

组合计数和生成函数

x义x

HZXJHS

YYYY/MM/DD

头像的天线少了一根，不要在意（



目录

0. 引言

前置知识
声明和警告

1. 定义

组合类和 OGF
形式幂级数
基础组合类

2. 构造组合类

笛卡尔积
和

Sequence 构造
插入一点群论
Cycle 构造
Multiset 构造

3. Pólya 定理 定义

内容
特殊情形

4. 例题和习题

神秘例题 1
神秘例题 2

ω. 结语和感谢

结语
感谢

引言

“生成函数”是组合对象在度量上的投影。

——李白天

引言

“生成函数”是组合对象在度量上的投影。

——李白天

我也不知道 EI 从哪里摘来的，所以就当是 EI 说的吧

引言

“生成函数”是组合对象在度量上的投影。

——李白天

我也不知道 EI 从哪里摘来的，所以就当是 EI 说的吧

生成函数是一种研究组合对象的有力工具。它可以把组合对象的“结构”表示为代数运算，从而大大有利于我们分析组合对象。

引言

“生成函数”是组合对象在度量上的投影。

——李白天

我也不知道 EI 从哪里摘来的，所以就当是 EI 说的吧

生成函数是一种研究组合对象的有力工具。它可以把组合对象的“结构”表示为代数运算，从而大大有利于我们分析组合对象。

近年来计数题出得很多，ZJOI2018 树，ZJOI2019 开关，ZJOI2020 抽卡，HNOI2019 白兔之舞，WC 2019 数树，CTS2019 珍珠……连 CSP2019 都出现了一道模数 998244353 的题（

引言

“生成函数”是组合对象在度量上的投影。

——李白天

我也不知道 ~~EI~~ 从哪里摘来的，所以就当是 ~~EI~~ 说的吧

生成函数是一种研究组合对象的有力工具。它可以把组合对象的“结构”表示为代数运算，从而大大有利于我们分析组合对象。

近年来计数题出得很多，ZJOI2018 树，ZJOI2019 开关，ZJOI2020 抽卡，HNOI2019 白兔之舞，WC 2019 数树，CTS2019 珍珠……连 CSP2019 都出现了一道模数 998244353 的题（

本课件将要从头介绍一种方法，它可以符号化组合对象，解救选手于组合意义的地狱之中（，甚至可以秒掉一些用通常方法几乎不可能在考场解出的题目。

引言 · 前置知识

- 多项式 \exp 及其前置（不影响听课，但是可能做不了某些例题）
- 分治 FFT（不影响听课，但是可能做不了某些例题）
- 拉格朗日反演（不影响听课，但是可能做不了某些例题）
- 群论，可以看[这里](#)。为了确保你真的理解了它，请切掉 SHOI2006 有色图。
- 你不需要什么计数和生成函数的知识，因为我们会从头开始介绍。

