生成函数

生成函数在离散数学中是其中最令人惊讶的、有益的、巧妙的发明。 大体来说，生成函数变换序列问题为函数问题。 这是好的，因为我们有一堆的数学机械操纵功能。 由于生成函数，我们可以适用于所有关于序列的机械问题。 这样，我们就可以用生成函数来解决各种计数问题。 有大量的关于的生成函数的数学，因此我们才只是品尝一下这门学科的问道而已。

在这个演讲中， 我们都会把序列放在尖括号中来更清楚地区别于其他许多出现的数学表达式。

1 生成函数

为无穷序列 $\langle g_0, g_1, g_2, g_3 \rangle$ 的一般生成函数是正式的幂级数:

$$G(x) = g_0 + g_1 x + g_2 x^2 + g_3 x^3 + \cdots$$

生成函数是一个"正式"幂级数在某种意义上说，我们通常把 x 为一个占位符而非一个数字。 只有在极少数情况下，我们将令 x 是实际数和实际求值生成函数 因此，我们基本上可以忘却收敛的问题。 并不是所有的生成函数，是普通的，但只有这些是我们将在这里考虑。

整个讲座中，我们将用双向箭头显示序列和及其生成函数的对应关系:

$$\langle g_0, g_1, g_2, g_3, \ldots \rangle \longleftrightarrow g_0 + g_1 x + g_2 x^2 + g_3 x^3 + \cdots$$

举例来说，这里有一些序列以及其生成函数:

$$\begin{aligned}
\langle 0,0,0,0,\ldots \rangle &\longleftrightarrow 0 + 0x + 0x^2 + 0x^3 + \cdots = 0 \\
\langle 1,0,0,0,\ldots \rangle &\longleftrightarrow 1 + 0x + 0x^2 + 0x^3 + \cdots = 1 \\
\langle 3,2,1,0,\ldots \rangle &\longleftrightarrow 3 + 2x + 1x^2 + 0x^3 + \cdots = 3 + 2x + x^2
\end{aligned}$$

模式很简单: 在序列中的第 i 个项(索引从 0)的是在生成函数中的 x^i 的系数。

回忆无限几何级数的和是:

$$1 + z + z^2 + z^3 + \cdots = \frac{1}{1-z}$$

这个方程当|z|≥1 的时候不成立，但我们再次表示，不会担心收敛问题。 这个公式给出了对于整个序列范围的闭形(closed-form)生成函数。 举例来说:

$$\begin{aligned}
\langle 1,1,1,1,\ldots \rangle &\longleftrightarrow 1 + x + x^2 + x^3 + \cdots &= \frac{1}{1-x} \\
\langle 1,-1,1,-1,\ldots \rangle &\longleftrightarrow 1 - x + x^2 - x^3 + x^4 - \cdots &= \frac{1}{1+x} \\
\langle 1,a,a^2,a^3,\ldots \rangle &\longleftrightarrow 1 + ax + a^2x^2 + a^3x^3 + \cdots &= \frac{1}{1-ax} \\
\langle 1,0,1,0,1,0,\ldots \rangle &\longleftrightarrow 1 + x^2 + x^4 + x^6 + \cdots &= \frac{1}{1-x^2}
\end{aligned}$$

2 在生成函数上的操作

生成函数的神奇是，我们可以通过在序列相关的生成函数上进行数学操作来实现对序列的各种操作。 让我们根据序列实验的各种操作和刻画它们的影响。

2.1 缩放

以一个常数乘以的生成函数,缩放和序列相关的每一项相同的常数倍。 举例来说，我们在上文已经指出:

$$\langle 1,0,1,0,1,0,\ldots\rangle \longleftrightarrow 1+x^2+x^4+x^6+\cdots=\frac{1}{1-x^2}$$

