)都是无二义性的。对这个事实的验证留给读者作为练习。

图 7-35 表示 $(x+y)/(x+3)$, $(x+(y/x))+3$ 和 $x+(y/(x+3))$ 的根树

例 6 $((x+y) \uparrow 2) + ((x-4)/3)$ 的前缀形式是什么?

解 通过遍历图 7-34 所示的表示这个表达式的二叉树, 就获得它的前缀形式。这样就产生 $+\uparrow +xy2/-x43$ 。

在表达式的前缀形式里, 二元运算(比如+)在它的两个运算对象之前。因此, 可以从右向左地求前缀形式的表达式的值。当遇到一个运算时, 就对在这个运算右边紧接着的两个运算对象来执行相应的运算。另外, 每当一个运算执行时, 就认为结果是新的运算对象。

例 7 前缀表达式 $+- * 235 / \uparrow 234$ 的值是什么?

解 用来从右向左地求这个表达式的值所用的步骤, 以及用右边的运算对象来执行的运算, 都如图 7-36 所示。这个表达式的值是 3。

通过以后序来遍历表达式的二叉树, 就获得它的后缀形式。写成后缀形式的表达式称为是逆波兰记法。逆波兰记法下的表达式都是无二义性的, 所以就不需要括号。对这个事实的验证留给读者。

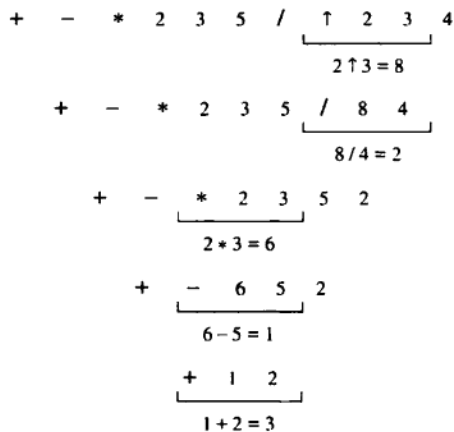
例 8 $((x+y) \uparrow 2) + ((x-4)/3)$ 的后缀形式是什么?

解 这个表达式的后缀形式是这样获得的: 执行图 7-34 所示的表示它的二叉树的后序遍历, 这样就产生后缀表达式 $xy+2 \uparrow x4-3/+$ 。

在表达式的后缀形式里, 二元运算都是在它的两个运算对象之后。所以, 为了从一个表达式的后缀形式来求它的值, 就从左向右地进行, 每当一个运算后面跟着两个运算对象时, 就执行这个运算。在一个运算执行之后, 这个运算的结果就成为一个新的运算对象。

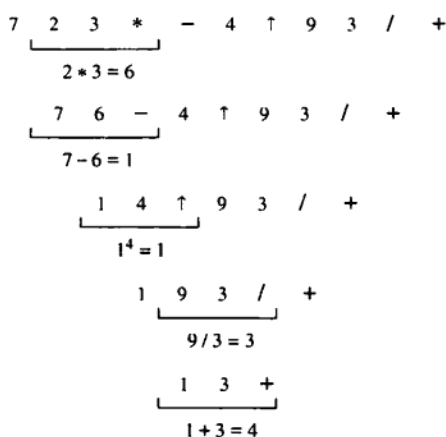
例 9 后缀表达式 $723 * -4 \uparrow 93 / +$ 的值是什么?

解 如图 7-37 所示, 求这个表达式的值所用的步骤是这样的: 从左边开始, 当两个运算对象后面接着一个运算时, 就执行这个运算。这个表达式的值是 4。



表达式的值: 3

图 7-36 求一个前缀表达式的值



表达式的值: 4

图 7-37 求一个后缀表达式的值

根树可以用来表示其他类型的表达式, 比如那些表示复合命题、集合组合的表达式。在这些例子里出现比如命题否定这样的一元运算。为了表示这样的运算及其运算对象, 就用顶点表示运算并且用这个顶点的子女表示运算对象。

例 10 求表示复合命题 $(\neg(p \wedge q)) \leftrightarrow (\neg p \vee \neg q)$ 的有序根树。然后用这个根树求这个表达式的前缀、后缀和中缀形式。

解 这个复合命题的有序根树是自底向上地构造的。首先, 构造 $\neg p$ 和 $\neg q$ 的子树(其中把 $\neg$ 当作一元运算)。另外, 构造 $p \wedge q$ 的子树。然后构造 $\neg(p \wedge q)$ 和 $(\neg p) \vee (\neg q)$ 的子树。最后, 用这两个子树来构造最终的根树。这个过程的步骤显示在图 7-38 中。

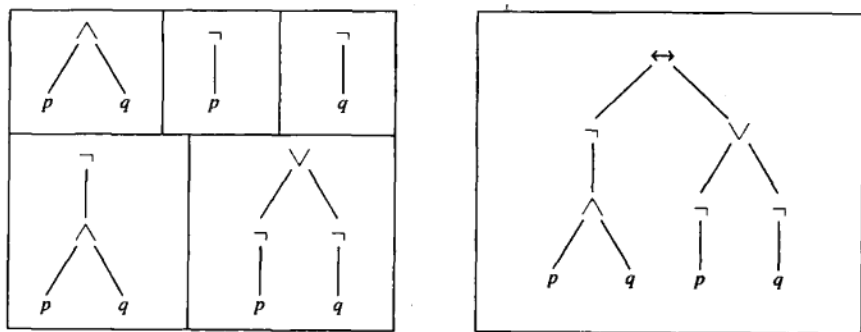


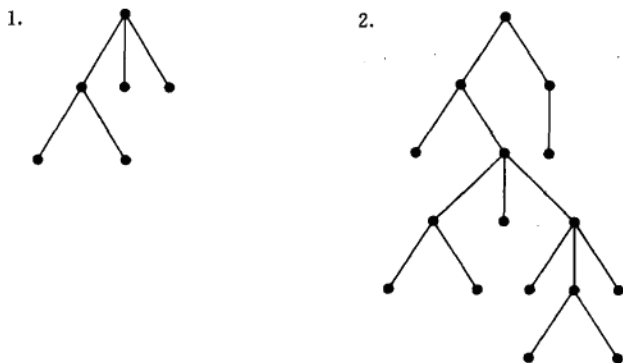
图 7-38 构造一个复合命题的根树

求表达式的前缀、后缀和中缀形式: 分别以前序、后序和中序来遍历这个根树(包含括号)。这些遍历分别给出 $\leftrightarrow \neg \wedge pq \vee \neg p \neg q$, $pq \wedge \neg p \neg q \neg \vee \leftrightarrow$ 和 $(\neg(p \wedge q)) \leftrightarrow ((\neg p) \vee (\neg q))$ 。

因为前缀表达式和后缀表达式都是无二义性的, 而且不用来回扫描就容易求出它们的值, 所以它们在计算机科学里大量使用。这样的表达式对编译器的构造是特别有用的。

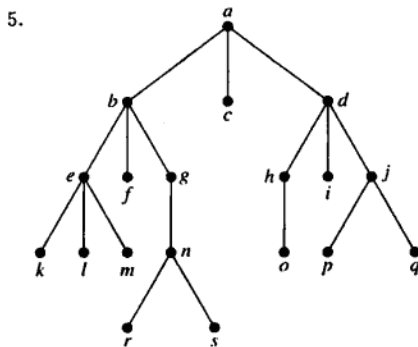
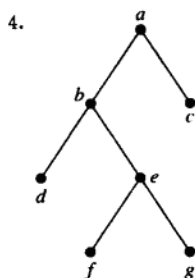
练习

在练习 1~2 里, 对给定的有序根树构造通用地址系统。然后利用这个通用地址系统并且用顶点的标记的字典顺序来排序顶点。

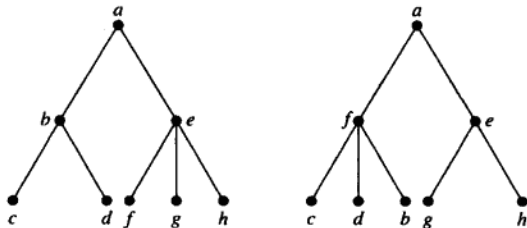


3. 假定在有序根树 T 里, 地址最大的顶点的地址是 2.3.4.3.1。是否有可能确定 T 里的顶点数?

在练习 4~5 里, 确定前序遍历访问所给的有序根树的顶点的顺序。



6. 使用后序遍历, 以什么顺序访问练习 4 里有序根树的顶点?
7. 使用后序遍历, 以什么顺序访问练习 5 里有序根树的顶点?
8. 用二叉树来表示表达式 $(x+xy)+(x/y)$ 和 $x+((xy+x)/y)$ 。表示方式应采用:
 - a) 前缀记法
 - b) 后缀记法
 - c) 中缀记法
9. 用有序根树来表示 $(A \cap B) - (A \cup (B - A))$ 。表示方式应采用:
 - a) 前缀记法
 - b) 后缀记法
 - c) 中缀记法
- *10. 有多少种方式给字符串 $A \cap B - A \cup B - A$ 完全加上括号以便产生中缀表达式?
11. 下面每个前缀表达式的值是什么?
 - a) $- * 2 / 8 4 3$
 - b) $\uparrow - * 3 3 * 4 2 5$
 - c) $+ - \uparrow 3 2 \uparrow 2 3 / 6 - 4 2$
 - d) $* + 3 + 3 \uparrow 3 + 3 3 3$
12. 构造前序遍历为 $a, b, f, c, g, h, i, d, e, j, k, l$ 的有序根树, 其中 a 有四个子女, c 有三个子女, j 有两个子女, b 和 e 都有一个子女, 所有其他顶点都是树叶。
- *13. 证明: 当规定了有序根树的后序遍历所生成的顶点列表, 并且规定了每个顶点的子女数时, 这个有序根树是唯一确定的。
14. 证明: 下图所示的两个有序根树的后序遍历产生相同的顶点列表。注意这个结果不与练习 27 里的命题相矛盾, 因为在这两个有序根树里内点的子女数是不同的。