引言 · 组合计数不全是生成函数

首先必须明确，生成函数虽然（如你将要看到的那样）很强大，但是它也有其局限性。最典型的局限性就是它在表达一些复杂结构的时候非常痛苦……比如降序排列之类的。

引言 · 组合计数不全是生成函数

首先必须明确，生成函数虽然（如你将要看到的那样）很强大，但是它也有其局限性。最典型的局限性就是它在表达一些复杂结构的时候非常痛苦……比如降序排列之类的。

举个例子，杨表是一个非常复杂的结构，但是它上面有不少相当精简的结论。用生成函数表达杨表……？别别别那会死人的

引言 · 组合计数不全是生成函数

首先必须明确，生成函数虽然（如你将要看到的那样）很强大，但是它也有其局限性。最典型的局限性就是它在表达一些复杂结构的时候非常痛苦……比如降序排列之类的。

举个例子，杨表是一个非常复杂的结构，但是它上面有不少相当精简的结论。用生成函数表达杨表……？别别别那会死人的

再比如 Cyanic 之前讲的《Combinatorial Proof》也是组合计数里的一个非常重要的方法。

引言 · 组合计数不全是生成函数

首先必须明确，生成函数虽然（如你将要看到的那样）很强大，但是它也有其局限性。最典型的局限性就是它在表达一些复杂结构的时候非常痛苦……比如降序排列之类的。

举个例子，杨表是一个非常复杂的结构，但是它上面有不少相当精简的结论。用生成函数表达杨表……？别别别那会死人的

再比如 Cyanic 之前讲的《Combinatorial Proof》也是组合计数里的一个非常重要的方法。

本课也许可以命名为 ~~Non-Combinatorial Proof~~

引言 · 组合计数不全是生成函数

其实生成函数还有一个弱点（组合意义还有一个优点？），就是有些从组合意义上考虑极为简单的结论用生成函数很难看出。

具体比较难举出例子，做多了就知道了。

（生成函数）蕴含的初等运算恐怕并不足够具有洞察力以支持得到，通过特殊的组合对象得出的结论的水平。

——李白天

引言 · 生成函数不能直接组合计数

如果你搞出了生成函数但不会处理它，那么生成函数一点用都没有。

引言 · 生成函数不能直接组合计数

如果你搞出了生成函数但不会处理它，那么生成函数一点用都没有。

你可能需要例如多项式科技，拉格朗日反演，
整式递推，多元拉格朗日反演之类的处理生成函数的工具。

引言 · 生成函数不能直接组合计数

如果你搞出了生成函数但不会处理它，那么生成函数一点用都没有。

你可能需要例如多项式科技，拉格朗日反演，整式递推，多元拉格朗日反演之类的处理生成函数的工具。

这些处理方法很杂很多，我们今天不讨论它们，只讨论生成函数本身。

定义：组合类和 OGF

我们先明确什么叫做组合类。

定义：组合类和 OGF

我们先明确什么叫做组合类。

一个**组合类**（combinatorial class）是一个可数集合 \mathcal{A} ，里面包含了一些**组合对象**，比如有标号树，烯烃，被染色的环之类的；

以及一个附带的大小函数（size function） $f : \mathcal{A} \rightarrow \mathbb{N}$ 。比如节点数，序列长度等等说的就是大小函数。记 $f(x) = |x|$ ，或者，在某些需要指明组合类的场合下， $|x|_{\mathcal{A}}$ 。

定义：组合类和 OGF

我们先明确什么叫做组合类。

一个**组合类** (combinatorial class) 是一个可数集合 \mathcal{A} ，里面包含了一些**组合对象**，比如有标号树，烯烃，被染色的环之类的；

以及一个附带的大小函数 (size function) $f : \mathcal{A} \rightarrow \mathbb{N}$ 。比如节点数，序列长度等等说的就是大小函数。记 $f(x) = |x|$ ，或者，在某些需要指明组合类的场合下， $|x|_{\mathcal{A}}$ 。

记对于自然数 n ， $A_n = \left| \{x \in \mathcal{A}, |x| = n\} \right|$ 。称 A_n 是 \mathcal{A} 的**计数序列** (counting sequence)。说人话就是大小为 n 的元素有几个。

定义：组合类和 OGF

一个计数序列的 **OGF** (ordinary generating function) 是形式幂级数

$$A(z) = \sum_{i \in \mathbb{N}} A_i z^i$$

因而对于组合类 \mathcal{A} , 它的 OGF 也是

$$\boxed{\sum_{x \in \mathcal{A}} z^{|x|}}$$

我们记

$$[z^n]A = A_n$$

注意 OGF 的第二个形式, 我们将会频繁地使用它。

定义：组合类和 OGF · example 1

假设我们要计数这样的小球序列的个数：长度为 n ，每一位都可以是黑色或白色的小球。

定义：组合类和 OGF · example 1

假设我们要计数这样的小球序列的个数：长度为 n ，每一位都可以是黑色或白色的小球。

那么我们可以研究这样一个组合类：由所有这样的小球序列组成的集合，大小函数定义为序列的长度。

$$\mathcal{A} = \{(), (\circ), (\bullet), (\circ, \circ), (\circ, \bullet), \dots\}$$

