离 散 数 学

--高级计数

2025年9月28日星期日

章节目录

1. 递归关系的应用

小节目录1

1.1 递推关系的应用

- 斐波那契数列
- 汉诺塔
- 计数问题

1.2 算法与递归关系（目前未包含在幻灯片中）

递推关系

定义：数列 $\{a_n\}$ 的递推关系是一个方程，它将 a_n 表示为该数列之前的一个或多个项的函数，即 , a_0, a_1, …, a_{n-1}, 其中 n 为所有满足 $n \geq n_0$的整数， n_0 是一个非负整数

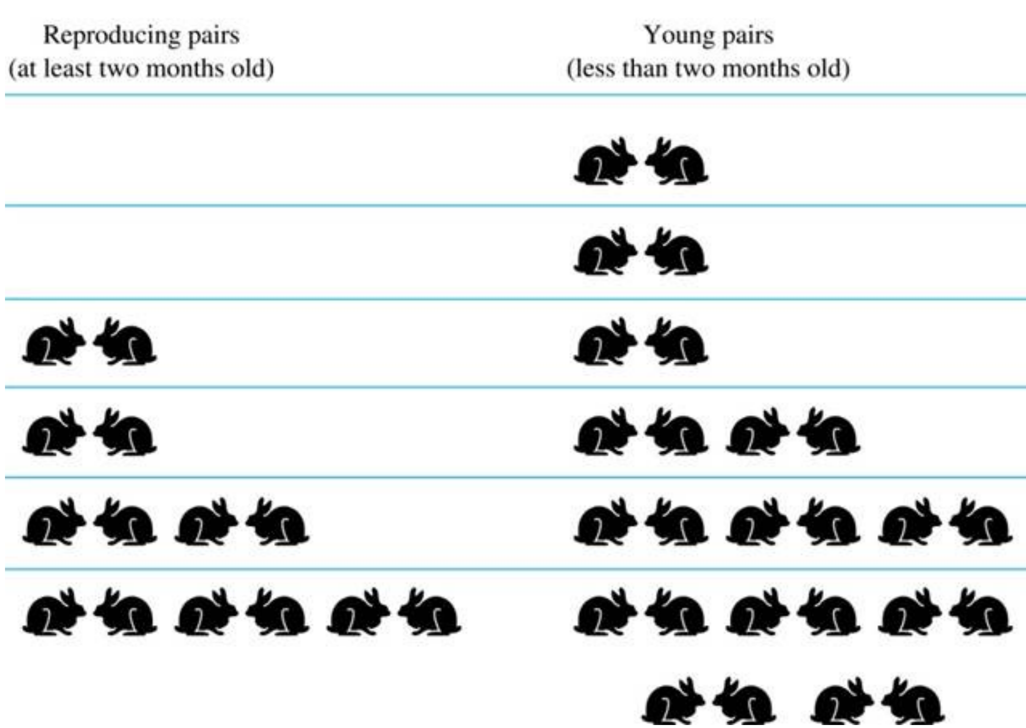
- 如果一个数列的各项满足递推关系，则称该数列为递推关系的解
- 数列的初始条件指定了递推关系生效前的各项

兔子与斐波那契数列[1]

例子：一对年轻的兔子（一公一母）被放置在一个岛上。兔子在满两个月前不会繁殖。两个月大后，每对兔子每个月都会产下一对兔子。假设兔子永远不会死亡，求在经过 n 个月后岛上兔子的对数的递归关系

这是十三世纪时莱昂纳多·皮萨诺（斐波那契）考虑的原始问题

兔子与斐波那契数列2



Month	Reproducing pairs	Young pairs	Total pairs
1	0	1	1
2	0	1	1
3	1	1	2
4	1	2	3
5	2	3	5
6	3	5	8

模拟岛上兔子种群的增长

Jump to long description

兔子与斐波那契数列3

解答：设 f_n 为 n 个月后岛上兔子的对数

- 在第一个月末，岛上有 $f_1 = 1$ 对兔子.
- 同样， $f_2 = 1$，因为这对兔子在第一个月内不会繁殖.
- 要找到 n 个月后岛上的兔子对数，将前一个月岛上的兔子对数 f_{n-1} 与新生兔子对数相加，新生兔子对数等于 f_{n-2}，因为每对新生兔子都来自至少两个月大的兔子对

因此，数列 $\{f_n\}$ 满足递归关系 $f_n = f_{n-1} + f_{n-2}$， 适用于 $n \geq 3$，初始条件为 $f_1 = 1$ 和 $f_2 = 1$。n 个月后岛上的兔子对数由第 n 个斐波那契数给出

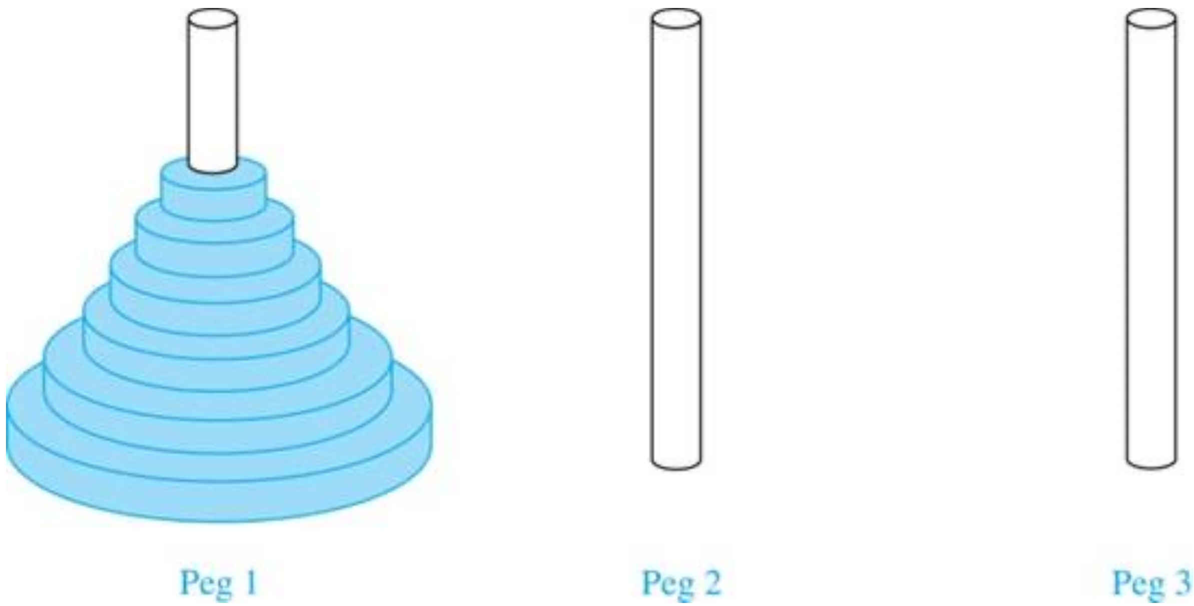
汉诺塔[1]