在一组符号和一组二元运算上用前缀记法表示的合式公式是用下面的规则来递归地定义的：

- (i) 若 x 是符号, 则 x 是用前缀记法表示的合式公式;
(ii) 若 X 和 Y 都是合式公式并且 $*$ 是运算, 则 $*XY$ 是合式公式。
- *15. 证明: 在一组符号和一组二元运算上用前缀记法表示的任何合式公式所包含的符号数都比运算数恰好多一个。
16. 给出在符号 $\{x, y, z\}$ 和二元运算集 $\{+, \times, \circ\}$ 上带三个以上运算的、用后缀记法表示的合式公式的 6 个例子。

7.4 生成树

7.4.1 引言

考虑一下图 7-39a 所示的简单图所表示的缅因州的道路系统。在冬天里保持道路通畅的唯一方式就是经常扫雪。高速公路部门希望只扫尽可能少的道路上的雪，而确保总是存在连接任何两

个乡镇的干净道路。如何才能做到这一点？

至少扫除5条道路上的雪才能保证在任何两个乡镇之间有一条通路。图7-39b说明了这样一些道路。注意表示这些道路的子图是树，因为它是连通的并且包含6个顶点和5条边。

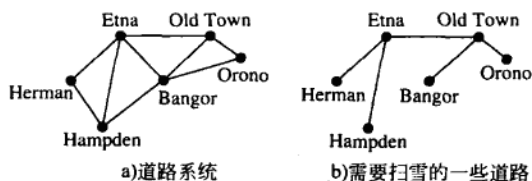


图 7-39

这个问题是用包含原来简单图的所有顶点、边数最小的连通子图来解决的。这样的图必然是树。

定义1 设 G 是简单图。 G 的生成树是包含 G 的每个顶点的 G 的子图。

有生成树的简单图必然是连通的，因为在任何两个顶点之间都有生成树里的通路。反过来也是对的；即每个连通图都有生成树。在证明这个结果之前将给出一个例子。

例1 找出图7-40所示简单图的生成树。

解 图 G 是连通的，但它不是树，因为它包含简单回路。删除边 $\{a, e\}$ 。这样就消除了一个简单回路，而且所得出的子图仍然是连通的并且仍然包含 G 的每个顶点。其次删除边 $\{e, f\}$ 以便消除第二个简单回路。最后，删除边 $\{c, g\}$ 以便产生一个没有简单回路的简单图。这个子图是生成树，因为它是包含 G 的每个顶点的树。图7-41中说明了用来产生这个生成树的边的删除序列。

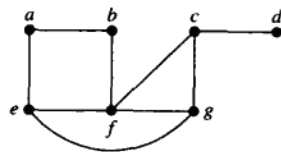


图 7-40 简单图 G

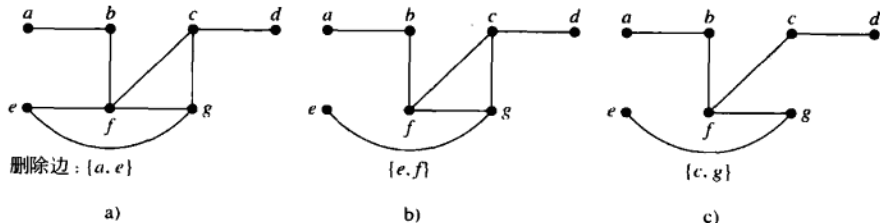


图 7-41 通过删除形成简单回路的边来产生 G 的一个生成树

图7-41所示的生成树不是唯一的 G 的生成树。例如，图7-42所示的每个树都是 G 的生成树。

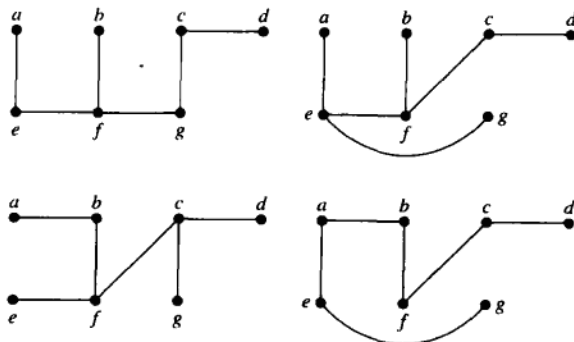


图 7-42 G 的一些生成树

定理 1 简单图是连通的当且仅当它具有生成树。

证 首先, 假定简单图 G 有生成树 T 。 T 包含 G 的每个顶点。另外, 在 T 的任何两个顶点之间都有 T 里的通路。因为 T 是 G 的子图, 所以在 G 的任何两个顶点之间都有 T 里的通路。因此, G 是连通的。

现在假定 G 是连通的。若 G 不是树, 则它必然包含简单回路。从这些简单回路中的一个里删除一条边。所得出的子图少了一条边, 但是仍然包含 G 的所有顶点并且是连通的。若这个子图不是树, 则它有简单回路; 所以像前面那样, 删除一个简单回路里的一条边。重复这个过程直到没有简单回路为止。这是可能的, 因为在图里只有有穷的边数。当没有简单回路剩下时, 这个过程终止。产生出一个树, 因为在删除边时这个图保持连通。这个树是生成树, 因为它包含 G 的每个顶点。 $\square$

例 2 说明, 在数据网络里生成树是重要的。

例 2 IP 组播 在网际协议(IP)网络上的组播里, 生成树的作用很重要。为了从源计算机发送数据到多个接收计算机, 每个接收计算机是一个子网, 可以分别发送数据到每个计算机。称为单点广播的这种类型的网络是无效的, 这是因为在网络上存有发送相同数据的多个副本。为了更有效地传送数据到多个接收计算机, 就使用 IP 组播。在 IP 组播里, 一个计算机在网络上发送数据的单一副本, 当数据到达中间路由器时, 就把数据分发给一个或更多的其他路由器, 以便接收计算机都在它们不同的子网里最终接收到这些数据。(路由器是专门在网络里子网之间分发 IP 数据报文的计算机。在组播里, 路由器使用 D 类地址, 每个都表示接收计算机可以加入的一个会话; 见 3.1 节例 16。)

为了让数据尽可能快地到达接收计算机, 在数据穿过网络的通路里就不应当存在环路(在图论名词里它们是回路)。即一旦数据已经到达一个具体的路由器, 数据就再也不应当返回这个路由器。为了避免环路, 组播路由器用网络算法来构造图 7-43 所示的生成树, 这个图以组播源、路由器和包含接收计算机的子网来作为顶点, 以边表示计算机和(或)路由器之间的连接。这个生成树的根就是组播源。包含接收计算机的子网就是这个树的树叶(注意不包含接收计算机的子网都不包含在这个图里)。在图 7-43 里说明这些内容。 $\blacksquare$

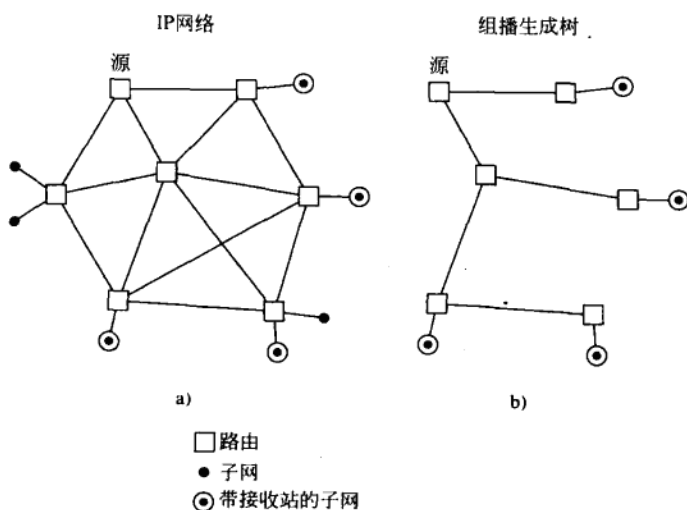


图 7-43 一个组播生成树

7.4.2 深度优先搜索

定理 1 的证明给出通过从简单回路删除边来找出生成树的算法。这个算法是低效的，因为它要求找出简单回路。另一种不采用删除边来构造生成树的方法是，通过相继地添加边来建立生成树。在这里将给出基于这个原理的两个算法。

可以用深度优先搜索来建立连通简单图的生成树。我们将形成一个根树，而生成树将是这个根树的无向底图。任意选择图中一个顶点作为根。通过相继地添加边来形成从这个顶点开始的通路，其中每条新边都与通路上的最后一个顶点以及还不通路上一个顶点相关联。继续尽可能地添加边到这条通路。若这条通路经过图的所有顶点，则由这条通路组成的树就是生成树。不过，若这条通路没有经过图的所有顶点，则必须添加其他的边。退到通路里的倒数第二个顶点，若有可能，则形成从这个顶点上开始的经过还没有访问过的顶点的通路。若不能这样做，则后退到通路里的另外一个顶点，即在通路里后退两个顶点，然后再试。

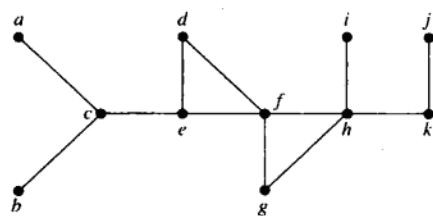
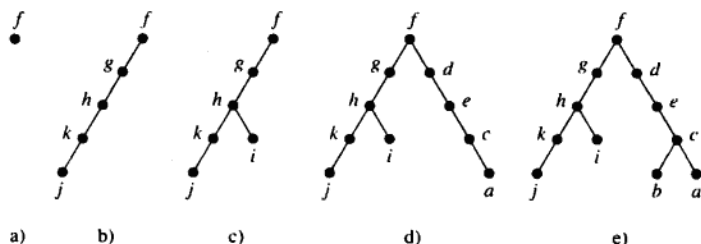
重复这个过程，从所访问过的最后一个顶点开始，在通路上一次后退一个顶点，只要有可能就形成新的通路，直到不能添加更多的边为止。因为这个图有有穷的边数并且是连通的，所以这个过程以产生生成树而告终。在这个算法的一个阶段上是通路末端的顶点将是根树里的树叶，而在其上开始构造一条通路的顶点将是内点。

读者应当注意到这个过程的递归本质。另外，注意若图中的顶点是排序的，则当总是选择在该顺序里可用的第一个顶点时，在这个过程的每个阶段上对边的选择就全都是确定的。不过，将不总是明显地对图的顶点排序。