定义：组合类和 OGF · example 1

假设我们要计数这样的小球序列的个数：长度为 n ，每一位都可以是黑色或白色的小球。

那么我们可以研究这样一个组合类：由所有这样的小球序列组成的集合，大小函数定义为序列的长度。

$$\mathcal{A} = \{(), (\circ), (\bullet), (\circ, \circ), (\circ, \bullet), \dots\}$$

我们显然有 $[z^n]\mathcal{A} = 2^n$ 。

定义：组合类和 OGF · 什么是形式幂级数

你可以理解为普通的多项式，但是：

定义：组合类和 OGF · 什么是形式幂级数

你可以理解为普通的多项式，但是：

- 允许有无限项

定义：组合类和 OGF · 什么是形式幂级数

你可以理解为普通的多项式，但是：

- 允许有无限项
- 不必考虑收敛性

定义：组合类和 OGF · 什么是形式幂级数

你可以理解为普通的多项式，但是：

- 允许有无限项
- 不必考虑收敛性

事实上形式幂级数里的 z 不应该当成数来理解，它只是一个用来标记的符号。

定义：组合类和 OGF . 一些基础的组合类

记 \mathcal{E} 是只由一个大小为 0 的元素构成的组合类。我们有 $E(z) = 1$ 。

定义：组合类和 OGF . 一些基础的组合类

记 \mathcal{E} 是只由一个大小为 0 的元素构成的组合类。我们有 $E(z) = 1$ 。

记 \mathcal{Z} 是只由一个大小为 1 的元素构成的组合类。我们有 $Z(z) = z$ 。

构造组合类

组合类都是由其他组合类**构造**出来的。一个构造是从一组组合类映射到一个组合类的函数。

构造组合类

组合类都是由其他组合类**构造**出来的。一个构造是从一组组合类映射到一个组合类的函数。

当我们说一个构造可以被直接翻译为生成函数运算，这意味着它可以直接和生成函数运算对应。一个经典的例子是 **exp 的组合意义**。

构造组合类

组合类都是由其他组合类**构造**出来的。一个构造是从一组组合类映射到一个组合类的函数。

当我们说一个构造可以被直接翻译为生成函数运算，这意味着它可以直接和生成函数运算对应。一个经典的例子是 **exp 的组合意义**。

我们下面研究的构造都可以直接翻译为生成函数运算。
不然今天我们就研究不了了

构造组合类 · 笛卡尔积

基础中的基础。

构造组合类 · 笛卡尔积

基础中的基础。

$\mathcal{A} = \mathcal{B} \times \mathcal{C}$ 被称为组合类 \mathcal{B} 和 \mathcal{C} 的**笛卡尔积**，其中 \mathcal{A} 是有序对的集合 $\{(\beta, \gamma) | \beta \in \mathcal{B}, \gamma \in \mathcal{C}\}$ ，大小函数定义为 $|\alpha|_{\mathcal{A}} = |\beta|_{\mathcal{B}} + |\gamma|_{\mathcal{C}}$ 。大小函数的定义还是很自然的，比如你把两棵树拼起来当然是形成了一棵大小为原来两棵树之和的新树。

构造组合类 · 笛卡尔积

基础中的基础。

$\mathcal{A} = \mathcal{B} \times \mathcal{C}$ 被称为组合类 \mathcal{B} 和 \mathcal{C} 的**笛卡尔积**，其中 \mathcal{A} 是有序对的集合 $\{(\beta, \gamma) | \beta \in \mathcal{B}, \gamma \in \mathcal{C}\}$ ，大小函数定义为 $|\alpha|_{\mathcal{A}} = |\beta|_{\mathcal{B}} + |\gamma|_{\mathcal{C}}$ 。大小函数的定义还是很自然的，比如你把两棵树拼起来当然是形成了一棵大小为原来两棵树之和的新树。