在十九世纪末，法国数学家爱德华·卢卡斯发明了一个谜题，包含三个柱子和不同大小的圆盘。最初，所有圆盘都按大小顺序放在第一个柱子上，最大的在底部

规则：你可以一次移动一个圆盘，从一个柱子移动到另一个柱子，只要大的圆盘永远不会放在小的圆盘上面

目标：通过允许的移动，将所有圆盘最终放在第二个柱子上，按大小顺序排列，最大的在底部

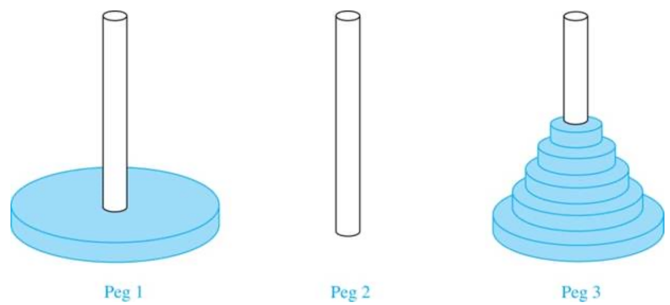
汉诺塔2



汉诺塔问题的最初放置

汉诺塔3

解答：设 $\{H_n\}$ 表示解汉诺塔谜题所需的移动次数，其中有 n 个圆盘。为序列 $\{H_n\}$ 设置递归关系。初始时，n 个圆盘在柱子 1 上。我们可以将顶部的 n–1 个圆盘按照谜题规则移动到柱子 3，这需要 H_{n-1} 次移动



首先，我们使用 1 次移动将最大的圆盘转移到第二个柱子。然后，我们将 n–1 个圆盘从柱子 3 移到柱子 2，需要额外的 H_{n-1} 次移动。这个过程不能用更少的步骤完成。因此,

$$H_n = 2H_{n-1} + 1.$$

初始条件是 $H_1 = 1$ ，因为一个圆盘可以在一次移动中从柱子 1 移到柱子 2.

汉诺塔4

我们可以使用迭代方法来求解这个递归关系，通过反复将 H_n 表示为序列之前项的函数.

$$
\begin{aligned}
H_n = 2H_{n-1} + 1 & \\
&= 2(2H_{n-2} + 1) + 1 = 2^2 H_{n-2} + 2 + 1 \\
&= 2^2(2H_{n-3} + 1) + 2 + 1 = 2^3 H_{n-3} + 2^2 + 2 + 1 \\
&\vdots \\
&= 2^{n-1}H_1 + 2^{n-2} + 2^{n-3} + \ldots + 2 + 1 \\
&= 2^{n-1} + 2^{n-2} + 2^{n-3} + \ldots + 2 + 1 \qquad \text{because } H_1 = 1 \\
&= 2^n - 1
\end{aligned}
$$

- 这个谜题还有一个神话。传说在河内的一个塔楼中，僧侣们按照谜题的规则将 64 个金盘从一个柱子转移到另一个柱子。他们每天移动一个盘子。当谜题完成时，世界将会结束.
- 根据这个神话中的 64 个金盘公式，
  $2^{64} - 1$ = 18,446, 744,073, 709,551,615天需要来解决谜题，这比 5000 亿年还要长
- 雷夫谜题（由亨利·杜德尼于 1907 年提出）类似，但有 4 个柱子。这个谜题的最小移动次数的著名猜想仍未解决

计数位串[1]

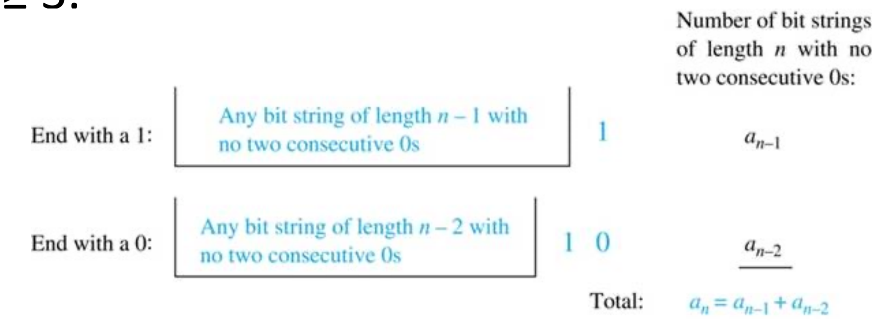
例子 3：找出长度为 n 的不包含两个连续 0 的位串的递归关系，并给出初始条件。长度为五的位串有多少个?

解答：设 a_n 表示长度为 n 的不包含两个连续 0 的位串的数量。为了得到 $\{a_n\}$ 的递归关系，需要注意长度为 n 的不包含两个连续 0 的位串的数量等于以 0 结尾的位串的数量加上以 1 结尾的位串的数量.

现在假设$n \geq 3$.

- 长度为 n 且以 1 结尾的不包含两个连续 0 的位串，就是长度为 n−1 的不包含两个连续 0 的位串的尾部加上一个 1。因此，这样的位串有 a_{n-1} 个
- 长度为 n 且以 0 结尾的不包含两个连续 0 的位串，就是长度为 n−2 的不包含两个连续 0 的位串加上 10。因此，这样的位串有 a_{n-2} 个

因此，我们得出 $a_n = a_{n-1} + a_{n-2}$ 对于 $n \geq 3$.



Jump to long description

位串$_2$

初始条件为:

- $a_1 = 2$, 因为位串 0 和 1 都没有连续的 0
- $a_2 = 3$, 因为位串 01、10 和 11 都没有连续的 0，而 00 有

为了求得 a_5，我们使用递归关系三次来计算，得到:

- $a_3 = a_2 + a_1 = 3 + 2 = 5$
- $a_4 = a_3 + a_2 = 5 + 3 = 8$
- $a_5 = a_4 + a_3 = 8 + 5 = 13$

注意到 $\{a_n\}$ 满足与斐波那契数列满足相同的递推关系。由于$a_1 = f_3$ 和 $a_2 = f_4$,我们得出$a_n = f_{n+2}$.