把生成函数乘以 2 给出

$$\frac{2}{1-x^2}=2+2x^2+2x^4+2x^6+\cdots$$

它生成序列：

<2，0，2，0，2，0，…>

规则 1（缩放规则） 如果

$$\langle f_0,f_1,f_2,\ldots\rangle \longleftrightarrow F(x),$$

那么,

$$\langle cf_0,\ cf_1,\ cf_2,\ \ldots\rangle \longleftrightarrow c\cdot F(x).$$

生成函数

证明。

$$\begin{aligned}\langle cf_0, cf_1, cf_2, \ldots\rangle &\longleftrightarrow cf_0 + cf_1x + cf_2x^2 + \cdots \\ &= c\cdot(f_0 + f_1x + f_2x^2 + \cdots) \\ &= cF(x)\end{aligned}$$

2．2 加法

生成函数加法，相当于一项一项对两个序列做加法。 举例来说，求和我们早期的两个生成函数，给出：

$$\begin{array}{rlllllllll} & \langle\ 1, & 1, & 1, & 1, & 1, & 1, & \ldots\ \rangle & \longleftrightarrow & \dfrac{1}{1-x} \\ + & \langle\ 1, & -1, & 1, & -1, & 1, & -1, & \ldots\ \rangle & \longleftrightarrow & \dfrac{1}{1+x} \\ \hline & \langle\ 2, & 0, & 2, & 0, & 2, & 0, & \ldots\ \rangle & \longleftrightarrow & \dfrac{1}{1-x}+\dfrac{1}{1+x}\end{array}$$

我们已经现在导出两个不同的表达式，它们都生成序列<2，0，2，0，…>。不惊奇，它们是相等的：

$$\frac{1}{1-x}+\frac{1}{1+x}=\frac{(1+x)+(1-x)}{(1-x)(1+x)}=\frac{2}{1-x^2}$$

规则 2（求和规则）如果

$$\begin{aligned}\langle f_0, f_1, f_2, \ldots\rangle &\longleftrightarrow F(x), \\ \langle g_0, g_1, g_2, \ldots\rangle &\longleftrightarrow G(x),\end{aligned}$$

那　么

$$\langle f_0+g_0,\ f_1+g_1,\ f_2+g_2,\ \ldots\rangle \longleftrightarrow F(x)+G(x).$$

证明：

$$
\begin{aligned}
\langle f_0+g_0,\ f_1+g_1,\ f_2+g_2,\ \ldots\rangle &\longleftrightarrow \sum_{n=0}^{\infty}(f_n+g_n)x^n \\
&= \left(\sum_{n=0}^{\infty} f_n x^n\right)+\left(\sum_{n=0}^{\infty} g_n x^n\right) \\
&= F(x)+G(x)
\end{aligned}
$$

$\sqcap$

2.3 右移

让我们再一次从简单序列开始，且它的生成函数是：

$$\langle 1,1,1,1,\ldots\rangle \longleftrightarrow \frac{1}{1-x}$$

现在让我们通过增加 k 个前导 0 右移序列

$$
\begin{aligned}
\langle\underbrace{0,0,\ldots,0}_{k\text{ zeroes}},1,1,1,\ldots\rangle &\longleftrightarrow x^k+x^{k+1}+x^{k+2}+x^{k+3}+\cdots \\
&= x^k\cdot(1+x+x^2+x^3+\cdots) \\
&= \frac{x^k}{1-x}
\end{aligned}
$$

显然，往序列中添加 k 个前导 0 相当于把生成函数乘以 x^k。通常情况下这也是真的。

规则 3（右移规则）

如果 $\langle f_0,f_1,f_2,\ldots\rangle \longleftrightarrow F(x)$，那么

$$\langle\underbrace{0,0,\ldots,0}_{k\ zeroes},f_0,f_1,f_2,\ldots\rangle \longleftrightarrow x^k\cdot F(x)$$

