

MAT120: 论证 第5讲讲义

数字与计数

今天我们将学习三讲代数课中的第二讲。事实上，这一讲会比较有趣，因为我们会稍微离开代数本身，转而研究“数”和“计数”这两个概念。听起来这也许像是一个很傻的话题，但我可以向你保证，它一点也不傻。到这一讲结束时，你会理解前沿数学家是如何思考这些概念的，而且我还会介绍一些你可能从未听过的、极其出人意料的数学内容。

1 其他数制中的计数

在前面的课中，我讨论过“符号”和“意义”之间的区别（比较正式的说法是“语法”和“语义”）。在任何语言中，都有一些符号，而这些符号带有某种解释。我们看到，命题逻辑也不例外：在这种“形式语言”中，有符号（字母、联结词、括号），而这些符号的意义由它们所携带的真值来解释。在某种意义上，我们的数字系统也同样属于语言的范畴：有符号（1、2、3，等等），而这些符号带意义。例如，符号“3”的意思就是三个。在本节中，我们会看到，我们使用的数字符号其实有多么随意。

1.1 数字符号简史

在历史上，人类曾经用过许多不同的方式来表示数。我们目前掌握的最早证据表明，数的最早表示方式可能是“刻痕系统”，也就是用一组线条来记录数量。刻痕系统最早的例子之一来自大约公元前20000至25000年：伊尚戈骨。它发现于今天的刚果，是一根动物骨头，上面刻有许多凹痕。骨头的一端还嵌有一块可拆卸的石英晶体，这说明这很可能是一种有明确目的的实践。有人推测，这些刻痕代表月相周期，或者可能用于记录月经周期。虽然我们最终永远无法完全知道历史的秘密，但我们也可以有趣地猜测一下：也许最早被我们知道的数学家，是非洲的女性。

快速跳过漫长的历史之后，世界各地又独立产生了许多其他类型的计数系统。在古代中国，现代汉字书写系统的早期形式正在出现，它也起初像一种用于表示较小数量的刻痕系统。巴比伦、古希腊、古罗马和印度也都出现过书写数字的系统，而且它们看起来彼此非常不同。不过，这些系统中的许多在表示小数时，都是从类似刻痕的方式开始的。到了大约数字4之后，单纯用数量本身来表示一个数就变得麻烦且低效，所以人们引入了更加抽象的符号。在下表中，请观察古代世界在书写1, 2 和3 时的相似之处。

文化	最早证据	1	2	3	4	5
伊尚戈骨	~公元前20000 年	I	II	III	IIII	IIII I
巴比伦	~公元前2000 年	Y	YY	YYY	YYY Y	YYY YY
埃及（僧侣体）	~公元前2500 年		U	UU	UUU	U S
中国（甲骨文）	~公元前1200 年	一	=	≡	≡	X
中国（算筹）	~公元前200 年	I	II	III	IIII	——
雅典（阿提卡）	~公元前500 年					□
罗马	~公元前500 年	I	II	III	IV	V
中美洲	~公元前300 年	•	••	•••	••••	——

1.2 数制的基数

在现代世界中，最常见的书写系统使用我们熟悉的符号：0, 1, 2, 3, 4, 5, 6, 7, 8, 9。这个书写系统叫作“印度-阿拉伯数字系统”，因为它混合了古代阿拉伯和印度的数字系统，并在中世纪传入欧洲。我们用来表示数的符号本身是任意的，它们完全可以长得不一样，而数字系统仍然可以正常运作。印度-阿拉伯数字的力量并不在于这些符号本身，而在于它们背后的系统。接下来我们会对此作一些详细讨论。

考虑一个像2173这样的数。在脑海里，我们把它读成“两千一百七十三”。我们可以把它在数学上拆开：

$$2173 = 2 \times 1000 + 1 \times 100 + 7 \times 10 + 3.$$

所以事实上，我们表示数的方式是通过10的幂来完成的。为了让这一点更清楚，请记得 $1 = 10^0$ ， $100 = 10^2$ ，而 $1000 = 10^3$ 。于是我们可以把2173改写为10的幂的组合：

$$2173 = 2 \times 1000 + 1 \times 100 + 7 \times 10 + 3 = 2 \times 10^3 + 1 \times 10^2 + 7 \times 10^1 + 3 \times 10^0.$$

事实上，当我们书写数字时，是从右往左把数字放在越来越高的10的幂所对应的列中。这样的数学系统叫作“十进制”，因为10是决定每一列如何工作的基本数字。在十进制计数系统中，有十个独特的数字符号（即0, 1, ..., 9），它们会在一个数位中依次出现。