深度优先搜索也称为回溯，因为这个算法返回以前访问过的顶点以便添加边。下面的例子说明回溯。

例 3 用深度优先搜索来找出图 7-44 所示图 G 的生成树。

解 在图 7-45 里显示深度优先搜索为了产生 G 的生成树而使用的步骤。任意地从顶点 f 开始。一条通路是这样建立的：相继地添加与还不通路上的顶点相关联的边，只要有可能就这样做。这样就产生通路 f, g, h, k, j (注意也可能建立其他的通路)。下一步，回溯到 k 。不存在从 k 开始包含还没有访问过的顶点的通路。所以回溯到 h 。形成通路 h, i 。然后回溯到 g ，然后再回溯到 f 。从 f 建立通路 f, d, e, c, a 。然后再回溯到 c 并且形成通路 c, b 。这样就产生了生成树。 ■

图 7-44 图 G 图 7-45 G 的深度优先搜索

一个图的深度优先搜索所选择的边都称为树边。这个图所有其他的边都必然连接一个顶点与这个顶点在树中的祖先或后代。这些边都称为背边(练习 20 要求证明这个事实)。

例 4 图 7-46 中强调了从顶点 f 开始的深度优先搜索所找到的树边，用粗线显示这些树边。用细黑线显示背边 (e, f) 和 (f, h) 。 ■

我们已经解释了如何用深度优先搜索来求图的生成树。但是，迄今为止的讨论还没有带出深度优先搜索的递归本质。为了弄清深度优先搜索的递归本质，需要几个术语。当执行深度优先搜索的步骤时，当把顶点 v 加入树时说从顶点 v 开始探索，当最后一次回溯回到 v 时说从顶点 v 结束探索。理解算法的递归本质所需要的关键事实是，当加入连接顶点 v 到顶点 w 的边时，在回到 v 完成从 v 的探索之前就结束了从 w 的探索。在算法 1 中构造了带顶点 $v_1, \dots, v_n$ 的图 G 的生成树，首先选择顶点 v_1 作为树根。开始时令 T 是只有这一个顶点的树。在每个步骤，加入一个新顶点到 T 以及从已在 T 中的一个顶点到这个新顶点的一条边，并且从这个新顶点探索。注意当算法完成时， T 没有简单回路，因为没有加入连接到已在树中的顶点的边。另外， T 在构造时保持连通(用数学归纳法可以轻而易举地证明最后这两个事实)。由于 G 是连通的， G 的每个顶点都被算法访问到并且加入到树中(读者可以验证)。所以 T 是 G 的生成树。

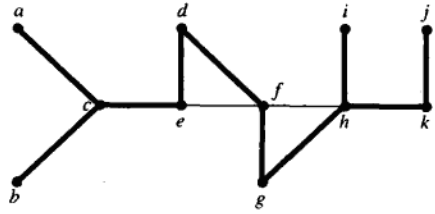


图 7-46 例 4 中深度优先搜索的树边和背边

算法 1 深度优先搜索

procedure DFS(G : 带顶点 $v_1, \dots, v_n$ 的连通图)

$T :=$ 只包含顶点 v_1 的树

$visit(v_1)$

procedure $visit(v$: G 的顶点)

for 与 v 相邻并且还不在于 T 中的每个顶点 w

begin

加入顶点 w 和边 $\{v, w\}$ 到 T

$visit(w)$

end

现在分析深度优先搜索算法的计算复杂性。关键事实是对于每个顶点 v 来说，当在搜索中首次遇到顶点 v 时，就调用过程 $visit(v)$ 并且以后不再调用这个过程。假设 G 的邻接表是可用的(参见 6.3 节)，求出与 v 相邻的顶点不需要任何计算。当遵循算法的步骤时，至多检查每条边两次以确定是否加入这条边及其一个端点到树中。因此，过程 DFS 用 $O(e)$ 或 $O(n^2)$ 个步骤来构造一个生成树，其中 e 和 n 分别是 G 的边数和顶点数。(注意一个步骤包括：检查一个顶点是否已在正在构造的树中，如果这个顶点还不在于树中，则加入这个顶点和对应的边。还利用了不等式 $e \leq n(n-1)/2$ ，对于任意简单图来说这个不等式都成立。)

深度优先搜索可以作为解决许多不同问题的算法的基础。例如，可以用来求图的连通分支并且可以用来求连通图的割点。将要看到，深度优先搜索是用来搜索计算困难问题的解的回溯技术的基础(参见 [GrYe99]、[Ma89] 和 [CoLeRiSt01] 对基于深度优先搜索的算法的讨论)。

7.4.3 宽度优先搜索

也可以通过使用宽度优先搜索来产生简单图的生成树。同样，将构造一个根树，而这个根树的无向底图就形成生成树。从图的顶点中任意地选择一个根。然后添加与这个顶点相关联的所有边。在这个阶段所添加的新顶点成为生成树在 1 层上的顶点。将新顶点任意排序。下一步，按顺序访问 1 层上的每个顶点，只要不产生简单回路，就将与这个顶点相关联的每条边添加到树里。这样就产生了树在 2 层上的顶点。遵循相同的过程，直到已经添加了树里的所

有顶点。因为在图中的边数是有限的，所以这个过程会终止。在产生了包含图中每一个顶点的树之后，生成树也就产生了。下面是宽度优先搜索的一个例子。

例 5 用宽度优先搜索来找出图 7-47 所示的图的生成树。

解 在图 7-48 里显示宽度优先搜索过程的各步骤。
选择顶点 e 作为根。然后添加与 e 相关联的所有边，所以添加了从 e 到 b, d, f 和 i 的边。这四个顶点都是在树的 1 层上。下一步，添加从 1 层上的顶点到还不在于树上的相邻顶点的边。因此，添加从 b 到 a 和 c 的边，以及从 d 到 h ，从 f 到 j 和 g ，从 i 到 k 的边。新顶点 a, c, h, j, g 和 k 都是在 2 层上。下一步，添加从这些顶点到还不在于树上的相邻顶点的边。这样就添加从 g 到 l 以及从 k 到 m 的边。

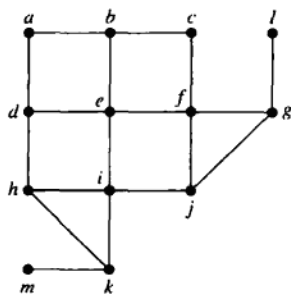


图 7-47 图 G

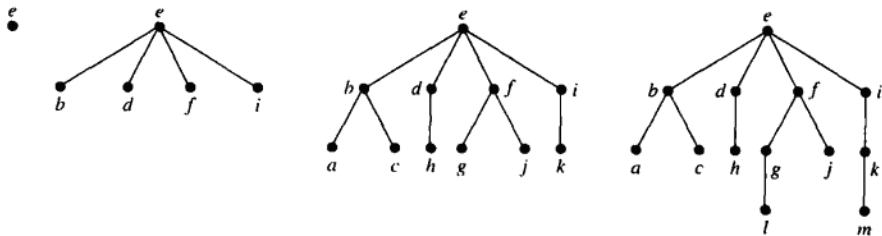


图 7-48 G 的宽度优先搜索

算法 2 的伪码描述了宽度优先搜索。在这个算法中假设连通图 G 的顶点排序为 $v_1, \dots, v_n$ 。在算法中，只要还没有产生简单回路就加入新的顶点和对应的边到树中，与正在处理的当前顶点相邻。

算法 2 宽度优先搜索

procedure BFS(G ; 带顶点 $v_1, \dots, v_n$ 的连通图)

$T :=$ 只包含顶点 v_1 的树

$L :=$ 空表

把 v_1 放入尚未处理顶点的表 L 中

while L 非空

begin

 删除 L 中第一个顶点 v

for v 的每个邻居 w

if w 既不在 L 中也不在 T 中 **then**

begin

 加入 w 到表 L 的末尾

 加入 w 和边 $\{v, w\}$ 到 T

end

end

现在分析宽度优先搜索的计算复杂性。对于图中的每个顶点 v 来说，检查所有与 v 相邻的顶点并且加入每个尚未访问过的顶点到树 T 中。假设图的邻接表是可用的，确定哪些顶点与给定顶点相邻就不需要任何计算。如同在深度优先搜索算法的分析中那样，我们检查每条边至多两次来确定是否应当加入这条边及其尚未在树中的端点。所以宽度优先搜索算法使用 $O(e)$ 或 $O(n^2)$ 个步骤。

7.4.4 回溯

有些问题只能通过执行对所有可行解的穷举搜索来解决。系统地搜索出一个解的一种方式是使用决策树，其中每个内点都表示一次决策，而每个树叶都表示一个可行解。为了通过回溯来求出一个解，首先尽可能地做出一系列决策来尝试得出一个解。可以用决策树里的通路来表示决策序列。一旦知道了决策序列的任何扩展都不能得出解，就回溯到父母顶点并且若有可能，则用另外一个决策序列来尝试得出一个解。继续这个过程，直到找到一个解，或者证明没有解存在为止。下面的例子说明回溯的有用性。

例6 图着色 如何用回溯来判定是否可以用 n 种颜色给一个图着色？

解 以下面的方式用回溯来解决这个问题。首先选择某个顶点 a 并且指定它颜色为 1。然后挑选第二个顶点 b ，而且若 b 不与 a 相邻，则指定它颜色为 1。否则，指定 b 颜色为 2。然后来到第三个顶点 c 。若有可能则对 c 用颜色 1。否则若有可能则用颜色 2。只有当颜色 1 和颜色 2 都不能用时才使用颜色 3。继续这个过程，只要有可能就为每个新顶点指定 n 种颜色中的一种，而且总是使用表中第一种允许的颜色。若遇到不能用 n 种颜色中任何一种来着色的顶点，则回溯到最后一次所做的指定，并且若有可能就改变最后着色的顶点的颜色，用表中下一种允许的颜色。若不可能改变这个颜色，则再回溯到更前面的指定，一次后退一步，直到有可能改变一个顶点的颜色为止。然后只要有可能就继续指定新顶点的颜色。若使用 n 种颜色的着色存在，则可以通过回溯来产生（但是这个过程是极其低效的）。

具体地说，考虑用三种颜色来着色图 7-49 所示的图。图 7-49 所示的树说明如何用回溯来构造 3 着色。在这个过程中，首先用红色，其次用蓝色，最后用绿色。显然用回溯来解决这个简单的例子只是为了能够比较好地说明这项技术。