证明：

$$
\begin{aligned}
\langle \overbrace{0,0,\ldots,0}^{k \text{ zeroes}}, f_0, f_1, f_2, \ldots \rangle \quad &\longleftrightarrow \quad f_0 x^k + f_1 x^{k+1} + f_2 x^{k+2} + \cdots \\
&= \quad x^k \cdot (f_0 + f_1 x + f_2 x^2 + f_3 x^3 + \cdots) \\
&= \quad x^k \cdot F(x)
\end{aligned}
$$

⌋

2.4 微分

如果我们对生成函数求导会发生什么? 作为一个例子，让我们对现在人们熟悉的为无限序列 1 的生成函数。

$$
\begin{aligned}
\frac{d}{dx}\left(1 + x + x^2 + x^3 + x^4 + \cdots\right) \quad &= \quad \frac{d}{dx}\left(\frac{1}{1-x}\right) \\
1 + 2x + 3x^2 + 4x^3 + \cdots \quad &= \quad \frac{1}{(1-x)^2} \\
\langle 1, 2, 3, 4, \ldots \rangle \quad &\longleftrightarrow \quad \frac{1}{(1-x)^2}
\end{aligned}
$$

我们发现了一个序列<1，2，3，4 ，。。。>的生成函数!。

一般来说，微分生成函数对应的序列有两个影响: 每项被它的索引倍乘，整个序列向左移动一个位置。

规则 4（求导规则）如果

$$\langle f_0, f_1, f_2, f_3, \ldots \rangle \longleftrightarrow F(x),$$

那么

$$\langle f_1, 2f_2, 3f_3, \ldots \rangle \longleftrightarrow F'(x).$$

证明：

$$\begin{aligned}\langle f_1, 2f_2, 3f_3, \ldots\rangle &= f_1 + 2f_2x + 3f_3x^2 + \cdots \\ &= \frac{d}{dx}\left(f_0 + f_1x + f_2x^2 + f_3x^3 + \cdots\right) \\ &= \frac{d}{dx}F(x)\end{aligned}$$

□

求导规则，是非常有用的。 事实上，有经常的，独立的，需要每个微分的两个效果， 用它们的指数乘以项，左移一个位置。 一般来说，我们只想要一个效果，并一定会取消另一个。 举例来说，让我试试看找到序列<0，1，4，9，16，…>的生成函数。如果我们开始按照顺序<1，1，1，1 ，……>再乘以每项由其指数两倍，然后我们就会有理想的效果:

$$\langle 0\cdot 0,\ 1\cdot 1,\ 2\cdot 2,\ 3\cdot 3,\ \ldots\rangle = \langle 0,1,4,9,\ldots\rangle$$

一个挑战是微分不仅由其指数乘以每项，而且左移整个序列一个位置。然而， 右移规则 3 告诉如何来取消不必要的左移：用 x 乘以生成函数。

我们的程序，因此，将从序列是<1，1，1，1 ，…>的生成函数开始，微分，乘以 x， 然后微分和乘以 x 一次。

$$\langle 1,1,1,1,\ldots\rangle \longleftrightarrow \frac{1}{1-x}$$

$$\langle 1,2,3,4,\ldots\rangle \longleftrightarrow \frac{d}{dx}\frac{1}{1-x} = \frac{1}{(1-x)^2}$$

$$\langle 0,1,2,3,\ldots\rangle \longleftrightarrow x\cdot\frac{1}{(1-x)^2} = \frac{x}{(1-x)^2}$$

$$\langle 1,4,9,16,\ldots\rangle \longleftrightarrow \frac{d}{dx}\frac{x}{(1-x)^2} = \frac{1+x}{(1-x)^3}$$

$$\langle 0,1,4,9,\ldots\rangle \longleftrightarrow x\cdot\frac{1+x}{(1-x)^3} = \frac{x(1+x)}{(1-x)^3}$$

因此，对于平方的生成函数是：

$$\frac{x(1+x)}{(1-x)^3}$$

3 Fibonacci 序列

有时候，我们可以找到更复杂序列的生成函数，对于较复杂的序列。 举例来说，这里是一个为 Fibonacci 数生成函数:

$$\langle 0,1,1,2,3,5,8,13,21,\ldots\rangle \longleftrightarrow \frac{x}{1-x-x^2}$$

Fibonacci 数是一个相当难看的一群，但生成函数很简单!