1.2.1 外星人

我们使用十进制进行数学运算是充分合理的：人类有10根手指，所以用手指数数对我们来说非常自然。不过，从数学角度看，完全没有什么特殊理由一定要坚持使用十进制。为了看清这一点，让我们考虑下面这个想法。

想象在遥远星空中的某颗行星上生活着一种外星人。这些外星人很奇怪，身体也和我们不同：他们有两条手臂，但每只手只有4根手指。我们可以想象，这个外星物种也许会像我们一样学会计数：他们会沿着自己的外星手指一个个数过去，直到数到最后一根手指，然后才进入书写系统中的新一列。但是，因为他们每只手只有4根手指，所以这些外星人会在第8根手指处停下，然后在下一列重新开始。

举个例子，我们现在展示这些外星人会如何用一种完全不同的方式写出数2173。他们不会把2173表示成10的幂之和，而会把它分解成8的幂之和：

$$\begin{aligned} 2173 &= 2048 + 64 + 56 + 5 \\ &= 4 \times 8^3 + 1 \times 8^2 + 7 \times 8^1 + 5 \times 8^0 \end{aligned}$$

每增加一个8的幂，就会给我们一个新的数位，而前面的乘数就给出了同一个数的另一种数字表示，也就是我们称为“2173”的那个数。按照外星人的系统，“两千一百七十三”因此会在八进制中写成“4175”。

在这些外星人看来，他们也许会觉得我们这些“奇怪的人类”居然使用十进制！他们可能会说类似“glib glob glib glob”的话，用英语表达就是“哇，那些人类数数真奇怪……他们在7后面还多用了两个符号，并且继续用它们数下去！”。那么，到底谁是正确的呢，是我们，还是他们？答案是两边都不是在使用某种“宇宙通用”的系统。我们只是选择了一种适合我们的系统而已。

1.2.2 其他基数

希望前面的例子已经说明，完全没有理由一定要坚持十进制。一般来说，我们也可以选择另一个基数，把它叫作 b ，然后把“2173”转换成一个 b 进制数，方法是把它写成 b 的递增幂的和：

$$2173 = a_k \times b^k + a_{k-1} \times b^{k-1} + \cdots + a_1 \times b + a_0 \times b^0.$$

根据 b 的大小，我们需要或多或少的 b 的幂来凑出2173。这样，2173转换成 b 进制之后就会写成： $a_k a_{k-1} \cdots a_1 a_0$ (b 进制)。还要注意，在 b 进制中，我们唯一可能使用的数字是 $0, 1, \dots, (b-1)$ 。

例

我们考虑数211（十进制），并把它写成五进制。为了把211写成五进制，我们首先需要把它表示成5的递增幂之和。首先，我们回忆 $5^2 = 25$ ， $5^3 = 125$ ，而 $5^4 = 625$ 。因为数211小于 5^4 ，所以为了用五进制表示211，我们不需要一直用到5的第4次幂。因此，我们预期所需的和为：

$$\begin{aligned} 211 &= a_3 \times 5^3 + a_2 \times 5^2 + a_1 \times 5^1 + a_0 \times 5^0 \\ &= a_3 \times 125 + a_2 \times 25 + a_1 \times 5 + a_0 \times 1. \end{aligned}$$

从这里开始，我们注意到 $2 \times 125 = 250$ ，这比211大，所以在这个和中只需要使用一个125。这意味着 $a_3 = 1$ 。接下来，我们取差 $211 - 125 = 86$ ，然后尝试写成

$$86 = a_2 \times 5^2 + a_1 \times 5^1 + a_0 \times 5^0.$$

同样，我们从最大的可用幂开始，也就是 $5^2 = 25$ 。我们观察到 $4 \times 25 = 100$ ，但 $3 \times 25 = 75$ ，所以为了凑出86，我们需要使用3个 $5^2 = 25$ 。这意味着 $a_2 = 3$ 。现在我们重复这个推理，取差 $86 - 75 = 11$ 。我们现在想表示 $11 = a_1 \times 5^1 + a_0 \times 5^0$ 。幸运的是，我们看到 $11 = 2 \times 5 + 1$ ，这意味着 $a_1 = 2$ 且 $a_0 = 1$ 。把这些放在一起，我们得到

$$\begin{aligned} 211 &= 1 \times 5^3 + 3 \times 5^2 + 2 \times 5^1 + 1 \times 5^0 \\ &= 1 \times 125 + 3 \times 25 + 2 \times 5 + 1 \times 1 \end{aligned}$$