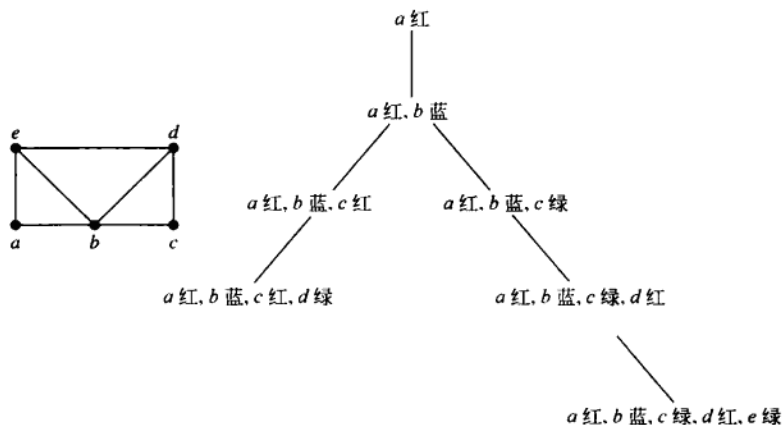


图 7-49 用回溯给图着色

在这个树里，从根开始的、表示指定红色给 a 的最初的通路，导致 a 红色、 b 蓝色、 c 红色而 d 绿色的着色。当以这种方式来着色 a , b , c 和 d 时，就不可能用这三种颜色中的任何一种来着色 e 。所以，回溯到表示这个着色的顶点的父母。因为没有其他颜色可以用在 d 上，所以再回溯一层。然后改变 c 的颜色为绿色。通过接着指定红色给 d 和绿色给 e ，就获得这个图的着色。 ■



例7 n 皇后问题 n 皇后问题问：在 $n \times n$ 棋盘上如何放置 n 个皇后，使得没有两个皇后可以互相攻击。如何用回溯来解决 n 皇后问题？

解 为了解决这个问题，必须在 $n \times n$ 棋盘上找出 n 个位置，使得这些位置中没有两个是在同一行上、同一列上或同一斜线上（斜线是由对某个 m 来说满足 $i + j = m$ 或对某个 m 来说满足 $i - j = m$ 的所有位置 (i, j) 组成的）。将用回溯来解决 n 皇后问题。从空棋盘开始。在 $k+1$ 阶段，

尝试在棋盘上第 $k+1$ 列里放置一个新皇后，其中在前 k 列里已经有了皇后。检查第 $k+1$ 列里的格子，从第一行的格子开始，寻找放置这个皇后的位置，使得它不与已经在棋盘上的皇后在同一行里或同一斜线上(已经知道它不在同一列里)。若不可能在第 $k+1$ 列里找到放置皇后的位置，则回溯到在第 k 列里对皇后的放置。在这一列里下一个允许的行里放置皇后，若这样的行存在的话。若没有这样的行存在，则继续回溯。

具体地说，图 7-50 显示四皇后问题的回溯解法。在这种解法里，在第一行第一列里放置一个皇后。然后在第二列的第三行里放置一个皇后。不过，这样就使得不可能在第三列里放置一个皇后。所以就回溯并且在第二列的第四行里放置一个皇后。当这样做时，就可以在第三列的第二行里放置一个皇后。但是没有办法在第四列里添加一个皇后。这说明当在第一行第一列里放置一个皇后时就得不到解。回溯到空棋盘，在第一列的第二行里放置一个皇后。这样就得出图 7-50 所示的解。

例 8 子集和 考虑下面的问题。给定一组正整数 $x_1, x_2, \dots, x_n$ ，求这组整数的和为 M 的一个子集。如何用回溯来解决这个问题？

解 从空无一项的和来开始。通过相继地添加项来构造这个和。若当添加这个序列里的一个整数到和里而这个和仍然小于 M 时，则包含这个整数。若得出使得添加任何一项就大于 M 的一个和，则通过去掉这个和的最后一项来回溯。

图 7-51 显示下面这个问题的回溯解法，求 $\{31, 27, 15, 11, 7, 5\}$ 的和等于 39 的子集。

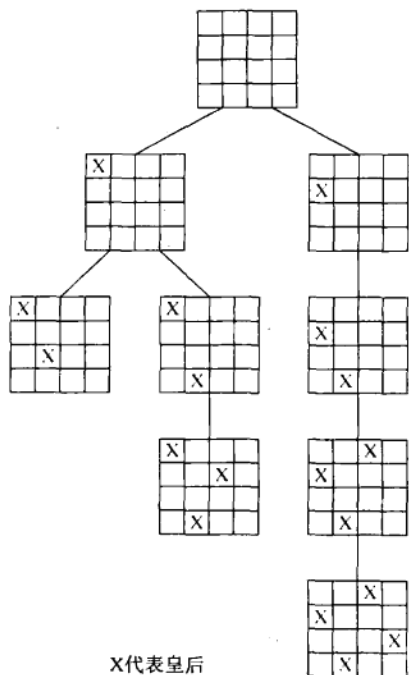


图 7-50 四皇后问题的回溯解法

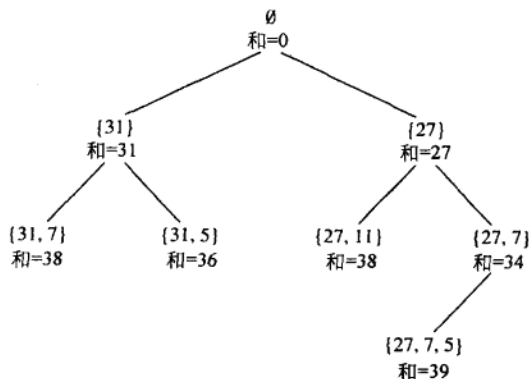


图 7-51 用回溯求等于 39 的和

7.4.5 有向图中的深度优先搜索

可以轻而易举地修改深度优先搜索和宽度优先搜索，使得以有向图作为输入时它们也能运行。但是，输出不一定是生成树，而可能是森林。在这两个算法中，只有当一条边从正在考虑的顶点发出并且到一个尚未加入的顶点时才加入这条边。如果在其中任何一个算法的某个阶段找不到从已经加入的顶点到尚未加入的顶点的边，则算法加入的下一个顶点成为生成森林中一个新树的根。

这一点在例 9 中解释。

例 9 给定图 7-52a 所示的图作为输入，深度优先搜索的输出是什么？

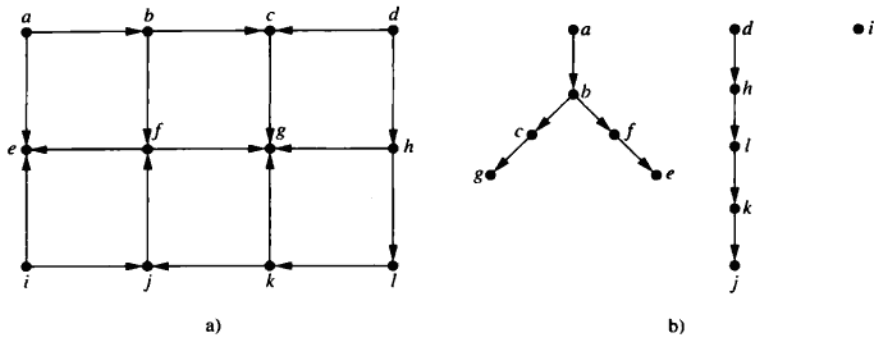


图 7-52 有向图的深度优先搜索

解 从顶点 a 开始深度优先搜索并且加入顶点 b 、 c 和 g 以及无路可走时对应的边。回溯到 c ，但是仍然无路可走，于是回溯到 b ，在这里加入顶点 f 和 e 以及对应的边。回溯最终又回到 a 。然后在 d 开始一个新的树并且加入顶点 h 、 l 、 k 和 j 以及对应的边。回溯到 k ，然后到 l ，然后到 h 并且回到 d 。最后，在 i 开始一个新的树，完成深度优先搜索。输出如图 7-52b 所示。

有向图中的深度优先搜索是许多算法的基础(参见[GrYe99]、[Ma89]和[CoLeRiSt01])。它可以用来确定有向图是否具有回路，可以用来完成图的拓扑排序，也可以用来求有向图的强连通分支。

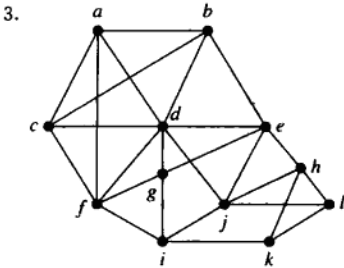
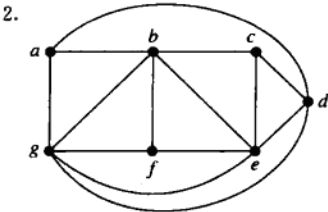
用深度优先搜索和宽度优先搜索在网络搜索引擎上的应用来结束本节。

例 10 网络蜘蛛 为了给网站建立索引，诸如 Hotbot 和 Lycos 这些著名的搜索引擎从已知的站点开始系统地探索网络。这些搜索引擎使用所谓网络蜘蛛(或网络爬虫、网络机器人)的程序来访问网站并且分析其内容。网络蜘蛛同时使用深度优先搜索和宽度优先搜索来创建索引。如 6.1 节例 8 所述，可以用所谓网络图的有向图来为网页和网页之间的链接建立模型。用顶点表示网页并且用有向边表示链接。利用深度优先搜索，选择一个初始的网页，循着一个链接(如果存在这样的链接的话)到达第二个网页，循着第二个网页的一个链接(如果存在这样的链接的话)到达第三个网页，等等，直到找到一个没有新的链接的网页为止。然后使用回溯来检查前面阶段的链接去寻找新的链接，等等。(由于实际限制，网络蜘蛛在深度优先搜索中的搜索深度是有限的。)利用宽度优先搜索，选择一个初始的网页并且遵循这个网页上的一个链接到达第二个网页，然后遵循初始网页上的第二个链接(如果存在的话)，等等，直到已经遵循了初始网页上的所有链接为止。然后逐页地遵循下一层网页上的链接，等等。

练习

1. 为了产生生成树，必须从带有 n 个顶点和 m 条边的连通图里删除多少条边？

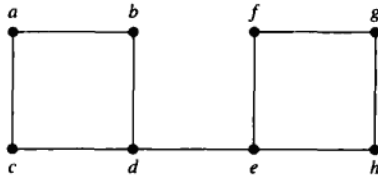
在练习 2~3 里，通过删除简单回路里的边来求所示的图的生成树。



4. 求下面每个图的生成树。

- a) K_3 b) $K_{4,4}$ c) $K_{1,6}$ d) Q_3 e) C_5 f) W_5

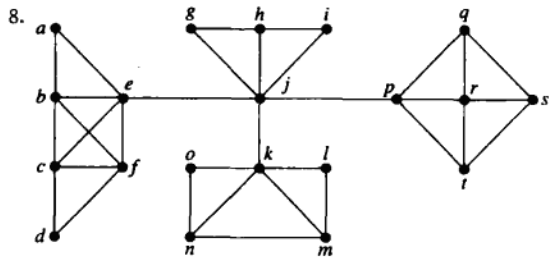
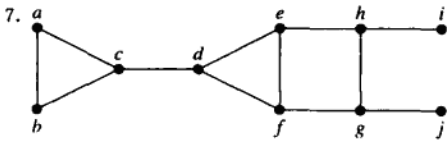
5. 画出下面所给的简单图的生成树。



*6. 下面的每个简单图各有多少个不同的生成树?