我们要推导这个生成函数，然后用它来寻找一个为 n Fibonacci 数的封闭的形式。 当然，我们已经有了一个封闭的形式 fibonacci 数，从快速求解线性递推快速手册过程获得。 但有几项理由来覆盖同一内容理由。 首先，我们将获得一些洞察为什么手册

方法对线性递归可行。 第二， 我们将使用的技术，适用于一大类递推方程，包括一些我们没有其他的方式来处理的方程。

3.1 发现一个生成函数

让我们通过回忆 Fibonacci 数的定义开始：

$$
\begin{aligned}
&f_0 = 0 \\
&f_1 = 1 \\
&f_n = f_{n-1} + f_{n-2} \qquad (\text{for } n \geq 2)
\end{aligned}
$$

我们可以扩大最后语句成为第一个无限序列方程。 因此，Fibonacci 数被定义为：

$$
\begin{aligned}
f_0 &= 0 \\
f_1 &= 1 \\
f_2 &= f_1 + f_0 \\
f_3 &= f_2 + f_1 \\
f_4 &= f_3 + f_2 \\
&\vdots
\end{aligned}
$$

现在总体计划是，确定一个函数 f(x) ，生成在相等符号左侧的序列，其是 Fibonacci 数。 那么我们得出一个函数生成在右侧序列。 最后，我们把两者等同起来，并为求解 f(x)的。 让我试试看。 首先，我们定义：

$$F(x) = f_0 + f_1 x + f_2 x^2 + f_3 x^3 + f_4 x^4 + \cdots$$

现在，我们需要为序列得出一个生成函数：

$$\langle 0,\ 1,\ f_1+f_0,\ f_2+f_1,\ f_3+f_2,\ \ldots\rangle$$

办法之一是把这个打破成我们知道生成函数的三个序列的和，然后应用求和规则：

$$\begin{array}{cccccccccl}
 & \langle & 0, & 1, & 0, & 0, & 0, & \ldots & \rangle & \longleftrightarrow\ x \\
 & \langle & 0, & f_0, & f_1, & f_2, & f_3, & \ldots & \rangle & \longleftrightarrow\ xF(x) \\
+ & \langle & 0, & 0, & f_0, & f_1, & f_2, & \ldots & \rangle & \longleftrightarrow\ x^2F(x) \\
\hline
 & \langle & 0, & 1+f_0, & f_1+f_0, & f_2+f_1, & f_3+f_2, & \ldots & \rangle & \longleftrightarrow\ x+xF(x)+x^2F(x)
\end{array}$$

这一序列几乎是和 Fibonacci 方程右边相同。一个美中不足的是，第二项是 $1+f_0$ 而非单纯 1 。 不过，这不等于什么，因为不管如何 $f_0 = 0$。

现在，如果我们让 $F(x)$ 等于新函数 $x + xF(x)+ x^2f(x)$ 然后我们一下子含蓄地写下来全部公式定义 Fibonacci 数字：

$$\begin{array}{ccccccccc}
F(x) & = & f_0 + & f_1\,x + & f_2\,x^2 + & f_3\,x^3 + & f_4\,x^4 + \ldots \\
\| & & \| & \| & \| & \| & \| \\
x+xF(x)+x^2F(x) & = & 0 + & (1+f_0)\,x + & (f_1+f_0)\,x^2 + & (f_2+f_1)\,x^3 + & (f_3+f_2)\,x^4 + \cdots
\end{array}$$