因此，我们把211写成五进制中的“1321”。

1.3 二进制计数

我们接下来研究二进制计数的思想。“binary”这个词最终来自拉丁词“bini”，意思大致是“两个东西在一起”。现代英语中的前缀“bi”也来自这里，例如bicycle、biannual、binoculars等等。在我们的语境中，二进制计数系统就是以2为基数的计数系统。这意味着 $b = 2$ ，因此用于计数的数字只有两个：0和1。换句话说，在二进制系统中，我们只使用0和1的组合来表示所有其他数。例如，“11”表示“三”，而“101”表示“五”。当然，我们通常能用十进制写出的任何数，也都可以“翻译”成二进制。

1.3.1 将十进制转换为二进制

前面我们看到

$$2173 = 2 \times 10^3 + 1 \times 10^2 + 7 \times 10^1 + 3 \times 10^0.$$

在这种情况下，数2173 可以通过把10 的幂相加来描述，而且这种描述是唯一的。如果我们想改成二进制，就必须把这些“ 10^n ”项替换成“ 2^n ”项。因为2 小于10，所以我们可以预期，可能需要超过3 个2 的幂才能一直凑到2173。事实上，使用计算器可以确认：

$$2173 = 2048 + 64 + 32 + 16 + 8 + 4 + 1.$$

更明确地写出来，我们必须一直用到2 的第11 次幂才能表示2173：

$$\begin{aligned} 2173 = & 1 \times 2^{11} + 0 \times 2^{10} + 0 \times 2^9 + 0 \times 2^8 + 0 \times 2^7 \\ & + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0. \end{aligned}$$

所以，为了把2173 改写成二进制，我们必须连续使用十二个数字。2173 的二进制表示因此是100001111101（二进制）。

请注意，2 的幂前面的每个倍数只会是1 或0。为什么？因为如果我们试图乘以一个更大的数（比如2），那么它就会直接变成下一个2 的幂，并被下一位吸收。

1.3.2 一般规则

转换成二进制的一般规则是，尝试把这个数写成2 的递减幂之和。每个2 的幂前面的数字要么是1，要么是0，而这些数字就构成了你一开始的十进制数的二进制表达。

要把数转换成二进制，我们遵循第1.2.2 节例子中给出的一般方法。例如，假设我们想把17 写成二进制。2 的幂是1, 2, 4, 8, 16, 32, ...。正如我们所见，17 介于16 和32 之间，所以为了用二进制表示17，我们不需要一直用到32。事实上，我们看到 $17 = 16 + 1$ ，所以把17 用所有对应的二进制位写出来就是：

$$17 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0.$$

因此，17 的二进制表达是10001。

练习

把下面这些数写成二进制。

1. 4
2. 7
3. 5

解答

$$\begin{aligned} 4 &= 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 && \text{所以写作100（二进制）} \\ 7 &= 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 && \text{所以写作111（二进制）} \\ 5 &= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 && \text{所以写作101（二进制）} \end{aligned}$$

1.3.3 二进制的优点

我们用二进制数到3: 0,1,2,3 会变成00, 01, 10, 11。如果我们把这些排成一行, 就突然得到了一个看起来很像真值表的东西。说得更清楚一些: 让我们假装 $0 = F$ 且 $1 = T$ 。那么“FF, FT, TF, TT”其实就是在用二进制从0 数到3!

这不是巧合: 计算机使用二进制数作为一种计数方式: $0 = \text{“关”}$, $1 = \text{“开”}$, 它们是电信号可能处于的两种状态(高电压/ 低电压)。计算机通过逻辑来解释这些开/ 关信号。组合信号并利用这些组合产生新信号的规则只有很少几种, 所以现代计算机实际上由数以百万计的微小机器组成, 这些机器有一个特殊名称: 逻辑门。逻辑门只有少数几种, 而且它们的工作方式与我们之前的真值表完全一样。这就是为什么我们用鸟居门来解释真值表: 在计算机内部, 确实有数以百万计的微小部件, 它们的行为就像真值表一样。所以, 二进制计数的整个思想对计算机非常有用。

1.4 其他进位系统

历史上曾经有很多不同类型的进位系统: 例如, 巴比伦人使用六十进制计数。现在看来这也许有点奇怪, 但在当时他们非常高产。巴比伦人创造了各种数学和科学系统, 其中一些在几千年后的今天仍然被我们使用。例如, 一个圆有360 度, 是因为巴比伦人; 一小时分为60 分钟, 一分钟分为60 秒, 也是因为巴比伦人。