- a) K_3 b) K_4 c) $K_{2,2}$ d) C_5

在练习 7~8 里, 用深度优先搜索来构造所给的简单图的生成树。选择 a 作为这个生成树的根并且假定顶点都以字母顺序来排序。



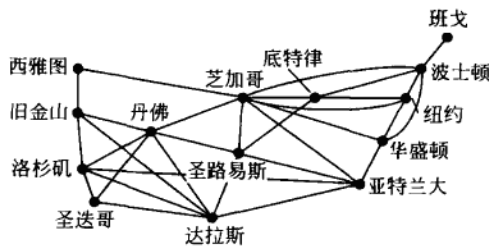
9. 用深度优先搜索求下列这些图的生成树。

- a) W_5 (参见 6.2 节例 6), 从 6 度顶点开始 b) K_5
c) $K_{3,4}$, 从 3 度顶点开始 d) Q_3

10. 描述一下轮图 W_n 的宽度优先搜索和深度优先搜索所产生的树, 从 n 度顶点开始, 其中 n 是整数满足 $n \geq 3$ (参见 6.2 节例 6)。说明答案的合理性。

11. 描述一下完全偶图 $K_{m,n}$ 的宽度优先搜索和深度优先搜索所产生的树, 其中 m 和 n 都是正整数。说明答案的合理性。

12. 假定一家航空公司必须压缩它的航班以节省资金。若它原来的航线如下图所示, 则可以中断哪些飞行以保持所有城市对之间的服务 (其中从一个城市飞往另一个城市可能需要换乘飞机)?



13. 哪些连通简单图恰好有一个生成树?

*14. 证明: 在连通简单图里顶点 v 和 u 之间的最短通路的长度, 等于在以 v 为根的 G 的宽度优先生成树里 u 的层数。

15. 用回溯来对下面的 n 值解决 n 皇后问题。

- a) $n=3$ b) $n=5$ c) $n=6$

16. 解释一下如何用回溯来找出图中的哈密顿通路或哈密顿回路。

图 G 的生成森林是包含 G 的每个顶点的森林, 使得当在两个顶点之间有 G 里的通路时, 这两个顶点就在同一个树里。

17. 证明: 每个有穷简单图都有生成森林。

18. 对带有 n 个顶点、 m 条边和 c 个连通分支的图来说, 必须删除多少条边才能产生它的生成森林?
19. 设计基于深度优先搜索来构造图的生成森林的算法。
20. 设 G 是连通图。证明: 如果 T 是用深度优先搜索构造的 G 的生成树, 则 G 的不在 T 中的边必定是背边, 换句话说, 这条边必定连接一个顶点到这个顶点在 T 中的祖先或后代。
21. 对于哪些图来说, 无论选择哪个顶点作为树根, 深度优先搜索和宽度优先搜索都产生同样的生成树? 说明答案的合理性。
22. 用数学归纳法证明: 宽度优先搜索按照顶点在所得出的生成树中的层数的顺序来访问这些顶点。
23. 用伪码来描述宽度优先搜索的一个变种, 它把整数 m 指定给在搜索中访问的第 m 个顶点。
24. 证明: 如果 G 是有向图并且 T 是用深度优先搜索构造的生成树, 则不在这个生成树上的每条边都是连接祖先到后代的前进边、连接后代到祖先的后退边, 或者连接一个顶点到从前访问过的子树的一个顶点的交叉边。

设 T_1 和 T_2 都是一个图的生成树。在 T_1 和 T_2 之间的距离是在 T_1 和 T_2 里非 T_1 和 T_2 所共有的边的数目。

25. 求图 7-40 所示图 G 的在图 7-41c 和图 7-42 里所示的每对生成树之间的距离。
- **26. 假定 T_1 和 T_2 都是简单图 G 的生成树。另外, 假定 e_1 是在 T_1 里但不在 T_2 里的一条边。证明: 存在着在 T_2 里但不在 T_1 里的一条边 e_2 , 使得若从 T_1 里删除 e_1 而添加 e_2 到 T_1 里, 则 T_1 仍然是生成树, 并且若从 T_2 里删除 e_2 而添加 e_1 到 T_2 里, 则 T_2 仍然是生成树。

有向图的根生成树是由这个图的边组成的根树, 使得这个图的每个顶点都是树里一条边的终点。

27. 对 6.5 节练习 10~12 里的每个有向图来说, 求这个图的根生成树, 或者确定不存在这样的树。
- *28. 给出构造每个顶点的入度和出度都相等的连通有向图的根生成树的算法。

7.5 最小生成树

7.5.1 引言

一个公司计划建立一个通信网络来连接它的五个计算机中心。可以用租用的电话线连接这些中心的任何一对。应当建立哪些连接, 以便保证在任何两个计算机中心之间都有通路, 且网络的总成本最小? 可以用图 7-53 所示的带权图为此问题建模, 其中顶点表示计算机中心, 边表示可能租用的电话线, 边上的权是边所表示的电话线的月租费。通过找出一棵生成树, 使得这棵树的各边的权之和为最小, 就可以解决这个问题。这样的生成树称为最小生成树。

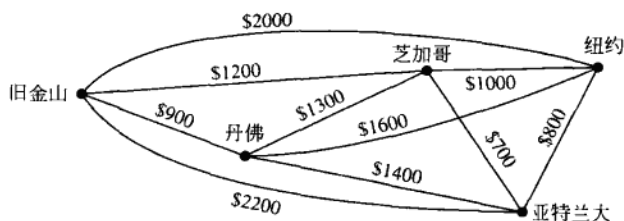


图 7-53 说明计算机网络里的线路的月租费的带权图

7.5.2 最小生成树算法

大量的问题可以这样解决: 求带权图里的一棵生成树, 使得这棵树的各边的权之和为最小。

定义 1 连通带权图里的最小生成树是具有最小可能的边的权之和的生成树。

下面将给出构造最小生成树的两个算法。这两个算法都是通过添加还没有使用过的、具有规定性质的、权最小的边来进行的。这些算法都是贪心算法的例子。贪心算法是在每个步骤上都做最优选择的算法。在算法的每个步骤上都最优化, 并不保证产生全局最优解。不过, 本节里给出的构造最小生成树的这两个算法都是产生最优解的贪心算法。

要讨论的第一个算法是罗伯特·普林在 1957 年给出的, 虽然这个算法的基本想法有更早的起源。为了执行普林算法, 首先选择带最小权的边, 把它放进生成树里。相继地向树里添加与已在树里的顶点关联的、并且不与已在树里的边形成简单回路的权最小的边。当已

经添加了 $n-1$ 条边时就停止。

本节稍后将证明这个算法产生任何连通带权图的最小生成树。算法 1 给出普林算法的伪码描述。

算法 1 普林算法

procedure *Prim*(G : 带 n 个顶点的连通无向图)

$T :=$ 权最小的边

for $i := 1$ **to** $n-2$

begin

$e :=$ 与 T 里顶点关联、并且若添加到 T 里则不形成简单回路的权最小的边

$T :=$ 添加 e 之后的 T

end (T 是 G 的最小生成树)

注意, 当有超过一条满足相应条件的带相同权的边时, 在算法的这个阶段里对所添加的边的选择就不是确定的。需要排序这些边以便让选择是确定的。在本节剩下的部分里将不再考虑这个问题。另外注意, 所给的连通带权简单图可能有多于一个的最小生成树(见练习 5)。例 1 和例 2 说明如何使用普林算法。

例 1 用普林算法设计连接图 7-53 所表示的所有计算机的、具有最小成本的通信网络。

解 办法是求图 7-53 的最小生成树。普林算法是这样执行的: 选择权最小的初始边, 并且相继地添加与树里顶点关联的不形成回路的权最小的边。在图 7-54 里, 加粗的边表示普林算法所产生的最小生成树, 并且显示出在每个步骤上所做的选择。