求解 $f(x)$ 的给出生成函数 Fibonacci 序列：

$$\begin{aligned}
& F(x) = x + xF(x) + x^2F(x) \\
\Rightarrow \qquad & F(x) = \frac{x}{1-x-x^2}
\end{aligned}$$

果然，这是很简单的我们在开始申明的生成函数！

3.2 找到了封闭形式

何必在乎对一序数的生成函数? 有几个答案，但这里是一个:如果我们能够为一个序列找到一个函数， 那么，我们往往可以找到一个第 n 个系数的闭形，这可能是相当有用! 举例来说， 在幂级数中的 x^n 的系数的闭形式在幂级数 $x/(1-x-x^2)$将是一个第 n 个 fibonacci 数明确的公式。

所以我们下一步的工作是从生成函数提取系数。有几种办法。对于生成函数是一个多项式的比例，我们可使用你所学到的计算中的部分分式的方法。 正如部分扩展的项是比较容易积分的， 这些项的系数是容易计算的。

让我们尝试用 fibonacci 数字的生成函数这种方法。 首先，我们因式分解分母:

$$1-x-x^2=(1-\alpha_1 x)(1-\alpha_2 x)$$

这里 $\alpha_1=\frac{1}{2}(1+\sqrt{5})$以及 $\alpha_2=\frac{1}{2}(1-\sqrt{5})$。下一步，我们发现 A_1 和 A_2 它们满足：

$$\frac{x}{1-x-x^2}=\frac{A_1}{1-\alpha_1 x}+\frac{A_2}{1-\alpha_2 x}$$

我们这样通过带入 x 的各种值来产生的在 A_1 和 A_2 中线性方程。 我们便可以求解线性系统找到 A_1 和 A_2。 这给出:

$$A_1 = \frac{1}{\alpha_1 - \alpha_2} = \frac{1}{\sqrt{5}}$$
$$A_2 = \frac{-1}{\alpha_1 - \alpha_2} = -\frac{1}{\sqrt{5}}$$

代入上述公式，给出 F(x)的部分分式扩展函数:

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

每个在部分分式扩张的项有一个简单通过集合和公式给出的幂级数:

$$\frac{1}{1-\alpha_1 x} = 1 + \alpha_1 x + \alpha_1^2 x^2 + \cdots$$
$$\frac{1}{1-\alpha_2 x} = 1 + \alpha_2 x + \alpha_2^2 x^2 + \cdots$$

代入这些级数给出幂级数生成函数:

$$\begin{aligned} F(x) &= \frac{1}{\sqrt{5}}\left(\frac{1}{1-\alpha_1 x} - \frac{1}{1-\alpha_2 x}\right) \\ &= \frac{1}{\sqrt{5}}\left((1 + \alpha_1 x + \alpha_1^2 x^2 + \cdots) - (1 + \alpha_2 x + \alpha_2^2 x^2 + \cdots)\right) \\ \Rightarrow f_n &= \frac{\alpha_1^n - \alpha_2^n}{\sqrt{5}} \\ &= \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right) \end{aligned}$$

我们发现，其利用求解线性递推，这是第 n 个 fibonacci 数同样可怕的公式。 另一种办法遮挡了在这方法之上的法光芒。 尤其，涉及重复根的特征方程奇怪的规则是寻找部分因式扩展的规则。

4 使用生成函数计数

生成函数对于求解计数问题特别有用。 尤其是所涉及从集合中选项问题导致好的生成函数。 当生成的函数用在这种方式中， x^n 的系数是选择 n 项的方式。

4．1 从一个集合中选择不同的项

从一个集生成函数为二项式系数后，直接从二项式定理：

$$\left\langle \binom{k}{0}, \binom{k}{1}, \binom{k}{2}, \ldots, \binom{k}{k}, 0, 0, 0, \ldots \right\rangle \longleftrightarrow \binom{k}{0} + \binom{k}{1}x + \binom{k}{2}x^2 + \cdots + \binom{k}{k}x^k$$
$$= (1+x)^k$$