计算机还使用另一种计数系统: 十六进制, 也叫作“hexadecimal”。原因是计算机喜欢用2 的幂的组合来计数, 因为这对它们来说更加高效。在这种情况下, 我们会遇到一个有趣的问题: 因为我们通常只用十进制计数, 所以只有10 个书写数字的符号, 也就是0,1,2,3,4,5,6,7,8 和9。如果我们使用十六进制计数, 就会发现符号不够用了。因此, 我们必须使用其他符号来表示十六进制中的10 到15。最常见的系统是使用大写字母A,B,C,D,E 和F, 也就是 $A=十$, $B=十一$, 依此类推。这就是为什么在Photoshop 这样的计算机程序中, 颜色会用类似“0FA123”这样的数据串来表示, 因为这实际上是一个十六进制数。

2 组合的计数

假设你刚吃完早餐, 正在决定今天穿什么。你有5 件不同的衬衫和3 条不同的裤子可供选择。用这些选项, 你能搭配出多少套不同的衣服?

我们用一些数学记号把这件事说清楚。我们把五件不同的衬衫标记为 s_1, s_2, s_3, s_4, s_5 , 把3 条不同的裤子标记为 t_1, t_2, t_3 。一套衣服由每一类中各选一个组成, 所以它可以写成一个有序对。例如, (s_3, t_2) 表示你选择了第三件衬衫和第二条裤子。现在假设我们决定选择第一件衬衫 s_1 。我们看到, 关于衣服搭配有三种选择, 对应三条不同的裤子: $(s_1, t_1), (s_1, t_2)$ 和 (s_1, t_3) 。同样, 如果我们选择第二件衬衫, 那么也有三种可能的搭配, 即 $(s_2, t_1), (s_2, t_2)$ 和 (s_2, t_3) 。这个推理对所有可能的衬衫都成立, 所以我们可以把所有选项排列成一个网格:

$$\begin{array}{ccccc} (s_1, t_1) & (s_2, t_1) & (s_3, t_1) & (s_4, t_1) & (s_5, t_1) \\ (s_1, t_2) & (s_2, t_2) & (s_3, t_2) & (s_4, t_2) & (s_5, t_2) \\ (s_1, t_3) & (s_2, t_3) & (s_3, t_3) & (s_4, t_3) & (s_5, t_3) \end{array}$$

正如你所见, 这个网格中的总数正好是15, 也就是可能的衬衫数量乘以可能的裤子数量。

这个“乘法规则”一般都适用：如果你分两个阶段作选择，那么可能结果的数量就是第一阶段选择数与第二阶段选择数的乘积。用符号来说：如果第一步中的每一个 m 种选择，在第二步中都有 n 种选择，那么组合总数就是 $m \cdot n$ 。这个规则也适用于由两个以上阶段组成的选择，下面的例子展示了这一点。

例

考虑一个命题公式 $(p \wedge q) \rightarrow (r \wedge (s \vee t))$ 。在这个公式中，有5个命题变量，每一个都可以取值 T 或 F 。 $(p \wedge q) \rightarrow (r \wedge (s \vee t))$ 的真值表会考虑这些命题变量可能取到的所有 T 和 F 的组合，然后在每一种情况下计算该公式的真值。在这个例子中，每个变量字母有两个可能的“选择”，即 T 或 F ，而总共有5个字母。因此，可能性的总数是 $2 \times 2 \times 2 \times 2 \times 2 = 2^5 = 32$ 。这意味着，与公式 $(p \wedge q) \rightarrow (r \wedge (s \vee t))$ 相关的真值表需要32行。

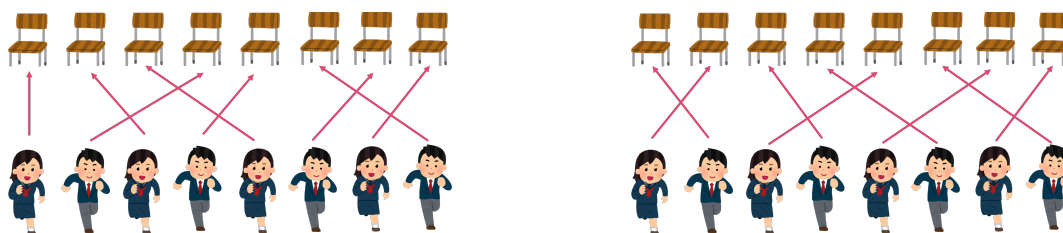
接下来，我们将研究更复杂的情形。在这些情形中，连续作出的选择会改变可能性的数量，因此也会改变可能组合的数量。