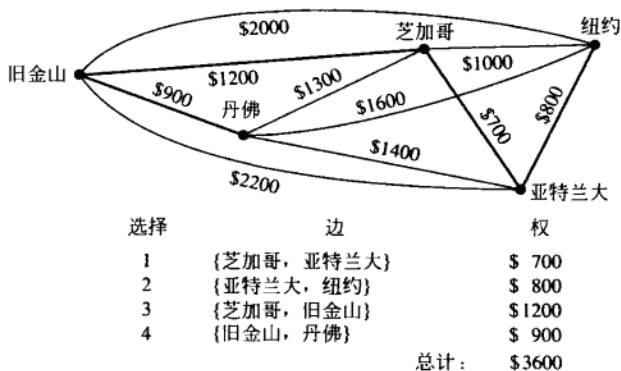


图 7-54 图 7-53 里带权图的最小生成树

例 2 用普林算法求图 7-55 所示的图的最小生成树。

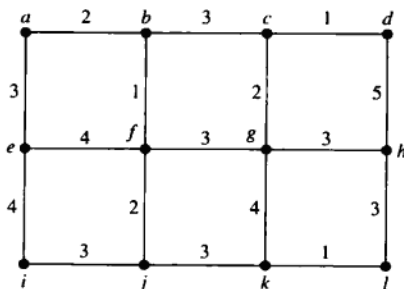


图 7-55 一个带权图

解 用普林算法所构造的最小生成树显示在图 7-56 里。相继选择的边都有显示。

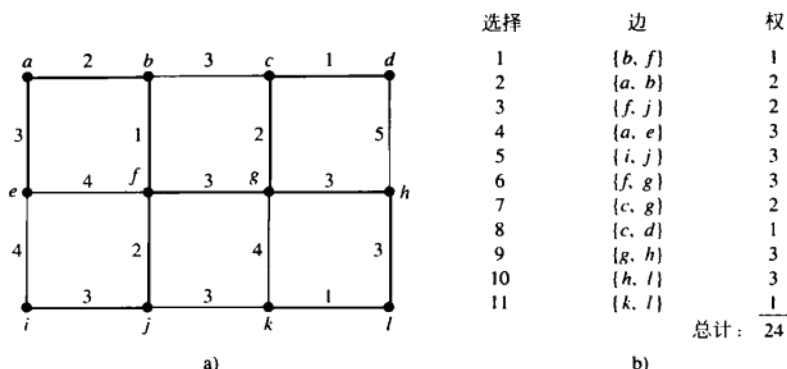


图 7-56 用普林算法构造的最小生成树

将要讨论的第二个算法是约瑟夫·克鲁斯卡尔在1956年发现的，但在此之前已经有人阐述过这一算法的基本思路。为了执行克鲁斯卡尔算法，要选择图中权最小的一条边。

相继添加不与已经选择的边形成简单回路的权最小的边。在已经挑选了 $n-1$ 条边之后就停止。

在本节末尾，把证明克鲁斯卡尔算法对每个连通带权图都产生最小生成树留作练习。在算法2里给出克鲁斯卡尔算法的伪代码。

算法2 克鲁斯卡尔算法

procedure *Kruskal*(G : 带 n 个顶点的带权连通无向图)

$T :=$ 空图

for $i := 1$ **to** $n-1$

begin

$e :=$ 当添加到 T 里时不形成简单回路的权最小的 G 里的边

$T :=$ 添加 e 之后的 T

end (T 是 G 的最小生成树)

读者应当注意普林算法与克鲁斯卡尔算法的区别。在普林算法里，选择与已在树里的一个顶点相关联、并且不形成回路的权最小的边；相反地，在克鲁斯卡尔算法里，选择不一定与已在树里的一个顶点相关联、并且不形成回路的权最小的边。注意，在普林算法里，若没有对边排序，则在这个过程的某个阶段上，对添加的边来说就可能有多种选择。因此，为了让这个过程是确定的，就需要对边进行排序。下面的例子说明如何使用克鲁斯卡尔算法。

例3 用克鲁斯卡尔算法求图7-55所示的带权图的最小生成树。

解 在图7-57里显示这个最小生成树和在克鲁斯卡尔算法每个阶段上对边的选择。现在将证明普林算法产生连通带权图的最小生成树。

证 设 G 是一个连通带权图。假定普林算法相继地选择的边是 $e_1, e_2, \dots, e_{n-1}$ 。设 S 是以 $e_1, e_2, \dots, e_{n-1}$ 作为边的树，而设 S_k 是以 $e_1, e_2, \dots, e_k$ 作为边的树。设 T 是包含边 $e_1, e_2, \dots, e_k$ 的 G 的最小生成树，其中 k 是满足下列性质的最大整数：存在着包含普林算法所选择的前 k 条边的最小生成树。若证明了 $S=T$ ，则该定理得证。

假定 $S \neq T$ ，所以 $k < n-1$ 。因此， T 包含边 $e_1, e_2, \dots, e_k$ ，但是不包含 e_{k+1} 。考虑由 T 和 e_{k+1} 所组成的图。因为这个图是连通的并且有 n 条边，边过多不可能是树，所以它必然包含简单回路。这个简单回路必然包含 e_{k+1} ，因为在 T 里没有简单回路。另外，在这个简单回路里必然有不属于 S_{k+1} 的边，因为 S_{k+1} 是一棵树。通过从 e_{k+1} 的一个端点开始，该端点也是边 $e_1, e_2, \dots, e_k$ 之一的端点，并且遵循回路直到它到达一条不在 e_{k+1} 里的边为止，就可以找出一条不在 S_{k+1} 里的边 e ，它有一个端点也是边 $e_1, e_2, \dots, e_k$ 之一的端点。通过从 T 里删除 e 并且添加 e_{k+1} ，就获得带

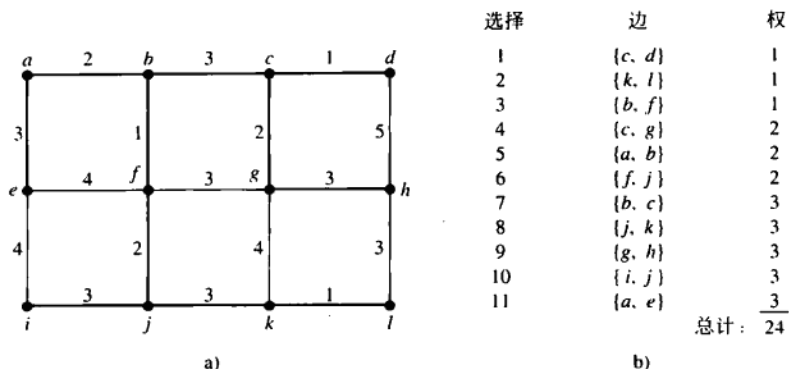


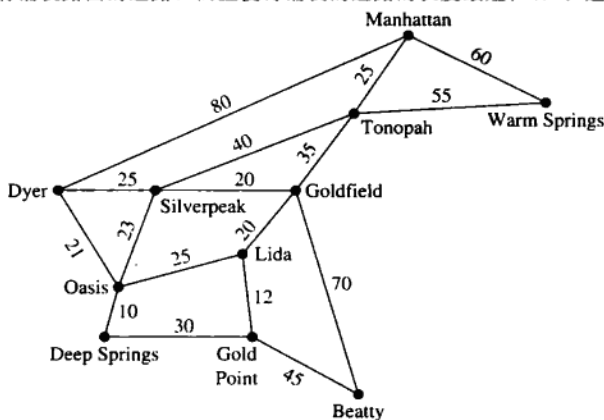
图 7-57 克鲁斯卡尔算法产生的最小生成树

$n-1$ 条边的树 T' (它是树, 因为它没有简单回路)。注意树 T' 包含 $e_1, e_2, \dots, e_{k+1}$ 。另外, 因为普林算法在第 $k+1$ 个步骤上选择 e_{k+1} , 并且在这个步骤上 e 也是可用的, 所以 e_{k+1} 的权就小于或等于 e 的权。根据这个观察结果就得出 T' 也是最小生成树, 因为它的边的权之和不超过 T 的边的权之和。这与对 k 的选择相矛盾, k 是使得包含 $e_1, e_2, \dots, e_k$ 的最小生成树存在的最大整数。因此, $k=n-1$ 并且 $S=T$ 。所以普林算法产生最小生成树。□

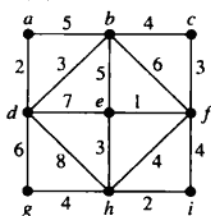
可以证明 (参见 [CoLeRiSt01]) 为了求出具有 e 条边和 v 个顶点的图的最小生成树, 可以用 $O(e \log e)$ 次运算来完成克鲁斯卡尔算法并且可以用 $O(e \log v)$ 次运算来完成普林算法。因此, 对于稀疏图来说使用克鲁斯卡尔算法更好。在稀疏图中, e 远远小于 $C(v, 2) = v(v-1)/2$, 即具有 n 个顶点的无向图的可能的总边数。否则, 这两个算法的复杂性没有什么差别。

练习

1. 下图所表示的道路都还没有铺设路面。边的权表示在成对的乡镇之间的道路长度。哪些道路应当铺设路面, 以便在每对乡镇之间都有铺设路面的道路, 而且使得铺设的道路的长度最短? (注: 这些乡镇都在内华达州。)



2. 用普林算法求下面所给的带权图的最小生成树。



3. 用克鲁斯卡尔算法设计在本节开头所描述的通信网络。
4. 用克鲁斯卡尔算法求练习2里带权图的最小生成树。
5. 找出具有多于一棵最小生成树的、带有最少可能边数的连通带权简单图。

连通带权无向图的最大生成树是带最大可能的权的生成树。

6. 设计与普林算法类似的、构造连通带权图的最大生成树的算法。
- *7. 设计求连通带权图里次最短生成树的算法。
8. 证明：若所有边的权都不相同，则连通带权图里有唯一的最小生成树。
9. 求图7-55的带权图里包含边 $\{e, i\}$ 和 $\{g, k\}$ 的总权最小的生成树。

索林(Sollin)算法从连通带权简单图 $G=(V, E)$ 这样产生最小生成树：相继地添加成组的边。假定对 V 里顶点进行了排序。这样就产生边的一个顺序，其中若 u_0 先于 u_1 ，或者若 $u_0 = u_1$ 并且 v_0 先于 v_1 ，则 $\{u_0, v_0\}$ 先于 $\{u_1, v_1\}$ 。这个算法首先同时选择每个顶点关联的权最小的边。在平局情形下选择在上述顺序里的第一条边。这样就产生出一个没有简单回路的图，即一些树组成的一个森林。其次，对森林里的每棵树，同时选择在该树里一个顶点与在不同的一棵树里顶点之间的最短的边。同样在平局情形下选择在上述顺序里的第一条边。(这样就产生出一个没有简单回路的图，它包含着比在这一步之前出现的更少的树。)继续进行同时地添加连接树的边的过程，直到已经选择了 $n-1$ 条边为止。在这个阶段上就已经构造了一棵最小生成树。

10. 用索林算法产生图7-53和图7-55所示的带权图的最小生成树。
- **11. 证明：索林算法产生连通无向带权图里的最小生成树。
- *12. 证明：若在索林算法的某个中间步骤里存在 r 棵树，则算法的下一次迭代至少添加 $\lceil n/2 \rceil$ 条边。
- *13. 证明：索林算法至少需要 $\log n$ 次迭代，以便从带有 n 个顶点的连通无向带权图产生一棵最小生成树。