计算对一个乘积进行括号化的方法数

例子：找出 C_n 的递归关系，其中 C_n 表示对 n + 1 个数 $x_0 \cdot x_1 \cdot x_2 \cdot \cdots \cdot x_n$ 进行括号化的方法数，以指定乘法的顺序。例如， $C_3 = 5$，因为所有可能的括号化 4 个数的方法如下

$((x_0 \cdot x_1) \cdot x_2) \cdot x_3$， $(x_0 \cdot (x_1 \cdot x_2)) \cdot x_3$， $(x_0 \cdot x_1) \cdot (x_2 \cdot x_3)$，$x_0 \cdot ((x_1 \cdot x_2) \cdot x_3)$， $x_0 \cdot (x_1 \cdot (x_2 \cdot x_3))$

解答：无论如何在 $x_0 \cdot x_1 \cdot x_2 \cdot \cdots \cdot x_n$ 中插入括号，总会有一个“·”运算符位于所有括号之外。这个最后的运算符位于两个 n + 1 个数之间，比如 x_k 和 x_{k+1} 。由于在乘积 $x_0 \cdot x_1 \cdot x_2 \cdot \cdots \cdot x_k$ 中插入括号的方法有 C_k 种，在乘积 $x_{k+1} \cdot x_{k+2} \cdot \cdots \cdot x_n$中插入括号的方法有 C_{n-k-1} 种，因此我们有，

$$\begin{aligned} C_n &= C_0 C_{n-1} + C_1 C_{n-2} + \cdots + C_{n-2} C_1 + C_{n-1} C_0 \\ &= \sum_{k=0}^{n-1} C_k C_{n-k-1} \end{aligned}$$

初始条件是 $C_0 = 1$ 和 $C_1 = 1$.

序列 $\{C_n\}$ 是**卡塔兰数**序列。这个递归关系可以通过生成函数法来求解

用生成函数求解递推方程$_1$

Solve the recurrence relation $a_k = 3a_{k-1}$ for $k = 1, 2, 3, \ldots$ and initial condition $a_0 = 2$.

Solution: Let $G(x)$ be the generating function for the sequence $\{a_k\}$, that is, $G(x) = \sum_{k=0}^{\infty} a_k x^k$. First note that

$$xG(x) = \sum_{k=0}^{\infty} a_k x^{k+1} = \sum_{k=1}^{\infty} a_{k-1} x^k.$$

Using the recurrence relation, we see that

$$\begin{aligned} G(x) - 3xG(x) &= \sum_{k=0}^{\infty} a_k x^k - 3\sum_{k=1}^{\infty} a_{k-1} x^k \\ &= a_0 + \sum_{k=1}^{\infty} (a_k - 3a_{k-1}) x^k \\ &= 2, \end{aligned}$$

because $a_0 = 2$ and $a_k = 3a_{k-1}$. Thus,

$$G(x) - 3xG(x) = (1 - 3x)G(x) = 2.$$

Solving for $G(x)$ shows that $G(x) = 2/(1 - 3x)$. Using the identity $1/(1 - ax) = \sum_{k=0}^{\infty} a^k x^k$, from Table 1, we have

$$G(x) = 2\sum_{k=0}^{\infty} 3^k x^k = \sum_{k=0}^{\infty} 2 \cdot 3^k x^k.$$

Consequently, $a_k = 2 \cdot 3^k$.

用生成函数求解递推方程$_2$

$a_n - 5a_{n-1} + 6a_{n-2} = 0$， $a_0 = 1, a_1 = -2$

$$
\begin{array}{rl}
G(x) = & a_0 + a_1x + a_2x^2 + a_3x^3 + \ldots \\
-5x\,G(x) = & \quad -5a_0x - 5a_1x^2 - 5a_2x^3 - \ldots \\
6x^2\,G(x) = & \quad\quad +6a_0x^2 + 6a_1x^3 + \ldots \\
\hline
\end{array}
$$

$$(1-5x+6x^2)G(x) = a_0 + (a_1-5a_0)x$$

$$G(x) = \frac{1-7x}{1-5x+6x^2} = \frac{5}{1-2x} - \frac{4}{1-3x}$$

$$= 5\sum_{n=0}^{\infty} 2^n x^n - 4\sum_{n=0}^{\infty} 3^n x^n$$

$$a_n = 5 \cdot 2^n - 4 \cdot 3^n$$

用生成函数求解递推方程2

$a_n = 8a_{n-1} + 10^{n-1}$ 其中$a_0=1,a_1=9$

We multiply both sides of the recurrence relation by x^n to obtain

$a_n x^n = 8a_{n-1}x^n + 10^{n-1}x^n$.

Let $G(x) = \sum_{n=0}^{\infty} a_n x^n$ be the generating function

$$\begin{aligned} G(x) - 1 &= \sum_{n=1}^{\infty} a_n x^n = \sum_{n=1}^{\infty} (8a_{n-1}x^n + 10^{n-1}x^n) \\ &= 8\sum_{n=1}^{\infty} a_{n-1}x^n + \sum_{n=1}^{\infty} 10^{n-1}x^n \\ &= 8x\sum_{n=1}^{\infty} a_{n-1}x^{n-1} + x\sum_{n=1}^{\infty} 10^{n-1}x^{n-1} \\ &= 8x\sum_{n=0}^{\infty} a_n x^n + x\sum_{n=0}^{\infty} 10^n x^n \\ &= 8xG(x) + x/(1 - 10x), \end{aligned}$$

Solving for $G(x)$ shows that

$$G(x) = \frac{1 - 9x}{(1 - 8x)(1 - 10x)}.$$

Expanding the right-hand side of this equation into partial fractions (as is done in the integration of rational functions studied in calculus) gives

$$G(x) = \frac{1}{2}\left(\frac{1}{1 - 8x} + \frac{1}{1 - 10x}\right).$$

Using Example 5 twice (once with $a = 8$ and once with $a = 10$) gives

$$\begin{aligned} G(x) &= \frac{1}{2}\left(\sum_{n=0}^{\infty} 8^n x^n + \sum_{n=0}^{\infty} 10^n x^n\right) \\ &= \sum_{n=0}^{\infty} \frac{1}{2}(8^n + 10^n)x^n. \end{aligned}$$