阶乘记号

在进入其他内容之前，我们首先需要引入一个新的记号：阶乘。简而言之，阶乘是一种运算，它把一个正整数与所有比它小的正整数相乘，一直到1为止。例如，“4的阶乘”就是 $4 \times 3 \times 2 \times 1 = 24$ ，而“8的阶乘”就是 $8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 40320$ 。我们停在1，是因为如果停在0，那么就会乘以0，而答案也会变成0。我们用感叹号来写阶乘记号，也就是说，“ n 的阶乘”在数学上写作 $n!$ 。当 n 大于2时，阶乘 $n!$ 总是大于 n 本身，而且通常会大得多。

2.1 用8名学生和8把椅子计算排列数

假设O'Connell博士有一个新班级，里面有8名学生。他的教室里有8把椅子。O'Connell博士决定通过把学生分配到教室里的椅子上，来创造一种“座位安排”。例如，下面的图片包含两种可能的座位安排。



把8名学生分配到教室里的8把椅子上，一共有多少种方法？换句话说，一共有多少种可能的“座位安排”？我们现在来计算这个数。

- **第一把椅子：** 因为有8名学生，所以第一把椅子有8种不同的学生选择。换句话说：第一把椅子有8种可能选择。因此，O'Connell博士选择一名学生，让他坐在第一把椅子上。
- **第二把椅子：** 因为已经有一名學生坐在第一把椅子上，所以现在只有7名学生可能坐在第二把椅子上。同样，O'Connell博士选择一名学生，让他坐下。
- **一般过程：** 如果我们一次又一次重复这个过程，每把椅子上坐一名学生，我们会看到，这个过程不断把选择数量减少1。最终，当7把椅子已经分配完后，只剩一种可能选择：最后一名学生必须坐在最后剩下的椅子上。因此，可能选择的数量由阶乘给出： $8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 8!$

一般来说，如果有 n 个东西，并且考察它们的排列或重新排列方式，那么一共有 $n!$ 种可能组合。例如，一副普通扑克牌有 52 张牌。如果我洗牌，我就创造了一种新的排列，也就是这些牌处在不同的顺序中。那么，这些牌有多少种不同的洗法？答案是 $52!$ ，这是一个巨大到难以想象的数。 $52!$ 近似等于 8.07×10^{67} 。作为参考：地球上的沙粒数量“只有” 7.6×10^{18} ，宇宙中的恒星数量大约是 4×10^{23} ，而一杯水中的原子数量约为 2.5×10^{25} 。

2.2 用 8 名学生和 5 把椅子计算排列数

现在假设 O'Connell 博士的教室里只有 5 把椅子。所以，如果他想让 8 名学生坐下，那么只有 5 名学生能坐下，其余 3 名学生必须站着。我们的下一个问题是：我们有多少种可能的方式让 8 名学生中的 5 名坐下？下面画出了两种这样的安排。



我们可以重复之前的推理，并看到第一把椅子有 8 种选择，第二把椅子有 7 种选择，依此类推。不过，因为现在只有 5 把椅子可以分配，所以我们必须在分配 5 把椅子之后停止，而不是一直分配到 8 把椅子。这意味着答案不会是 $8!$ ，因为我们并不关心最后那 3 名必须站着的学生如何排列。相反，答案只需要乘前 5 项，因为只有 5 把椅子可用。答案是： $8 \times 7 \times 6 \times 5 \times 4$ 种可能的座位安排。

一般来说，令 n 为学生数量， r 为椅子数量。我们把这 n 名学生安排到 r 把椅子上的方法数可以用公式

$$\frac{n!}{(n-r)!}$$

来计算。我们除以 $(n-r)!$ 的原因是， $n!$ 这一项数得比我们想要的更多：我们并不关心那 $(n-r)$ 名必须站着的学生如何排序。因此，我们除以 $(n-r)!$ 来从总计数中去掉这些组合。为了检查一下，令 $n = 8$ 且 $r = 5$ 。那么：

$$\frac{n!}{(n-r)!} = \frac{8!}{3!} = \frac{8 \times 7 \times 6 \times 5 \times 4 \times (3 \times 2 \times 1)}{(3 \times 2 \times 1)} = 8 \times 7 \times 6 \times 5 \times 4,$$

这与我们之前得到的答案相同。这些“重新排列”在数学中实际上叫作“排列”。把 n 个对象放入 r 个位置的所有排列数量写作 $P_{n,r}$ ，有时也写作 $P(n, r)$ ，其公式为：

$$P(n, r) = \frac{n!}{(n-r)!}.$$

2.3 选择的计数

在讨论排列时，我们关心学生被安排的顺序。也就是说，O'Connell 博士会把所有椅子排成一列，然后通过说“你第一个，你第二个，...”来分配不同的学生。我们可以想象，也许还有另一种做法：更一般地说，O'Connell 博士也许只是选择 r 名学生，而不关心他们在椅子上的顺序。这就相当于从 8 名学生中选择 5 名学生。下面画出了两种这样的选择。