关键术语和结果

术语

树：没有简单回路的连通无向图。

森林：没有简单回路的无向图。

根树：具有一个规定的顶点(称为根)，使得从这个根到任意其他顶点有唯一通路的有向图。

子树：树的子图，子图本身也是一棵树。

根树里 v 的父母：使得 (u, v) 是根树的一条边的顶点 u 。

根树里顶点 v 的子女：以 v 作为父母的任何顶点。

根树里顶点 v 的兄弟：与 v 具有相同父母的顶点。

根树里顶点 v 的祖先：在从根到 v 的通路上的任何顶点。

根树里顶点 v 的后代：以 v 作为祖先的任何顶点。

内点：具有子女的顶点。

树叶：没有子女的顶点。

顶点的层：从根到这个顶点的通路的长度。

树的高度：树里顶点的最大层数。

m 元树：每个内点都有不超过 m 个子女的树。

满 m 元树：每个内点都有恰好 m 个子女的树。

二叉树：满足 $m=2$ 的 m 元树(可以指定每个子女作为父母的左子女或右子女)。

有序树：在其中对每个内点的子女都线性地排序的树。

平衡树：在其中每个顶点都是在 h 层或 $h-1$ 层上的树，其中 h 是这棵树的高度。

二叉搜索树：这样的二叉树，在其中以项对顶点进行标记，使得一个顶点的标记大于这个顶点的左子树里所有顶点的标记，并且小于这个顶点的右子树里所有顶点的标记。

决策树：性质如下的根树，在其中每个顶点表示一次决策的可能输出，而树叶表示可能的解。

前缀码：一种编码，其中一个字符的编码永远不是另外一个字符的编码的前缀。

最小最大策略：第一个选手和第二个选手分别移动到具有最大值和最小值的顶点所表示的局面的

策略。

博弈树里顶点的值：对于树叶来说，就是当游戏在这个树叶所表示的局面里结束时第一个选手的得分。对于分别在偶数或奇数层上的内点来说，就是这个内点的子女的最大值或最小值。

树的遍历：树的顶点的列表。

前序遍历：通过规定列出根，接着列出第一棵子树，接着以从左到右的出现顺序列出其余子树，来递归地定义的有序根树的顶点列表。

中序遍历：通过规定列出第一棵子树，接着列出根，接着以从左到右的出现顺序列出其余子树，来递归地定义的有序根树的顶点列表。

后序遍历：通过规定以从左到右的出现顺序列出各子树，接着列出根，来递归地定义的有序根树的顶点列表。

中缀记法：从表示表达式(包括全套括号)的二叉树的中序遍历所获得的表达式形式。

前缀记法，或波兰记法：从表示表达式的二叉树的前序遍历所获得的表达式形式。

后缀记法，或逆波兰记法：从表示表达式的二叉树的后序遍历所获得的表达式形式。

生成树：包含图的所有顶点的树。

最小生成树：带最小可能的边的权之和的生成树。

贪心算法：在每个步骤上都做出最优选择的最优化算法。

结果

一个图是树，当且仅当在它的任何两个顶点之间都存在唯一简单通路。

带有 n 个顶点的树具有 $n-1$ 条边。

带有 i 个内点的满 m 元树具有 $mi+1$ 个顶点。

在满 m 元树的顶点数、树叶数和内点数之间的关系(见 7.1 节定理 4)。

在高度为 h 的满 m 元树里至多有 m^h 个树叶。

若 m 元树有 l 个树叶，则它的高度至少是 $\lceil \log_m l \rceil$ 。若这树也是满的和平衡的，则它的高度就是 $\lceil \log_m l \rceil$ 。

赫夫曼编码：给定一组符号的频率，为这些符号构造最优二元码的过程。

深度优先搜索，或回溯：构造生成树的过程，通过添加形成通路的边，直到不可能这样做为止，然后沿这条通路往回移动，直到找到可以形成新的通路的顶点为止。

宽度优先搜索：构造生成树的过程，通过相继添加与上次添加的边相关联的所有边，除非形成简单回路。

普林算法：产生带权图里最小生成树的过程，通过相继添加与已经在树里的顶点相关联的所有边中权最小的边，使得再添加边时不会产生简单回路。

克鲁斯卡尔算法：产生带权图里最小生成树的过程，通过相继添加还不在树里的权最小的边，使得再添加边时不会产生简单回路。

复习题

1. a) 定义树。 b) 定义森林。
2. 在树的顶点之间能否有两条不同的简单通路？
3. 至少给出三个例子说明如何在建模中使用树。
4. a) 定义根树和这样的树的根。
b) 定义根树里顶点的父母和顶点的子女。
c) 什么是根树里的内点、树叶和子树？
d) 画出至少带 10 个顶点的根树，其中每个顶点的度都不超过 3。指出树根、每个顶点的父母、每个顶点的子女、内点和树叶。
5. a) 带 n 个顶点的树有多少条边？
b) 为确定带有 n 个顶点的森林里的边数，你需要知道什么值？

6. a) 定义正则 m 元树。
b) 若正则 m 元树有 i 个内点, 则它有多少个顶点? 此树有多少个树叶?
7. a) 什么是根树的高度?
b) 什么是平衡树?
c) 高度为 h 的 m 元树可以有多少个树叶?
8. a) 什么是二叉搜索树?
b) 描述构造二叉搜索树的算法。
c) 构造单词 *vireo*, *warbler*, *egret*, *grosbeak*, *nuthatch* 和 *kingfisher* 的二叉搜索树。
9. a) 什么是前缀码?
b) 二叉树如何表示前缀码?
10. a) 定义前序遍历、中序遍历和后序遍历。
b) 给出至少带 12 个顶点的二叉树的前序遍历、中序遍历和后序遍历的实例。
11. a) 解释一下如何用前序遍历、中序遍历和后序遍历来求算术表达式的前缀形式、中缀形式和后缀形式。
b) 画出表示 $((x-3)+((x/4)+(x-y)\uparrow 3))$ 的有序根树。
c) 求在 (b) 里的表达式的前缀和后缀形式。
12. 证明: 排序算法所使用的比较次数至少是 $\lceil \log n! \rceil$ 。
13. a) 描述一下赫夫曼编码算法, 这个算法求一组给定频率的符号的最优编码。
b) 用赫夫曼编码求下列符号和频率的最优编码: A: 0.2, B: 0.1, C: 0.3, C: 0.4。
14. 画出 *nim* 游戏的博弈树, 假设初始局面由两堆石头组成, 分别含有 1 块和 4 块石头。假如两个选手都遵循最优策略的话, 谁将赢得游戏呢?
15. a) 什么是简单图的生成树?
b) 哪些简单图具有生成树?
c) 描述需要求出简单图的生成树的应用, 至少举两个不同的应用。
16. a) 描述求简单图里生成树的两个不同算法。
b) 用你所选择的至少带 8 个顶点和 15 条边的图, 来解释你在 (a) 里所描述的两个算法是如何求简单图的生成树的。
17. a) 解释如何用回溯来确定能否用 n 种颜色来着色简单图。
b) 用例子说明如何用回溯来证明: 色数等于 4 的图不能用 3 种颜色来着色, 但是可以用 4 种颜色来着色。
18. a) 什么是连通带权图的最小生成树?
b) 至少描述出两个不同的、需要求出连通带权图的最小生成树的应用。
19. a) 描述求最小生成树的普林算法和克鲁斯卡尔算法。
b) 用至少带 8 个顶点和 15 条边的图, 来解释克鲁斯卡尔算法和普林算法是如何求最小生成树的。

补充练习

- *1. 证明: 简单图是树当且仅当它不包含简单回路, 并且添加连接两个不相邻顶点的一条边, 就产生恰好有两条回路的新图(其中不认为包含相同的边的回路是不同的)。
- *2. 有多少种非同构的带 6 个顶点的根树?
3. 证明: 每一个至少有一条边的树都至少有两个悬挂点。
4. 证明: 有 $n-1$ 个悬挂点的、带有 n 个顶点的树必然同构于 $K_{1,n-1}$ 。
5. 带有 n 个顶点的树的顶点的度之和是什么?
- *6. 假定 $d_1, d_2, \dots, d_n$ 是和为 $2n-2$ 的 n 个正整数。证明: 存在一个带有 n 个顶点的树, 使得这些顶点的度为 $d_1, d_2, \dots, d_n$ 。
7. 证明: 每个树都是可平面图。
8. 证明: 每个树都是偶图。
9. 证明: 每个森林都可以用两种颜色来着色。

例 10.10 k 度 B 树是一个根树, 它的所有树叶都是同一层上, 它的根具有至少两个和至多 k 个子女, 除非根就是树叶, 并且除根外的每个内点有至少 $\lceil k/2 \rceil$ 个、但不超过 k 个子女。当用 B 树来表示计算机文件时, 就可以有效地访问这些文件。

10. 画出三种不同的高度为4的3度B树。
 *11. 给出高度为 h 的 k 度B树里树叶数的上界和下界。
 *12. 给出有 n 个树叶的 k 度B树的高度的上界和下界。

二项式树 $B_k (i=0, 1, 2, \dots)$ 是如下递归定义的有序根树。

基础步骤：二项式树 B_0 是具有单个顶点的树。

递归步骤：设 k 是非负整数。为了构造二项式树 B_{k+1} ，把 B_k 的一个副本加入 B_k 的第二个副本，方法是加入一条边，这条边让 B_k 的第一个副本的根成为 B_k 的第二个副本的最左子女。

13. 对 $k=0, 1, 2, 3, 4$ ，画出 B_k 。
 14. B_k 有多少个顶点？证明你的答案是正确的。
 15. 求 B_k 的高度。证明你的答案是正确的。
 16. B_k 在深度 j 有多少个顶点？其中 $0 \leq j < k$ 。证明你的答案是正确的。
 17. B_k 的根的度数是多少？证明你的答案是正确的。
 18. 证明： B_k 中度数最大的顶点是根。

若根树 T 满足下面的递归定义，则称它为 S_k 树。若它只有一个顶点，则它是 S_0 树。对 $k > 0$ 来说，若通过把一个 S_{k-1} 树的根作为一个新树的根，把另外一个 S_{k-1} 树的根作为新树的根的子树，从两个 S_{k-1} 树来建立一个新树，则这个新树是 S_k 树。