Consequently, we have shown that

$$a_n = \frac{1}{2}(8^n + 10^n).$$

2. 求解线性递归关系

小节目录

2.1 线性齐次递归关系

2.2 带有常系数的线性齐次递归关系的求解

2.3 带有常系数的线性非齐次递归关系的求解

线性齐次递推关系

定义：一个具有常系数的 k 阶线性齐次递推关系的形式为$a_n = c_1a_{n-1} + c_2a_{n-2} + \dots\dots + c_k\, a_{n-k}$,其中 $c_1, c_2, \dots\dots, c_k$为实数，且 $c_k \neq 0$

- 它是线性的，因为右边是序列前项的倍数之和
- 它是齐次的，因为所出现的各项都是a_i的倍数。序列各项的系数都是常数而不是依赖于n的函数
- 其阶数为 k，因为 a_n 是由该序列的前 k 项表达的

通过强归纳法，满足这种递推关系的序列由递推关系和 k 个初始条件$a_0 = C_0, a_1 = C_1 ,\dots , a_{k-1} = C_{k-1}$唯一确定

线性齐次递推关系的例子

$P_n = (1.11)P_{n-1}$ 一阶线性齐次递归关系的形式

$f_n = f_{n-1} + f_{n-2}$ 二阶线性齐次递归关系的形式

$a_n = a_{n-1} + a_{n-2}^2$ 非线性

$H_n = 2H_{n-1} + 1$ 非齐次

$B_n = nB_{n-1}$ 系数不是常数

求解线性齐次递推关系

基本方法是寻找形式为 $a_n = r^n$的解，其中 r 是常数.

注意， $a_n = r^n$是递推关系 $a_n = c_1a_{n-1} + c_2a_{n-2} + \cdots + c_k a_{n-k}$ 的解，当且仅当$r^n = c_1r^{n-1} + c_2r^{n-2} + \cdots + c_k r^{n-k}$

代数操作得到该递推关系的特征方程:

$$r^k - c_1r^{k-1} - c_2r^{k-2} - \cdots - c_{k-1}r - c_k = 0$$

序列 $\{a_n\}$以 $a_n = r^n$作为解，当且仅当 r是特征方程的解

特征方程的解称为递推关系的特征根。根用于给出递推关系所有解的显式公式

求解二阶线性齐次递推关系

定理 1：设 c_1 和 c_2为实数。假设方程 $r^2 - c_1 r - c_2 = 0$有两个不同的根 r_1 和 r_2。那么数列 $\{a_n\}$是递推关系 $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ 的解，当且仅当 $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$, $n = 0,1,2,\ldots$, 其中 α_1 和 α_2 是常数

使用定理1

例子：递推关系 $a_n = a_{n-1} + 2a_{n-2}$解是什么？初始条件为 $a_0 = 2$ 和 $a_1 = 7$ ？

解：特征方程是$r^2 - r - 2 = 0$.

它的根是 $r=2$ 和 $r=-1$。因此，数列 $\{a_n\}$是递归关系的解，当且仅当

$a_n = \alpha_1 2^n + \alpha_2(-1)^n$,其中 α_1 和 α_2 是常数.

要找到常数 α_1和 α_2 ，注意到

$a_0 = 2 = \alpha_1 + \alpha_2$ 和 $a_1 = 7 = \alpha_1 2 + \alpha_2(-1)$

解这些方程，我们得到 $\alpha_1 = 3$ 和 $\alpha_2 = -1$

因此，解是序列 $\{a_n\}$，其中$a_n = 3\cdot 2^n - (-1)^n$

斐波那契数列的显式公式[1]

我们可以使用定理 1 来找到斐波那契数列的显式公式。斐波那契数列满足递归关系$f_n = f_{n-1} + f_{n-2}$和初始条件: $f_0 = 0$ 和 $f_1 = 1$.

解：特征方程
$r^2 - r - 1 = 0$的根是

$$r_1 = \frac{1+\sqrt{5}}{2}$$

$$r_2 = \frac{1-\sqrt{5}}{2}$$

斐波那契数列2

因此根据定理1

$$f_n = \alpha_1\left(\frac{1+\sqrt{5}}{2}\right)^n + \alpha_2\left(\frac{1-\sqrt{5}}{2}\right)^n$$

α_1 和 α_2是常数。

使用初始条件$f_0 = 0$ 和 $f_1 = 1$, 我们有

$$f_0 = \alpha_1 + \alpha_2 = 0$$

$$f_1 = \alpha_1\left(\frac{1+\sqrt{5}}{2}\right) + \alpha_2\left(\frac{1-\sqrt{5}}{2}\right) = 1.$$

求解，我们得到 $\alpha_1 = \frac{1}{\sqrt{5}}, \qquad \alpha_2 - \frac{1}{\sqrt{5}}.$

因此, $f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$

当存在重复特征根时的解法

定理 2： 设 c_1 和 c_2 为实数且 $c_2 \neq 0$。假设方程 $r^2 - c_1 r - c_2 = 0$ 只有一个根 r_0。那么数列 $\{a_n\}$ 是递归关系 $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ 的解，当且仅当

$$a_n = \alpha_1 r_0^n + \alpha_2 n r_0^n$$

对于 $n = 0,1,2,\ldots$，其中 α_1 和 α_2 是常数.

使用定理二

例子：递归关系 $a_n = 6a_{n-1} - 9a_{n-2}$的解是什么？初始条件为 $a_0 =$ 1 和 $a_1 = 6$?

解：特征方程是$r^2 - 6r + 9 = 0.$

唯一的根是 $r = 3$。因此，数列$\{a_n\}$是递推关系的解，当且仅当

$$a_n = \alpha_1 3^n + \alpha_2 n(3)^n$$

其中 α_1 和 α_2是常数

要找到常数α_1 和 α_2, 注意到

$$a_0 = 1 = \alpha_1 \quad 和 \quad a_1 = 6 = \alpha_1 \cdot 3 + \alpha_2 \cdot 3$$

解这些方程，我们得到 $\alpha_1 = 1$ 和 $\alpha_2 = 1$

因此,

$$a_n = 3^n + n3^n$$

求解任意阶线性齐次递归关系

这个定理可以用于求解具有常系数的任意阶线性齐次递归关系，当特征方程具有不同的根时

定理 3： 设 $c_1, c_2, \ldots, c_k$为实数。假设特征方程

$r^k - c_1 r^{k-1} - \cdots - c_k = 0$

有 k 个不同的根 $r_1, r_2, \ldots, r_k$。那么数列$\{a_n\}$是递归关系的解

$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots.. + c_k a_{n-k}$

当且仅当

$$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n + \cdots + \alpha_k r_k^n$$

对于 $n = 0, 1, 2, \ldots$, 其中 $\alpha_1, \alpha_2, \ldots, \alpha_k$ 是常数.