2.3.1 组合公式

$$C_{n,r} = \frac{n!}{r!(n-r)!}.$$

2.3.2 帕斯卡三角形

第0行:			1										$C_{0,0}$	
第1行:			1		1								$C_{1,0}$ $C_{1,1}$	
第2行:			1		2		1						$C_{2,0}$ $C_{2,1}$ $C_{2,2}$	
第3行:			1		3		3		1				$C_{3,0}$ $C_{3,1}$ $C_{3,2}$ $C_{3,3}$	
第4行:			1		4		6		4		1		$C_{4,0}$ $C_{4,1}$ $C_{4,2}$ $C_{4,3}$ $C_{4,4}$	
第5行:			1		5		10		10		5		1	$C_{5,0}$ $C_{5,1}$ $C_{5,2}$ $C_{5,3}$ $C_{5,4}$ $C_{5,5}$
⋮							⋮							⋮

请注意，每一行由 $C_{n,r}$ 中的“ n ”编号，而该行中的各项则通过让 r 在0到 n 之间取不同值来得到。这意味着我们不一定需要直接计算 $C_{n,r}$ 的值，而可以直接从这个三角形中读出它的值。

3 子集的计数

几讲之前，我们看过一个例子：上帝把世界上所有对象排成一排，并把它们放进盒子里。例如，上帝有一个装着所有吉他的盒子，一个装着所有泡菜的盒子，等等。我们用这个例子来引出“量词”的讨论。量词是一类小小的语言工具，它们允许我们“往这些盒子里面看，并检查里面的东西”。事实上，这个类比当时是有意保持非正式的。在现实中，我们脑海中的盒子并没有清楚而精确的“边界”，所以很难恰当地分类对象。例如：什么东西应该放入所有椅子的盒子中？凳子算椅子吗？椅子的画算椅子吗？树桩算椅子吗？坏掉的椅子又算吗？

在数学中，一切都以理想化的方式被精确定义。因此，我们可以避免定义不清的概念，并且可以使用明确的、定义良好的盒子来分类对象。在最简单的解释中（也就是我将介绍的解释），这些“盒子”叫作“集合”，而它们是现代数学的基础。

3.1 什么是集合？

朴素地说，一个集合就是一组无序对象，很像一个盒子。集合本质上没有什么额外结构，它只由它所包含的元素来定义。我们用花括号 $\{$ 和 $\}$ 来书写集合。所以，包含数字2, 4和5的集合写作 $\{2, 4, 5\}$ 。顺序完全不重要，所以同一个集合也可以写作 $\{4, 2, 5\}$ 或 $\{5, 2, 4\}$ 。如果一个对象属于某个集合，我们就称它为这个集合的一个“元素”。因此，在我刚才给出的例子中，集合 $\{2, 4, 5\}$ 的元素是2, 4和5（也只有这些）。

3.2 集合的规则

集合有许多规则和有趣性质，研究集合及其规则的领域叫作“集合论”。这是一个非常有趣的主题，不过在这门课中，我们只关心如何数这些东西，所以不能太深入地讨论集合论。对我们的目的来说，有几条重要规则需要说明。

规则1: 集合只由其中的对象定义，而且同一个对象不能出现两次。这意味着集合 $\{1, 1, 2\}$ 与集合 $\{1, 2\}$ 是同一个集合。所以我们的第一条规则是：不要把同一个对象写很多次。

规则2: 有一个基本对象，它是不包含任何东西的集合。非正式地说，你可以把它想象成这样： $\{\}$ 。这叫作“空集”，因为里面什么都没有。我们用符号 $\emptyset$ 表示空集。

规则3: 集合基本上可以包含任何数学对象，包括集合本身！我们用多重括号来书写这种情况：记号 $\{1, 2, \{3, 4\}\}$ 表示一个有三个元素的集合：1、2和集合 $\{3, 4\}$ 。

规则4: 一个集合不能把它自己作为元素包含进去。

事实上，集合实际上有相当多的规则。有兴趣的读者可以了解公理化集合论，例如策梅洛-弗兰克尔集合论。不过，因为我们这里只是非常轻地讨论这些内容，所以我们只采用非正式的集合处理方式。