此时，

$$A_n = \sum_{i+j=n} B_i C_j$$

也就是普通卷积，从而有

$$A(z) = B(z)C(z)$$

构造组合类 · example 2

回顾 example 1。一个小球序列：

构造组合类 · example 2

回顾 example 1。一个小球序列：

- 可以为空

构造组合类 · example 2

回顾 example 1。一个小球序列：

- 可以为空
- 否则可以看成第一个球和剩下的序列组成的有序对

构造组合类 · example 2

回顾 example 1。一个小球序列：

- 可以为空
- 否则可以看成第一个球和剩下的序列组成的有序对

从而我们可以说

$$\mathcal{A} = \mathcal{E} + \{\circ, \bullet\} \times \mathcal{A}$$

不过我们还需要明确加法。

构造组合类 · 不交并

$\mathcal{A} = \mathcal{B} \cup \mathcal{C}$ ($\mathcal{B} \cap \mathcal{C} = \emptyset$) 被称为 \mathcal{B} 和 \mathcal{C} 的**不交并**, 其元素 α 的大小函数 $|\alpha|_{\mathcal{A}} = \begin{cases} |\alpha|_{\mathcal{B}} & (\alpha \in \mathcal{B}) \\ |\alpha|_{\mathcal{C}} & (\alpha \in \mathcal{C}) \end{cases}$ 。

构造组合类 · 不交并

$\mathcal{A} = \mathcal{B} \cup \mathcal{C}$ ($\mathcal{B} \cap \mathcal{C} = \emptyset$) 被称为 \mathcal{B} 和 \mathcal{C} 的**不交并**, 其元素 α 的大小函数 $|\alpha|_{\mathcal{A}} = \begin{cases} |\alpha|_{\mathcal{B}} & (\alpha \in \mathcal{B}) \\ |\alpha|_{\mathcal{C}} & (\alpha \in \mathcal{C}) \end{cases}$ 。

此时,

$$A_n = B_n + C_n,$$

$$\boxed{A(z) = B(z) + C(z)}$$

构造组合类 · 不交并

$\mathcal{A} = \mathcal{B} \cup \mathcal{C}$ ($\mathcal{B} \cap \mathcal{C} = \emptyset$) 被称为 \mathcal{B} 和 \mathcal{C} 的**不交并**, 其元素 α 的大小函数 $|\alpha|_{\mathcal{A}} = \begin{cases} |\alpha|_{\mathcal{B}} & (\alpha \in \mathcal{B}) \\ |\alpha|_{\mathcal{C}} & (\alpha \in \mathcal{C}) \end{cases}$ 。

此时,

$$A_n = B_n + C_n,$$

$$\boxed{A(z) = B(z) + C(z)}$$

平角裤平角裤! 不支持多重集!

构造组合类 · 和

多重集是好东西，但是我们还是谨慎一点，绕开它吧。我们使用下面的定义：

构造组合类 · 和

多重集是好东西，但是我们还是谨慎一点，绕开它吧。我们使用下面的定义：

$$B + C := (\mathcal{E}_1 \times B) \cup (\mathcal{E}_2 \times C)$$

其中 \mathcal{E}_1 和 \mathcal{E}_2 是两个不同的，仅由一个大小为 0 的对象构成的组合类。

我们有

$$\boxed{\mathcal{A} = B + C \Rightarrow A(z) = B(z) + C(z)}$$

构造组合类 · hint

我们指出，一个足够合理的生成函数的加法必须能对应组合类的和（线性），乘法必须能对应组合的笛卡尔积（积性）。否则它的性质就不太好，没有什么研究的必要。

从而观察笛卡尔积来构造合适的生成函数是一个常用的手段。

构造组合类 · Sequence 构造

若 $[z^0]\mathcal{B} = 0$ ，则我们定义

$$\text{SEQ}(\mathcal{B}) := \mathcal{E} + \mathcal{B} + \mathcal{B} \times \mathcal{B} + \mathcal{B} \times \mathcal{B} \times \mathcal{B} + \dots$$

是 \mathcal{B} 的 Sequence 构造。换句话说，如果 $\mathcal{A} = \text{SEQ}(\mathcal{B})$ ，则

$$\mathcal{A} = \{(\beta_1, \dots, \beta_l) \mid l \geq 0, \beta \in \mathcal{B}\}$$