求解任意阶线性齐次递归关系

例子：求出具有初始条件a_0=2，a_1=5和a_2=15的递推关系$a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$的解

解：特征方程是$r^3 - 6r^2 + 11r - 6 = (r-1)(r-2)(r-3) = 0$

3个特征根分别是r=1, r=2, r=3, 因此$a_n = \alpha_1 1^n + \alpha_2 2^n + \alpha_3 3^n$

α_1=1, α_2= −1, α_3=2

$a_n = 1^n - 2^n + 2 \cdot 3^n$

求解任意阶线性齐次递归关系

定理 4：设 $c_1, c_2, \ldots,$ 为实数。假设特征方程

$$r^k - c_1 r^{k-1} - \cdots - c_k = 0$$

有 t 个不同的根 $r_1, r_2, \ldots, r_t$ 它们的重数分别为 $m_1, m_2, \ldots, m_t$, 其中 $m_i \geq 1$ 对于 $i = 0, 1, 2, \ldots, t$ 且 $m_1 + m_2 + \ldots + m_t = k$。那么数列 $\{a_n\}$ 是递归关系的解 $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots.. + c_k a_{n-k}$

当且仅当

$$\begin{aligned} a_n = &(\alpha_{1,0} + \alpha_{1,1} n + \cdots + \alpha_{1,m_1-1} n^{m_1-1}) r_1^n \\ &+ (\alpha_{2,0} + \alpha_{2,1} n + \cdots + \alpha_{2,m_2-1} n^{m_2-1}) r_2^n \\ &+ \cdots + (\alpha_{t,0} + \alpha_{t,1} n + \cdots + \alpha_{t,m_t-1} n^{m_t-1}) r_t^n \end{aligned}$$

对于 $n = 0, 1, 2, \ldots,$ 其中 $\alpha_{i,j}$ 是常数 $1 \leq i \leq t$ 且 $0 \leq j \leq m_{i-1}$.

求解任意阶线性齐次递归关系

例子：求出具有初始条件a_0=1，a_1= −2和a_2= −1的递推关系$a_n = -3a_{n-1} - 3a_{n-2} - a_{n-3}$的解

解：特征方程是$r^3 + 3r^2 + 3r + 1 = (r+1)^3 = 0$

所以只有一个3重根 $r = -1$

因此$a_n = \alpha_{1,0}(-1)^n + \alpha_{1,1}\, n(-1)^n + \alpha_{1,2}\, n(-1)^n$

$\alpha_{1,0} = 1,\ \alpha_{1,1} = 3,\ \alpha_{1,2} = -2$

$a_n = (1+3n-2n^2)(-1)^n$

常系数线性非齐次的递推关系

定义：带有常系数的线性**非齐次**递推关系的形式为:

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots\dots + c_k a_{n-k} + F(n),$$

其中 $c_1, c_2, \dots, c_k$ 是实数， F(n) 是一个不全为零的函数，仅依赖于 n

递推关系

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots\dots + c_k a_{n-k}$$

称为**相伴（或关联）的齐次递推关系**

常系数线性非齐次的递推关系2

以下是带有常系数的线性非齐次递归关系:

$$a_n = a_{n-1} + 2^n,$$

$$a_n = a_{n-1} + a_{n-2} + n^2 + n + 1,$$

$$a_n = 3a_{n-1} + n3^n,$$

$$a_n = a_{n-1} + a_{n-2} + a_{n-3} + n!$$

以下是各线性非齐次递归关系对应的相伴线性齐次递归关系:

$$a_n = a_{n-1},$$

$$a_n = a_{n-1} + a_{n-2},$$

$$a_n = 3a_{n-1},$$

$$a_n = a_{n-1} + a_{n-2} + a_{n-3}$$

求解常系数线性非齐次的递推关系[1]

定理 5：如果 $\{a_n^{(p)}\}$是常系数非齐次线性递推关系

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + F(n),$$

的一个特解，那么每个解都是$\{a_n^{(p)} + a_n^{(h)}\}$的形式，其中 $\{a_n^{(h)}\}$是相伴的齐次递推关系

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}.$$

的一个解

求解常系数线性非齐次的递推关系2

例子：求解递归关系 $a_n = 3a_{n-1} + 2n$ 。找到满足初始条件 $a_1 = 3$的解

解：相伴的线性齐次递归关系是 $a_n = 3a_{n-1}$ 。它的解为 $a_n^{(h)} = \alpha 3^n$ ，其中 α 是常数

因为$F(n)= 2n$是一个一次多项式，我们可以尝试线性函数 $p_n = cn + d$作为特解。

假设 $p_n = cn + d$ 是特解,其中c，d是常数

代入 $a_n = 3a_{n-1} + 2n$ 后，得到 $cn + d = 3(c(n-1) + d)+ 2n$

化简得到 $(2 + 2c)n + (2d - 3c) = 0$

因此，$cn+d$是特解当且仅当且仅当 c = − 1 和 d = − 3/2.

所以，特解为 $a_n^{(p)} = -n - 3/2$

根据定理 5，所有解的形式为$a_n = a_n^{(p)} + a_n^{(h)} = -n - 3/2 + \alpha 3^n$, 其中 α 是常数

要找到满足$a_1 = 3$的通解, 将 $n = 1$代入通解

有 $3 = -1 - 3/2 + 3\alpha$, 解得 $\alpha = 11/6$。因此，解为 $a_n = -n - 3/2 + (11/6)3^n$

3. 分治算法与递归关系

小节目录

3.1 分治算法与递归关系

3.2 例子

- 二分查找
- 归并排序
- 整数快速乘法

3.3 主定理

分治算法范式

定义：分治算法通过首先将一个问题划分为一个或多个相同问题的小规模实例，然后利用这些较小问题的解来解决原始问题

例子:

- 二分查找：通过将要查找的元素与中间元素进行比较来工作。原始列表被划分为两个子列表，搜索在适当的子列表中递归进行
- 归并排序：将一个列表划分为两个大致相等大小的子列表，每个子列表通过归并排序递归排序。排序通过逐步合并列表对来完成

分治递归关系

假设一个递归算法将一个规模为 n 的问题划分为 a 个子问题

每个子问题的规模为n/b

假设在合并步骤中需要 $g(n)$ 次额外操作

则解决规模为 n 的问题所需的操作数 $f(n)$满足以下递归关系:

$$f(n) = af(n/b) + g(n)$$

这被称为分治递归关系

例子: 二分查找

二分查找将对大小为 n 的序列中的元素的搜索问题减少到对大小为 $n/2$ 的序列中的搜索。实现这种减少需要两个比较：

- 一个比较用于决定是搜索序列的上半部分还是下半部分
- 另一个比较用于确定序列是否包含元素

因此，如果 $f(n)$ 是在大小为 n 的序列中查找一个元素所需的比较次数，那么满足的递归关系是

$$f\left(n\right)=f\left(n/2\right)+2$$

当 n 是偶数

例子: 归并排序

归并排序算法将一个规模为 n（假设 n 是偶数）的列表分成两个规模为 $n/2$的子列表。它使用少于 n 次比较来合并这两个已排序的列表.