3.3 子集的计数

顾名思义，“子集”是另一个集合，它只包含我们一开始集合中的一部分元素。说得更精确一些：如果 X 的每一个元素也都在 Y 中，那么集合 X 就是集合 Y 的一个子集。例如，集合 $\{2, 4\}$ 是 $\{2, 4, 5\}$ 的一个子集。

练习

令 $X = \{a, b, c, d\}$ 。下面这些集合是不是 X 的子集？请用是或否回答。

1. $\{a, b\}$
2. $\{c, e\}$
3. $\{a, b, c, d\}$
4. $\emptyset$

答案：是，否，是，是。

从任何集合 X 出发，我们都可以构造另一个集合，它是 X 的所有子集组成的集合。例如，如果我们有集合 $X = \{1, 2, 3\}$ ，那么一共有8个子集：

$$\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}.$$

X 的幂集写作 $\mathcal{P}(X)$ ，它就是包含这8个子集的集合。

我们应该如何计算幂集的大小？因为我们是在考察所有可能的子集，所以看起来 X 的幂集总是会比 X 本身更大。事实上，幂集的大小总是等于2的 X 的大小次方。在前面的例子中， $X = \{1, 2, 3\}$ 的大小是3。 X 的幂集有8个元素，也就是 $8 = 2^3$ 。这个规律一般都成立：大小为4的集合，其幂集大小为 $2^4 = 16$ ；大小为1000的集合，其幂集大小为 2^{1000} 。

3.3.1 如何计算给定大小的子集

从某种意义上说，选取一个特定大小的子集，就像是从 n 个选择中选出 r 个元素。集合不关心顺序，所以实际上我们可以使用一般规则：如果集合 X 的大小为 n ，那么 X 中大小为 r 的子集数量由公式 $C_{n,r}$ 给出。如果我们把所有可能的 r 所对应的 $C_{n,r}$ 全部加起来，就会得到 2^n 。一旦我们沿着帕斯卡三角形的行求和，这一点就会变得更清楚。

$$\begin{array}{lcl} \text{第0行:} & & 1 = 1 \\ \text{第1行:} & & 1 + 1 = 2 \\ \text{第2行:} & & 1 + 2 + 1 = 4 \\ \text{第3行:} & & 1 + 3 + 3 + 1 = 8 \\ \text{第4行:} & & 1 + 4 + 6 + 4 + 1 = 16 \\ \text{第5行:} & & 1 + 5 + 10 + 10 + 5 + 1 = 32 \\ & \vdots & \vdots \end{array}$$

正如你所见，帕斯卡三角形每一行的和给出了逐渐增大的2的幂，而这正是描述幂集大小的一种方式。总结一下，给定一个大小为 n 的集合 X ，

- X 中大小为 r 的子集数量由 $C_{n,r}$ 给出。它可以通过查看第 n 行中的第 r 项，直接从帕斯卡三角形中读出。¹
- X 的所有子集总数由 2^n 给出。它可以通过把帕斯卡三角形第 n 行中出现的所有项相加来计算。

例

考虑集合 $X = \{a, b, c, d\}$ ，它是一个大小为4的集合。我们可以使用 $C_{4,r}$ 来计算大小为 r 的所有子集，其中 r 从0到4取值。

- $C_{4,0} = 1$ 计算的是唯一一个大小为0的子集，也就是空集 $\emptyset$ 。
- $C_{4,1} = 4$ 计算的是大小为1的子集，也就是集合 $\{a\}$ 、 $\{b\}$ 、 $\{c\}$ 和 $\{d\}$ 。
- $C_{4,2} = 6$ 计算的是大小为2的子集，也就是集合 $\{a, b\}$ 、 $\{a, c\}$ 、 $\{a, d\}$ 、 $\{b, c\}$ 、 $\{b, d\}$ 和 $\{c, d\}$ 。
- $C_{4,3} = 4$ 计算的是大小为3的子集，也就是集合 $\{a, b, c\}$ 、 $\{a, b, d\}$ 、 $\{a, c, d\}$ 和 $\{b, c, d\}$ 。
- $C_{4,4} = 1$ 计算的是唯一一个大小为4的子集，也就是整个集合 X 本身。

请注意，所有 $C_{4,r}$ 项的和为： $1 + 4 + 6 + 4 + 1 = 16$ ，这正好等于 2^4 。

3.4 走向无穷，以及更远！

我们现在要讨论一个特殊的東西：无穷。到目前为止，我们一直在非常简单的层面上讨论集合的大小。例如， $X = \{a, b, c\}$ 是大小为3的集合，或者 $Y = \{2, 3, 5\}$ 也是大小为3的集合。现在我们要考虑所有自然数构成的集合，也就是集合

$$\mathbb{N} = \{0, 1, 2, 3, \dots\}.$$

显然，这个集合不同于任何有限集合，因为它在某种意义上“无限大”。这不应该太有争议：显然集合 $\mathbb{N}$ 不可能有有限大小，因为它包含所有自然数，所以 $\mathbb{N}$ 的大小必须比任何有限数都大。