即 \mathcal{A} 是一些 $\beta \in \mathcal{B}$ 构成的有序列表。

构造组合类 · Sequence 构造

我们有

$$\mathcal{A} = \text{SEQ}(\mathcal{B}) \Rightarrow A(z) = \frac{1}{1 - B(z)}$$

构造组合类 · Sequence 构造

我们有

$$\mathcal{A} = \text{SEQ}(\mathcal{B}) \Rightarrow A(z) = \frac{1}{1 - B(z)}$$

请注意你可以，或者说应当把 $\frac{1}{1 - B(z)}$ 看成
 $1 + B(z) + B^2(z) + B^3(z) + \dots$ 的简写。

构造组合类 · Sequence 构造

我们有

$$\mathcal{A} = \text{SEQ}(\mathcal{B}) \Rightarrow A(z) = \frac{1}{1 - B(z)}$$

请注意你可以，或者说应当把 $\frac{1}{1 - B(z)}$ 看成 $1 + B(z) + B^2(z) + B^3(z) + \dots$ 的简写。

但是它在代数运算下的行为和你平常见到的 $\frac{1}{1 - x}$ 的确一样，
比如说有 $\frac{1 - B(z)}{1 - B(z)} = 1$ 。

构造组合类 · Sequence 构造

我们有

$$\mathcal{A} = \text{SEQ}(\mathcal{B}) \Rightarrow A(z) = \frac{1}{1 - B(z)}$$

请注意你可以，或者说应当把 $\frac{1}{1 - B(z)}$ 看成 $1 + B(z) + B^2(z) + B^3(z) + \dots$ 的简写。

但是它在代数运算下的行为和你平常见到的 $\frac{1}{1 - x}$ 的确一样，
比如说有 $\frac{1 - B(z)}{1 - B(z)} = 1$ 。

不如说，因为它有这些性质所以我们才这么简写。

构造组合类 · final solution for example 1

我们现在可以说, $\mathcal{A} = \text{SEQ}(\{\circ, \bullet\})$ 。

构造组合类 · final solution for example 1

我们现在可以说, $\mathcal{A} = \text{SEQ}(\{\circ, \bullet\})$ 。

显然有 $\{\circ, \bullet\}$ 的 OGF 为 $2z$, 从而

$$A(z) = \frac{1}{1 - 2z}$$

构造组合类 · 插入一点群论

首先定义本质相同是什么意思。

两个元素 β_1, β_2 在映射群列 G_0, G_1, G_2, \dots 下本质相同，如果 $|\beta_1| = |\beta_2|$ ，且存在映射 $g \in G_{|\beta|}$ 使得 $g(\beta_1) = \beta_2$ 。

构造组合类 · 插入一点群论

首先定义本质相同是什么意思。

两个元素 β_1, β_2 在映射群列 G_0, G_1, G_2, \dots 下本质相同，如果 $|\beta_1| = |\beta_2|$ ，且存在映射 $g \in G_{|\beta|}$ 使得 $g(\beta_1) = \beta_2$ 。

定义 \mathcal{A}/G 是一个组合类：其中的元素 α 是一个组合对象的非空集合， α 中的元素互相等价，如果 $\alpha_1 \neq \alpha_2$ ，则 α_1 和 α_2 中的元素两两不等价。我们把 α 称为一个**等价类**（equivalence class）。

构造组合类 · 插入一点群论

首先定义本质相同是什么意思。

两个元素 β_1, β_2 在映射群列 G_0, G_1, G_2, \dots 下本质相同，如果 $|\beta_1| = |\beta_2|$ ，且存在映射 $g \in G_{|\beta|}$ 使得 $g(\beta_1) = \beta_2$ 。

定义 \mathcal{A}/G 是一个组合类：其中的元素 α 是一个组合对象的非空集合， α 中的元素互相等价，如果 $\alpha_1 \neq \alpha_2$ ，则 α_1 和 α_2 中的元素两两不等价。我们把 α 称为一个**等价类**（equivalence class）。

$|\alpha|_{\mathcal{A}/G}$ 等于任意 α 中元素的大小，显然它们都相等。不然不可能等价。

构造组合类 · example 3

我们定义 G_i 是任意 $1..i$ 的排列的置换构成的群。则考虑 example 1 中的 \mathcal{A} , $\mathcal{A}/G =$

$$\begin{aligned} &\{(), \\ &(\circ), (\bullet), \\ &(\circ, \circ), (\circ, \bullet), (\bullet, \bullet), \\ &(\circ, \circ, \circ), (\circ, \circ, \bullet), (\circ, \bullet, \bullet), (\bullet, \bullet, \bullet), \\ &\dots\} \end{aligned}$$