因此，排序一个规模为 n 的序列所需的比较次数 $M(n)$ 满足以下递归关系

$$M(n) = 2M(n/2) + n.$$

例子: 整数快速乘法

一个快速乘法算法将每个2n位整数分解为两块，每一块有n位。然后，原来2n 位的二进制整数的乘法被分解成3 个n 位二进制数的乘法，还要加上移位和加法

假设 a 和 b 是二进制扩展长度为 2n 的整数。设

$a = (a_{2n-1}a_{2n-2} \ldots a_1a_0)_2$ 和 $b = (b_{2n-1}b_{2n-2} \ldots b_1b_0)_2$.

令 $a = 2^nA_1 + A_0$, $b = 2^nB_1 + B_0$, 其中

$A_1 = (a_{2n-1} \ldots a_{n+1}a_n)_2$, $A_0 = (a_{n-1} \ldots a_1a_0)_2$,

$B_1 = (b_{2n-1} \ldots b_{n+1}b_n)_2$, $B_0 = (b_{n-1} \ldots b_1b_0)_2$.

该算法基于以下事实：ab 可以被重新表示为:

$ab = (2^{2n} + 2^n)A_1B_1 + 2^n(A_1 - A_0)(B_0 - B_1) + (2^n + 1)A_0B_0$.

这个公式表明，两个 2n 位整数的乘法可以通过三个 n 位整数的乘法以及加法、减法和移位来完成。因此，如果 f(n) 是乘两个 n 位整数所需的总操作数，则

$$f(2n) = 3f(n) + Cn$$

其中 C_n 表示总的位操作数；这些加法、减法和移位是 n 位操作的常数倍

估算分治算法函数的复杂度

定理 1：设 f（n） 是一个递增函数，满足递归关系

$$f(n) = af(n/b) + c$$

其中 n 是 b 的倍数，$a \geq 1$，b 是大于 1 的整数，c 是一个正实数. 那么

$$f(n) \text{ is } \begin{cases} O(n^{\log_b a}) & \text{if} \quad a > 1 \\ O(\log n) & \text{if} \quad a = 1. \end{cases}$$

此外，当 n = b^k 和 $a \neq 1$时，其中 k 是一个正整数,

$$f(n) = C_1 n^{\log_b a} + C_2$$

其中 $C_1 = f(1) + c/(a-1)$ and $C_1 = -c/(a-1)$.

二分查找的复杂度

二分查找示例：给出二分查找使用的比较次数的大-O 估计.

解答：由于二分查找使用的比较次数是 $f(n)=f(n/2)+2$，其中 n 是偶数，根据定理 1，得出 $f(n)$的时间复杂度是 $O(\log\ n)$.

估计分治算法函数的复杂度2

定理 2： 主定理（Master Theorem）：设 $f(n)$ 是一个递增函数，满足递归关系

$$f(n) = af(n/b) + cn^d$$

当 $n = b^k$, k为大于1的正整数, c为正实数，d为非负实数. 可具体化为

$$f(n) \text{ is } \begin{cases} O(n^d) & \text{if } a < b^d, \\ O(n^d \log n) & \text{if } a = b^d, \\ O(n^{\log_b a}) & \text{if } a > b^d. \end{cases}$$

二分排序复杂度

归并排序示例：给出归并排序所使用的比较次数的大O估计.

解决方案：由于归并排序对大小为 n 的列表进行排序所使用的比较次数小于 $M(n)$，其中 $M(n)=2M(n/2)+\mathrm{n}$。根据主定理，$M(n)$ 是 $O(n\log\ n)$.

快速整数乘法算法的复杂度

整数乘法示例：给出使用快速乘法算法乘以两个 n 位整数所需的比特操作的渐近复杂度估计.

解决方案：我们已经证明，当 n 为偶数时，$f(n) = 3f(n/2) + Cn$，其中 $f(n)$ 是乘以两个 n 位整数所需的比特操作数量。因此，根据主定理，$a = 3$，$b = 2$，$c = Cn$，$d = 0$（因此我们有 $a > b^d$ 的情况），可以得出 $f(n)$ 是 $O(n^{\log 3})$,

注意 $\log 3 \approx 1.6$。因此，快速乘法算法相比于使用 $O(n^2)$ 比特操作的传统算法有显著的改进

4. 容斥原理

小节目录5

容斥原理

我们知道两个有限集合的并集的元素数量的公式是:

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

我们将这个公式推广到任意大小的有限集合

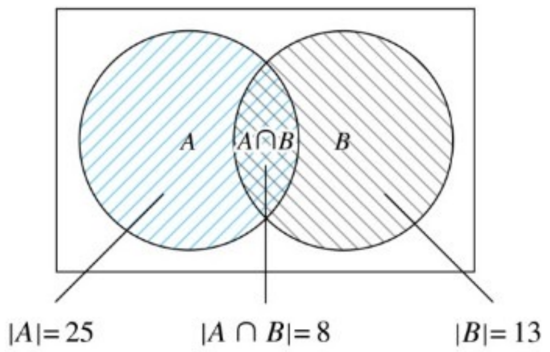
两个有限集

示例：在一门离散数学课上，每位学生都是计算机科学或数学专业，或两者兼修。主修计算机科学（可能同时主修数学）的学生人数是 25；主修数学（可能同时主修计算机科学）的学生人数是 13；同时主修计算机科学和数学的学生人数是 8。这个班级总共有多少学生?