现在考虑下面的问题： $\mathbb{N} = \{0, 1, 2, 3, \dots\}$ 的幂集怎么样？它有多大？

答案很奇怪，但它是正确的： $\mathbb{N}$ 的幂集，写作 $\mathcal{P}(\mathbb{N})$ ，也是无限大的。但是，它在大小上确实比 $\mathbb{N}$ 本身更大，而且我们可以用数学证明这一点。这个论证其实相当简单，所以我现在把它重复一遍。

3.4.1 对角线论证

假设 $\mathbb{N}$ 和它的幂集其实是一样大的。这意味着，如果我取这两个集合中的内容，并把它们排成一列，那么我可以把这两个集合中的内容一一配对。于是我们这样做：回忆 $\mathbb{N} = \{0, 1, 2, 3, \dots\}$ ，然后把它的幂集写成：

$$\mathcal{P}(\mathbb{N}) = \{A_0, A_1, A_2, A_3, \dots\}.$$

每一个 A_n 都是 $\mathbb{N}$ 的子集，所以它本身也是一个包含一堆数字的集合。因此，我们可以逐一查看 $\mathbb{N}$ 中的数字，并问：“这个特定数字是否属于这个特定子集 A_n ？”答案总是是或否。这意味着我们可

¹这里，我们从0开始数行，也就是说，第 n 行有 $(n+1)$ 个数字，对应 $r = 0, 1, 2, \dots, n-1, n$ 。

以创建一个巨大的表格，用来记录每个子集 A_n 中包含哪些数字。为了说明，假设

$$\begin{aligned} A_0 &= \{2, 3, 4\} \\ A_1 &= \{0, 2, 4, 6, 8, \dots\} \\ A_2 &= \{1, 2, 3, 9, 27, 1000\} \\ A_3 &= \{3\} \\ A_4 &= \emptyset \\ &\vdots \end{aligned}$$

那么，我们的表格会像这样：

	0	1	2	3	4	...
A_0	×	×	✓	✓	✓	...
A_1	✓	×	✓	×	✓	...
A_2	×	✓	✓	✓	×	...
A_3	×	×	×	✓	×	...
A_4	×	×	×	×	×	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

现在，神奇的地方来了：我们要创造一个新的 $\mathbb{N}$ 的子集 B ，而它不在这个表格中。因为 B 是 $\mathbb{N}$ 的一个子集，所以我们将通过逐个指定它包含哪些数字来定义它。首先，我们看第一行第一列。如果答案是“否”，那就意味着0 不在子集 A_0 中。因此，我们决定把“0”放入新的子集 B 中。所以， B 与 A_0 不同，因为 A_0 不包含0，而 B 包含0。现在，我们向下一行并向右一列移动。同样，这一格中是“否”，也就是说数字1 不在子集 A_1 中。因此，我们也故意把数字1 放入集合 B 。这意味着，按照构造， B 与 A_1 不同，因为它们并不包含完全相同的数字。再向下移动一格并向右移动，我们看到2 确实属于子集 A_2 。所以，我们故意把数字2 排除在 B 之外。这保证了 B 也不同于 A_2 。

我们可以沿着表格的对角线无限重复这个过程，并不断向 B 中加入或排除元素，从而强迫 B 与每一个 $A_0, A_1, A_2, \dots$ 都不同。结果就是一个新的子集 B ，它最终不同于表格中的每一个子集。

	0	1	2	3	4	...
A_0	×	×	✓	✓	✓	...
A_1	✓	×	✓	×	✓	...
A_2	×	✓	✓	✓	×	...
A_3	×	×	×	✓	×	...
A_4	×	×	×	×	×	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

	0	1	2	3	4	...
B	✓	✓	×	×	✓	...

但是，列表 $A_0, A_1, A_2, \dots$ 原本应该包含 $\mathbb{N}$ 的所有子集！这怎么可能呢？答案是：这是不可能的。因此，我们最初关于 $\mathbb{N}$ 可以与 $\mathcal{P}(\mathbb{N})$ 一一对应的假设必定是错误的。换句话说： $\mathcal{P}(\mathbb{N})$ 严格大于 $\mathbb{N}$ 。我们刚刚证明了，确实存在无限大的对象，它们字面意义上比其他无限大的对象还要大。