19. 对 $k=0, 1, 2, 3, 4$ ，画出一个 S_k 树。
 20. 证明： S_k 树有 2^k 个顶点并且在 k 层上有唯一一个顶点。在 k 层上的这个顶点称为把柄。
 *21. 假定 T 是带有把柄 v 的 S_k 树。证明： T 可以从不相交的树 $T_0, T_1, \dots, T_{k-1}$ 来这样获得，其中 v 不在这些树的任何一个里，对 $i=0, 1, \dots, k-1$ 来说， T_i 是 S_i 树；对 $i=0, 1, \dots, k-2$ ，连接 v 到 v_i 并且连接 v_i 到 v_{i+1} 。

有序根树在层顺序下的顶点列表从根开始，接着是从左到右在1层上的顶点，从左到右在2层上的顶点，依此类推。

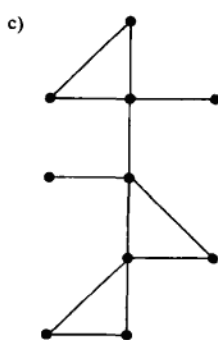
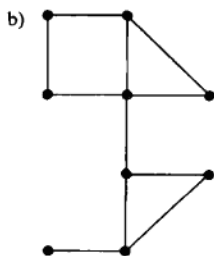
22. 列出7.3节图7-27和图7-33里的有序根树在层顺序下的顶点列表。
 23. 设计列出有序根树在层顺序下的顶点列表的算法。
 *24. 设计一种算法，确定一组通用地址能否成为根树的树叶地址。
 25. 设计从树叶的通用地址来构造根树的算法。

图的割集是这样一些边的集合，使得删除这些边就产生一个子图，这个子图的连通分支比原来的图要多，但是这些边的任何真子集都没有这个性质。

26. 证明：图的割集必然与这个图的任何生成树都有至少一条公共边。

仙人掌图是连通图，其中没有边是在多于一条的简单回路上，这些简单回路不经过除了起点以外的任何顶点超过一次，而且不在除了终点以外的其他地方经过起点（其中不认为由相同的边所组成的两个回路是不同的）。

27. 下面的图哪些是仙人掌图？

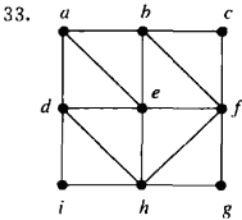
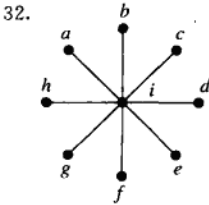
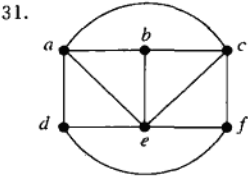


28. 树是否必然是仙人掌图？

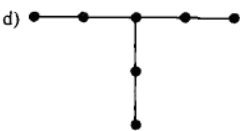
29. 证明：若在树里添加一条回路，它包含一些起止于树里顶点上的新边，则形成一个仙人掌图。
- *30. 证明：若在连通图里，每条不经过任何顶点(除起点以外)超过一次的回路都包含奇数条边，则这个图必然是仙人掌图。

简单图 G 的限制度数生成树具有下面性质，在这个树里顶点的度不能超过某个规定的界限。在运输系统的模型里，在交叉路口处的道路数目是有限的，在通信网络的模型里，进入一个结点的连接数目是有限的，这种模型都可以使用限制度数生成树。

在练习 31~33 里，求所给的图的限制度数生成树，其中每个顶点的度都小于或等于 3，或者证明不存在这样的生成树。

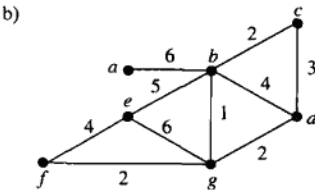
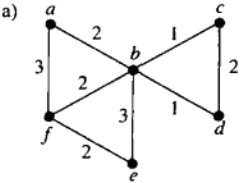


34. 证明：简单图的限制度数生成树(其中每个顶点的度都不超过 2)是由该图里的单独一条哈密顿回路所组成的。
35. 若可以用整数 $1, 2, \dots, n$ 来标记带有 n 个顶点的树的顶点，使得相邻顶点的标记之差的绝对值全都是不同的，则称这棵树为优美的。证明：下面的树都是优美的。



毛虫图是含有一条简单通路的树，使得不包含在这条通路里的每个顶点都与这条通路里的一个顶点相邻。

36. 练习 35 里的图，哪些是毛虫图？
37. 带 6 个顶点的互不同构的毛虫图有多少种？
- **38. a) 证明或反驳：其边形成一条简单通路的所有树都是优美的。
- b) 证明或反驳：所有的毛虫图都是优美的。
39. 假设在一个很长的位串中，位 0 出现的频率是 0.9，位 1 出现的频率是 0.1，并且位都是独立地出现的。
- a) 为 4 个两位的块 00、01、10 和 11 构造赫夫曼编码。用这个编码来编码一个位串，所需要的平均位数是多少？
- b) 为 8 个三位的块构造赫夫曼编码。用这个编码来编码一个位串，所需要的平均位数是多少？
40. 假设 G 是没有回路的有向图。描述如何用深度优先搜索来完成 G 的顶点的拓扑排序。
- *41. 假定 e 是带权图里与顶点 v 关联的一条边，使得 e 的权不超过与顶点 v 关联的任何其他边的权。证明：存在一棵包含这条边的最小生成树。
42. 三对夫妇到达一条河流的岸边。每个妻子都容易嫉妒，当她的丈夫与其他的妻子(或其他人)在一起、但是她不在场时，她就不信任她的丈夫。六个人如何用一条只能装载不超过两个人的船来渡河，使得每位丈夫无法与妻子之外的女人单独相处？解答时使用图论模型。
- *43. 证明：若在带权图里没有两条边具有相同的权，则在每个最小生成树里都包含着与顶点 v 关联的权最小的边。
44. 求下面两个图的最小生成树，其中在生成树里每个顶点的度都不超过 2。



计算机题目

按下述给定的输入与输出写出程序。

1. 给定无向简单图的相邻矩阵，确定这个图是不是树。
2. 给定根树的相邻矩阵和这棵树里的一个顶点，求出这个顶点的父母、子女、祖先、后代和层数。
3. 给定根树的边的列表和这棵树里的一个顶点，求出这个顶点的父母、子女、祖先、后代和层数。
4. 给定项的列表，构造包含这些项的二叉搜索树。
5. 给定二叉搜索树和一个项，在这个二叉搜索树里求出这个项的位置或添加这个项。
6. 给定有序根树的边的有序列表，求出它的边的通用地址。
7. 给定有序根树的边的有序列表，以前序、中序和后序列出它的顶点。
8. 给定前缀形式的算术表达式，求它的值。
9. 给定后缀形式的算术表达式，求它的值。
10. 给定一组符号的频率，用赫夫曼编码来求这些符号的最优编码。
11. 给定轮流取石头游戏的开局，确定第一个选手的最优策略。
12. 给定连通无向简单图的相邻矩阵，用深度优先搜索找出这个图的生成树。
13. 给定连通无向简单图的相邻矩阵，用宽度优先搜索找出这个图的生成树。
14. 给定一组正整数和一个正整数 N ，利用回溯求这些整数的和为 N 的子集合。
- *15. 给定无向简单图的相邻矩阵，若有可能，则利用回溯来用三种颜色为这个图着色。
- *16. 给定一个正整数 n ，利用回溯来解决 n 皇后问题。
17. 给定带权无向连通图的边的列表和它们的权，用普林算法求这个图的最小生成树。
18. 给定带权无向连通图的边的列表和它们的权，用克鲁斯卡尔算法求这个图的最小生成树。

计算和研究

利用计算程序或你所编写的程序来做下面的练习。

1. 显示所有的带有 6 个顶点的树。
2. 显示全部的互不同构的带有 7 个顶点的树。
- *3. 根据 ASC II 码字符在典型输入中出现的频率，构造它们的赫夫曼编码。
4. 对 $n=1, 2, 3, 4, 5, 6$ ，计算 K_n 的不同的生成树的个数。猜想它们的公式。
5. 对于 $n=100$ ， $n=1000$ 和 $n=10\,000$ 来说，比较一下排序 n 个元素的表所需要的比较次数，其中元素都是随机选择的正整数，使用选择排序、插入排序、归并排序和快速排序。
6. 对于不超过 10 的所有正整数 n 来说，计算一下在 $n \times n$ 棋盘上放置 n 个皇后，使得这些皇后不能互相攻击的不同方式数。
- *7. 求把美国 50 个州的首府互相连接起来的图的最小生成树，其中每条边上的权是首府之间的距离。
8. 画出 4×4 棋盘上跳棋游戏的完全博弈树。

写作题目

利用本书以外的资料，就下列问题写出短文。

1. 解释凯莱如何用树来枚举特定类型的碳水化合物化合物的个数。
2. 定义 AVL 树(有时也称为均衡树)。解释如何以及为什么在许多不同的算法中都用到 AVL 树。
3. 定义二叉树并解释如何用二叉树来表示图像。描述如何通过操纵对应的二叉树来旋转、缩放和转换图像。
4. 定义一个堆，并解释如何把树转化成堆。堆为什么在排序中 useful?
5. 描述一下针对连续读入字符时字母频率发生变化的数据压缩的动态规划算法，比如自适应赫夫曼编码。
6. 解释如何用 $\alpha-\beta$ 剪枝来简化对博弈树的值的计算。
7. 描述下棋程序(比如深蓝)所使用的技术。
8. 为树网这种类型的图下定义。解释一下这种图如何用在非常大的系统集成和并行计算中。
9. 讨论在 IP 组播中避免路由器之间回路所用的算法。

10. 描述基于深度优先搜索来求图的节点的算法。
11. 描述基于深度优先搜索来求有向图的强连通分支的算法。
12. 描述在 Web 上不同搜索引擎的网络爬虫和网络蜘蛛所用的搜索技术。
13. 描述一下求图的最小生成树的算法,使得生成树上任意顶点的最大度数不超过一个固定的常数 k 。
14. 就复杂度和应用场合而言,对一些最重要的排序算法进行比较。
15. 讨论一下构造最小生成树的算法的历史和起源。
16. 描述产生随机树的算法。