解答:

$$\begin{aligned}|A \cup B| &= |A| + |B| - |A \cap B| \\ &= 25 + 13 - 8 = 30\end{aligned}$$

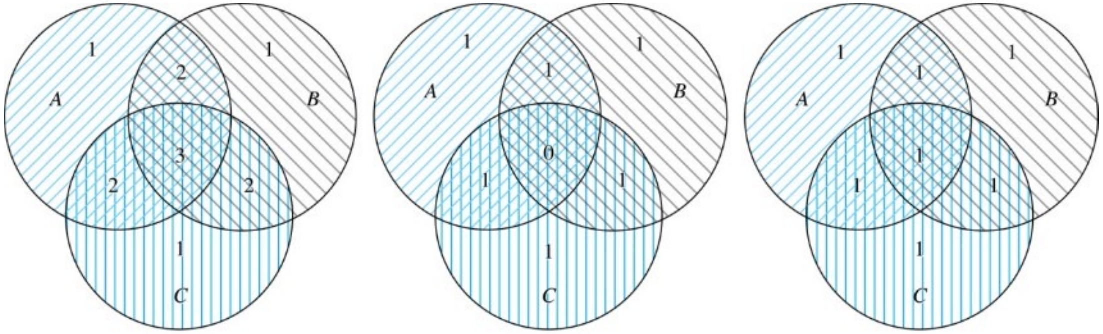
$|A \cup B| = |A| + |B| - |A \cap B| = 25 + 13 - 8 = 30$



Jump to long description

三个有限集[1]

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



(a) Count of elements by $|A|+|B|+|C|$

(b) Count of elements by $|A|+|B|+|C|-|A\cap B|-|A\cap C|-|B\cap C|$

(c) Count of elements by $|A|+|B|+|C|-|A\cap B|-|A\cap C|-|B\cap C|+|A\cap B\cap C|$

Jump to long description

三个有限集[2]

示例：总共有 1232 名学生修过西班牙语课程，879 名学生修过法语课程，114 名学生修过俄语课程。此外，103 名学生修过西班牙语和法语课程，23 名学生修过西班牙语和俄语课程，14 名学生修过法语和俄语课程。如果 2092名学生修过至少一种西班牙语、法语和俄语课程，那么有多少学生修过这三种语言的课程.

解答：设 S 为修过西班牙语课程的学生集合，F 为修过法语课程的学生集合，R 为修过俄语课程的学生集合。那么，我们有

$|S| = 1232$, $|F| = 879$, $|R| = 114$, $|S \cap F| = 103$, $|S \cap R| = 23$, $|F \cap R| = 14$, 和 $|S \cup F \cup R| = 2092$

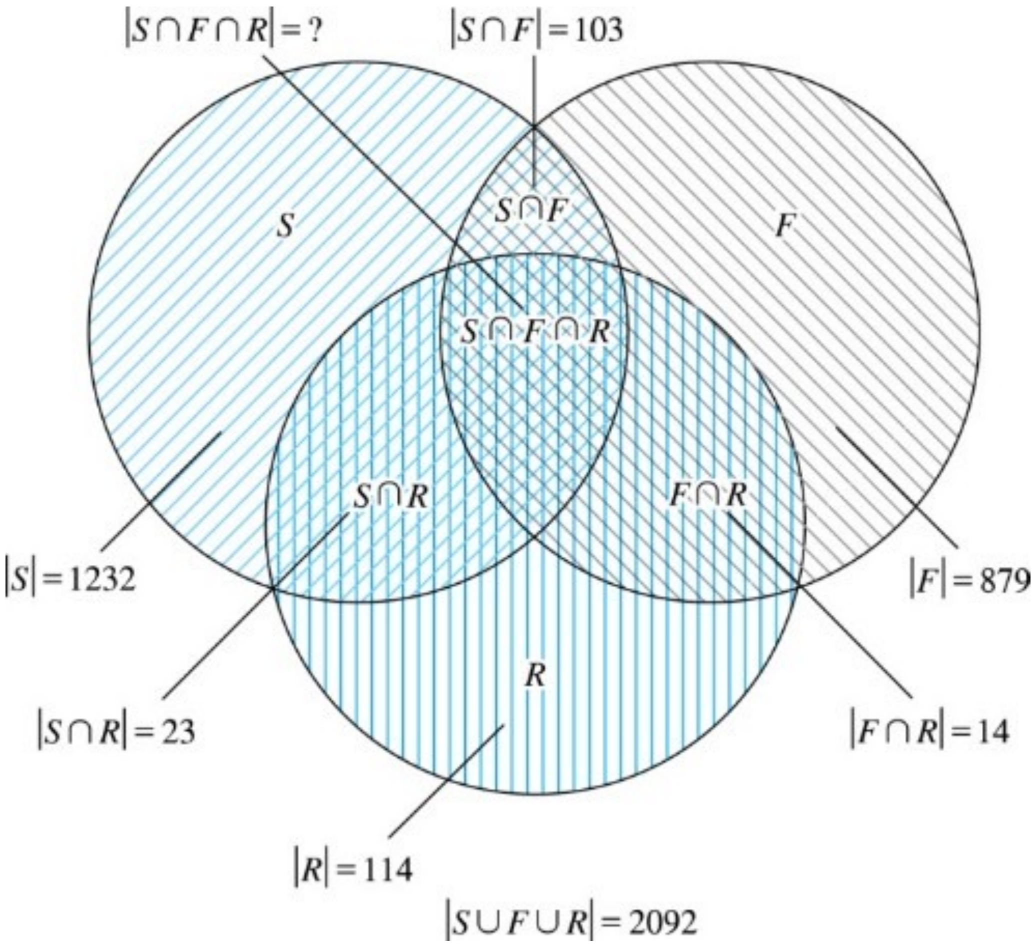
使用等式

$$|S \cup F \cup R| = |S| + |F| + |R| - |S \cap F| - |S \cap R| - |F \cap R| + |S \cap F \cap R|,$$

我们得到 $2092 = 1232 + 879 + 114 - 103 - 23 - 14 + |S \cap F \cap R|$.

解得 $|S \cap F \cap R| = 7$

三个有限集合的示例图解



容斥原理[1]

定理 1. 容斥原理: 令 $A_1, A_2, \ldots, A_n$ 为有限集，那么：

$$|A_1 \cup A_2 \cup \cdots \cup A_n| = \sum_{1 \leq i \leq n} |A_i| - \sum_{1 \leq i < j \leq n} |A_i \cap A_j|$$

$$+ \sum_{1 \leq i < j < k \leq n} |A_i \cap A_j \cap A_k| - \cdots + (-1)^{n+1} |A_1 \cap A_2 \cap \cdots \cap A_n|$$

容斥原理2

证明：一个元素在并集的右边表达式中恰好被计数一次。考虑一个元素 a 属于 $A1,A2,\ldots,An$ 中的 r 个集合，其中 $1 \leq r \leq n$.