因此，$(1+x)^k$ 中的系数式 x^n 是从 k 元素集中选择 n 个不同的项。 举例来说，x^2 的系数为 $\binom{k}{2}$，从 k 个元素集中选择 2 个元素的方法。同样，系数 x^{k+1} 式从 k 个元素集合中选择 k+1 项的方式，其是零。

4.2 构建计数的生成函数

有时候，我们可以把计数问题直翻译为生成函数问题来求解。 举例来说， 我们可以计算出：$(1+x)^k$ 的生成多种方式从 k 元素的子集来选择 n 个不同的项，而甚至不需要借助二项式定理或者甚至过分追究二项式系数！

这里是如何。 首先，考虑单个元素集和{a_1} 。从这集合为多种方式选择 n 个元素生成函数简单的式 1 + x ： 我们有 1 方式选择零个元素，一种方式来选择的一个元素，0 种方式选择一个以上的元素。同样，从集合{a_2}中选择 n 个元素的方法数目，也是由生成函数 1 +x 给出的。在这两种情况中元素不同的事实是无关紧要。

现在这里是主要的技巧:从不相交集合中选择元素的生成函数式从每个集合选择元素的生成函数乘积。 我们将很快证明这个，但让我们先看看一个例子。 根据这个原理，从{a_1,a_2}中选择 n 个元素的生成函数式：

$$\underbrace{(1+x)}_{\substack{\text{OGF for}\\ \{a_1\}}} \cdot \underbrace{(1+x)}_{\substack{\text{OGF for}\\ \{a_2\}}} = \underbrace{(1+x)^2}_{\substack{\text{OGF for}\\ \{a_1,a_2\}}} = 1+2x+x^2$$

果然，集合{a_1 ， a_2 } ，我们有 1 方式选择零个元素， 2 种方式选择的一个元素， 0 种方式选择两个以上的元素。

反复应用这个规则给出从 k 元素集合{a_1 ， a_2 ，…， a_k }选择 k 个元素的生成函数 ：

$$\underbrace{(1+x)}_{\substack{\text{OGF for}\\ \{a_1\}}} \cdot \underbrace{(1+x)}_{\substack{\text{OGF for}\\ \{a_2\}}} \cdots \underbrace{(1+x)}_{\substack{\text{OGF for}\\ \{a_k\}}} = \underbrace{(1+x)^k}_{\substack{\text{OGF for}\\ \{a_1,a_2,\ldots,a_k\}}}$$

这是同样的生成函数，通过我们用二项式定理得到。 但这次我们直接从计数问题翻译为生成函数。

我们可以扩大这些想法为一个总原则：

规则 5 （卷积规则） 设 a （ x ）是从集合 A 选择元素的生成函数，令 b （ x ）是从集合 B 选择元素的生成函数。如果 A 和 B 是不相交的， 从并 A∪B 中选择元素的生成函数是乘积 A(x)·B(x)。

这项规定相当含糊：规则到底控制什么来从集合选择元素？ 不可思议的是，在许多不同的解释选择中卷积规则仍然的有效的。 举例来说， 我们可以坚持选定不同元素，或者我们可以让同一元素选择有限次数或者任何次数。非正式的，唯一的限制是：(1)元素被选择的顺序是忽略的 (2)对从 A 和 B 选择元素的限制也应用在从 A∪B 中选择元素。(正式的说， 必须有一个在从 A∪B 中选择 n 个元素和从 A 和 B 包括总共 n 个元素的有序对的选择的双射)。

证明：定义：

$$A(x) = \sum_{n=0}^{\infty} a_n x^n, \qquad B(x) = \sum_{n=0}^{\infty} b_n x^n, \qquad C(x) = A(x) \cdot B(x) = \sum_{n=0}^{\infty} c_n x^n.$$