- 被 $\sum |A_i|$ 计算了 $C(r,1)$ 次
- 被 $\sum |A_i \cap A_j|$ 计算了 $C(r,2)$ 次
- 一般来说，它涉及 m 个 A_i 集合的求和，被计数了 $C(r,m)$次

容斥原理3

因此这个元素实际上被计算了

$$C(r,1)-C(r,2)+C(r,3)-\cdots+(-1)^{r+1}C(r,r)$$

次，被等式的右边,（等于“1”）.

根据二项式展开，我们有

$$C(r,0)-C(r,1)+C(r,2)-\cdots+(-1)^{r}C(r,r)=0.$$

因此,

$$1=C(r,0)=C(r,1)-C(r,2)+\cdots+(-1)^{r+1}C(r,r).$$

6. 应用容斥原理的场景

小节目录6

6.1 计数 Onto 函数

6.2 错位排列

容斥原理的另一种形式

容斥原理有另一种表述形式，它在计数问题中很有用。特别是，这种形式可以用于求解在一个集合中的元素数，使得这些元素不具有 n 个性质 P_1，P_2，…，P_n 中的任何一条性质。

设 A_i 是具有性质 P_i 的元素的子集。具有所有这些性质 P_{i_1}，P_{i_2}，…，P_{i_k} 的元素数将记为 $N(P_{i_1}P_{i_2}\cdots P_{i_k})$。用集合的术语写这些等式，有

$$|A_{i_1}\cap A_{i_2}\cap\cdots\cap A_{i_k}|=N(P_{i_1}P_{i_2}\cdots P_{i_k})$$

如果不具有 n 个性质 P_1，P_2，…，P_n 中的任何一个的元素数记为 $N(P_1'P_2'\cdots P_n')$，集合中的元素数记为 N，那么有

$$N(P_1'P_2'\cdots P_n')=N-|A_1\cup A_2\cup\cdots\cup A_n|$$

由容斥原理，有

$$N(P_1'P_2'\cdots P_n')=N-\sum_{1\leqslant i\leqslant n}N(P_i)+\sum_{1\leqslant i<j\leqslant n}N(P_iP_j)-\sum_{1\leqslant i<j<k\leqslant n}N(P_iP_jP_k)$$
$$+\cdots+(-1)^nN(P_1P_2\cdots P_n)$$

Onto（满射）函数的数量[1]

例子：有多少个从一个有六个元素的集合到一个有三个元素的集合的 onto 函数?

解答：假设余像集合中的元素为：b_1, b_2, b_3。设 P_1, P_2, P_3 分别表示 b_1, b_2, b_3 不在函数的值域中的性质。函数是 onto 函数的条件是当且仅当它没有性质P_1, P_2, P_3

根据容斥原理，从一个有六个元素的集合到一个有三个元素的集合的 onto 函数的数量为

$N - [N(P_1) + N(P_2) + N(P_3)] + [N(P_1P_2) + N(P_1P_3) + N(P_2P_3)] - N(P_1P_2P_3)$

- 这里从一个有六个元素的集合到一个有三个元素的集合的总函数数为 $N = 3^6$.
- 不包含 $b1$ 在值域中的函数数量为 $N(P_1) = 2^6$. 类似的， $N(P_2) = N(P_3) = 2^6$.
- 注意到 $N(P_1P_2) = N(P_1P_3) = N(P_2P_3) = 1$ 和 $N(P_1P_2P_3)= 0$.

因此，从一个有六个元素的集合到一个有三个元素的集合的 onto 函数数量为:

$3^6 - 3 \cdot 2^6 + 3 = 729 - 192 + 3 = 540$

Onto 函数的数量$_2$

定理 1: m，n为正整数且$m \geq n$. 那么有

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

个onto 函数 从一个有m个元素的集合到有n个元素的集合

证明来自于容斥原理

错位排列[1]

定义：错位排列是指一个排列中，所有的元素都没有出现在它们的原始位置上

例子：排列 21453 是 12345 的一个错排，因为没有一个数字留在它的原始位置上。但是排列 21543 不是 12345 的错排，因为数字 4 仍在它的原始位置上

错位排列₂

定理 2：一个包含 n 个元素的集合的错排数为

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

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 $i=1, 2, \cdots, n$，没有性质 P_i 的排列数，或

$$D_n=N(P'_1P'_2\cdots P'_n)$$

使用容斥原理得到

$$D_n=N-\sum_i N(P_i)+\sum_{i<j}N(P_iP_j)-\sum_{i<j<k}N(P_iP_jP_k)+\cdots+(-1)^nN(P_1P_2\cdots P_n)$$

其中 N 是 n 个元素的排列数。这个等式说明所有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 $N=n!$，因为 N 只是 n 个元素排列的总数。而且，$N(P_i)=(n-1)!$。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，所以第 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，

$$N(P_iP_j)=(n-2)!$$

因为这是保持元素 i 和 j 不变的排列数，但是其余 $(n-2)$ 个元素的位置可以任意地安排。一般来说，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，$\cdots$，i_m 不变的排列数，但是其他 $(n-m)$ 个元素的位置可以任意安排。由于存在 $C(n, m)$ 种方式从 n 个元素中选择 m 个，所以有

$$\sum_{1\leqslant i\leqslant n}N(P_i)=C(n,1)(n-1)!$$

$$\sum_{1\leqslant i<j\leqslant n}N(P_iP_j)=C(n,2)(n-2)!$$

一般地，有

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

所以，把这些等式代入关于 D_n 的公式得到

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

简化这个表达式得

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

错位排列[3]

帽子寄存问题：一位新员工在餐厅为 n 个人寄存帽子，但忘记在帽子上放置领取单号。当顾客回来取帽子时，寄存员从剩下的帽子中随机给他们分发帽子。问：没有一个人拿到自己帽子的概率是多少

解：答案是帽子排列的方式中没有一个帽子在其原位置的排列数量，除以 n!，即 n 个帽子的全排列数量.

备注：可以证明，当 n 无限增大时，错位排列的概率趋近于 1/e

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

TABLE 1 The Probability of a Derangement.

n	2	3	4	5	6	7
$D_n/n!$	0.50000	0.33333	0.37500	0.36667	0.36806	0.36786