让我们首先求值 A(x)·B(x)和表达系数 c_n 中计了—根据 a 和 b 的系数。 我们可以为所有在乘积中的元素制表：

	b_0x^0	b_1x^1	b_2x^2	b_3x^3	$\dots$
a_0x^0	$a_0b_0x^0$	$a_0b_1x^1$	$a_0b_2x^2$	$a_0b_3x^3$	$\dots$
a_1x^1	$a_1b_0x^1$	$a_1b_1x^2$	$a_1b_2x^3$	$\dots$	
a_2x^2	$a_2b_0x^2$	$a_2b_1x^3$	$\dots$		
a_3x^3	$a_3b_0x^3$	$\dots$			
$\vdots$	$\dots$				

注意到，所有的元素涉及相同的在一个/-斜的对角线 x 的幂。 收集这些元素一起，我们在乘积中发现系数式 x^n 中的是:

$$c_n = a_0b_n + a_1b_{n-1} + a_2b_{n-2} + \cdots + a_nb_0$$

现在我们必须表明,这也是多种方式从 A∪B 中选择 n 个元素。一般来说， 我们可以从 A∪B 选择一个总的 n 项，通过从 A 选择 j 元素，从 B 选择 j 个元素。这个能在 a_jb_{n-j} 方式完成。综合所有的可能的 j 的值给出总的：

$$a_0b_n + a_1b_{n-1} + a_2b_{n-2} + \cdots + a_nb_0$$

方式来从 A∪B 中选择 n 个元素。这也是以上 c_n 的精确的值。

表达式 $c_n = a_0b_n + a_1b_{n-1} + a_2b_{n-2} + \cdots + a_nb_0$ 可信号处理课程中熟悉； 序数 $\langle c_0, c_1, c_2, \dots \rangle$ 是 $\langle a_0, a_1, a_2, \dots \rangle$ 和 $\langle b_0, b_1, b_2, \dots \rangle$ 的卷积。

4.3 选择重复的元素

我们首先考虑的计数问题要求计算选择一打甜甜圈的方法，当五个可变的种类可用的时候。我们能按照如下一般化这个问题：从 n 个元素的集合中有多少种方式选择 k 个元素，如果我们允许取多个同样的项。在这些项种，甜甜圈问题问我们从集合中选择一打甜甜圈的方式有多少种：

{巧克力，柠檬馅，糖的，光亮的，平的}

如果我们允许取多个同一种甜甜圈。让我们从生成函数的角度求解这个问题。

假设我们从包含一个元素的集合中选择 n 个元素(允许重复)。那么，有一种方式选择 0 个项，一种方式选择一项，另一种方式选择两项，等等。因此，从 1 个元素的集合选择 n 个可重复的元素的生成函数是：

$$
\begin{aligned}
\langle 1, 1, 1, 1, \ldots \rangle \quad &\longleftrightarrow \quad 1 + x + x^2 + x^3 + \cdots \\
&= \quad \frac{1}{1-x}
\end{aligned}
$$

卷积规则说从不相交的集合中选择项的生长函数是从每个集合中选择项的生成函数：

$$
\underbrace{\frac{1}{1-x}}_{\substack{\text{OGF for} \\ \{a_1\}}} \cdot \underbrace{\frac{1}{1-x}}_{\substack{\text{OGF for} \\ \{a_2\}}} \cdots \underbrace{\frac{1}{1-x}}_{\substack{\text{OGF for} \\ \{a_n\}}} = \underbrace{\frac{1}{(1-x)^n}}_{\substack{\text{OGF for} \\ \{a_1, a_2, \ldots, a_n\}}}
$$

因此,从n个元素中选择可重复元素的生成函数允是$1/(1-x)^n$。

现在我们需要寻找生成函数的系数。我们能够尝试使用部分分式，但是$(1-x)^n$有一个讨厌的重根1。一种可选的方法式使用泰勒定理。

定理1 （泰勒定理）

$$f(x) - f(0) - f'(0)x - \frac{f''(0)}{2!}x^2 + \frac{f'''(0)}{3!}x^3 + \cdots - \frac{f^{(k)}(0)}{k!}x^k - \cdots$$

这个定理说$1/(1-x)^n$的第k个系数等于它的第k此微分为0且能被k!整除。计算第k个微分，变为非常困难。令

$$g(x) = \frac{1}{(1-x)^n} = (1-x)^{-n}$$

那么我有：

$$\begin{aligned}
G'(x) &= n(1-x)^{-n-1} \\
G''(x) &= n(n+1)(1-x)^{-n-2} \\
G'''(x) &= n(n+1)(n+2)(1-x)^{-n-3} \\
G^{(k)}(x) &= n(n+1)\cdots(n+k-1)(1-x)^{-n-k}
\end{aligned}$$

因此，在生成函数的系中的x^k的系数式：

$$
\begin{aligned}
G^{(k)}(0)/k! &= \frac{n(n+1)\cdots(n+k-1)}{k!} \\
&= \frac{(n+k-1)!}{(n-1)!\,k!} \\
&= \binom{n+k-1}{k}
\end{aligned}
$$

因此，从 n 个元素中选择 k 个可重复的元素的方式是：

$$\binom{n+k-1}{k}$$

这很有道理,因为存在一个选择和(n+k-1)有 k 个 0 和 n-1 个 1(分割 n 个项的不同类型)的 bit 序列之间的双射。

5 一个“可能”的计数问题

到目前为止，我们已经用了我们可能用作另一种方式的生成函数。但是这里是一个模糊的计数问题——真正的在顶部!受到以下限制有多少种方式我们能用 n 个水果填充一个包？

苹果的数目必须是偶数。

香蕉的数目必须是 5 的倍数。

至少有 4 个桔子。

至多有 1 个梨。

例如，存在 7 种方式形成一个有 6 种水果的包：

Apples	6	4	4	2	2	0	0
Bananas	0	0	0	0	0	5	5
Oranges	0	2	1	4	3	1	0
Pears	0	0	1	0	1	0	1

这些限制如此复杂，求解问题看起来没有希望了！但是让我们看生成函数是否能发现什么。

让我们首先构造选择苹果的生成函数。我们能选择 0 苹果，1 种方式，0 种方式选择 1 个苹果（因为苹果的数目必须是偶数），一种方式选择 2 个苹果，0 种方式选择 3 个苹果，等等。因此，我们有：

$$A(x) = 1 + x^2 + x^4 + x^6 + \cdots = \frac{1}{1 - x^2}$$

同样，选择香蕉的生成函数是：

$$B(x) = 1 + x^5 + x^{10} + x^{15} + \cdots = \frac{1}{1 - x^5}$$

现在我们能一种方式选择 0 个桔子，一种方式选择 1 个桔子，等等。然而，我们不能选择超过 4 个桔子，因此我们有生成函数：

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

这里我们使用了几何和公式。最后，我们能仅可以选择 0 或者 1 个梨。因此我们有：

$$P(x) = 1 + x$$

卷积规则说在 4 种水果种选择的生成函数是：

$$\begin{aligned} A(x)B(x)O(x)P(x) &= \frac{1}{1-x^2}\frac{1}{1-x^5}\frac{1-x^5}{1-x}(1+x) \\ &= \frac{1}{(1-x)^2} \\ &= 1 + 2x + 3x^2 + 4x^3 + \cdots \end{aligned}$$

几乎所有的都削掉了！我们剩下 $1/(1-x)^2$，我们发现早期的幂级数：x^n 的系数是简单的 n+1。因此形成一个 n 个水果的包的方式的数目是 n +1。这和我们计算出的例子是一致的，因为存在 7 个不同的包括 6 个水果的包！很奇妙！