这里我们为了方便, 从 α 中抽出一个代表元素来表示 α 。比如最中间的那个 (\circ, \bullet) 实际表示 $\{(\circ, \bullet), (\bullet, \circ)\}$ 。

构造组合类 · Cycle 构造

若 $[z^0]\mathcal{B} = 0$ ，则我们定义

$$\text{CYC}(\mathcal{B}) := (\text{SEQ}(\mathcal{B}) - \mathcal{E})/S$$

其中 S 是所有循环移位 (circular shift) 构成的群。(就是旋转啦)

构造组合类 · Cycle 构造

若 $[z^0]\mathcal{B} = 0$ ，则我们定义

$$\text{CYC}(\mathcal{B}) := (\text{SEQ}(\mathcal{B}) - \mathcal{E})/S$$

其中 S 是所有循环移位 (circular shift) 构成的群。(就是旋转啦)

根据 Burnside 引理我们容易得到

$$\mathcal{A} = \text{CYC}(\mathcal{B}) \Rightarrow A(z) = \sum_{k=1}^{\infty} \frac{\varphi(k)}{k} \log \frac{1}{1 - B(z^k)}$$

这里不给出证明。

构造组合类 · example 4

$\text{CYC}(\{\circ, \bullet\})$ 就是所有的双色珠子串成的本质不同（可以旋转，不能翻转）的环构成的等价类。

构造组合类 · example 4

$CYC(\{\circ, \bullet\})$ 就是所有的双色珠子串成的本质不同（可以旋转，不能翻转）的环构成的等价类。

不列举了，想必精通群论的您早就知道了

构造组合类 · Multiset 构造

若 $[z^0]\mathcal{B} = 0$, 则我们定义

$$\text{MSET}(\mathcal{B}) = \text{SEQ}(\mathcal{B})/R$$

其中 R 是任意置换构成的群。相当于生成一个 $\beta \in \mathcal{B}$ 组成的无序列表。

构造组合类 · Multiset 构造

若 $[z^0]\mathcal{B} = 0$ ，则我们定义

$$\text{MSET}(\mathcal{B}) = \text{SEQ}(\mathcal{B})/R$$

其中 R 是任意置换构成的群。相当于生成一个 $\beta \in \mathcal{B}$ 组成的无序列表。

在无标号计数里非常重要，所以有诸如 Pólya Exp，欧拉变换等一众别称。

构造组合类 · Multiset 构造

比如, $(\beta_1, \beta_2, \beta_3)$ 和
 $(\beta_1, \beta_3, \beta_2), (\beta_2, \beta_1, \beta_3), (\beta_2, \beta_3, \beta_1), (\beta_3, \beta_1, \beta_2), (\beta_3, \beta_2, \beta_1)$ 等价。
这是整个 $(\beta_1, \beta_2, \beta_3)$ 所属的等价类。

构造组合类 · Multiset 构造

比如, $(\beta_1, \beta_2, \beta_3)$ 和
 $(\beta_1, \beta_3, \beta_2), (\beta_2, \beta_1, \beta_3), (\beta_2, \beta_3, \beta_1), (\beta_3, \beta_1, \beta_2), (\beta_3, \beta_2, \beta_1)$ 等价。
这是整个 $(\beta_1, \beta_2, \beta_3)$ 所属的等价类。

再比如, $(\beta_1, \beta_1, \beta_2)$ 和 $(\beta_1, \beta_2, \beta_1), (\beta_2, \beta_1, \beta_1)$ 等价。这是整个
 $(\beta_1, \beta_1, \beta_2)$ 所属的等价类。

构造组合类 · Multiset 构造

比如, $(\beta_1, \beta_2, \beta_3)$ 和
 $(\beta_1, \beta_3, \beta_2), (\beta_2, \beta_1, \beta_3), (\beta_2, \beta_3, \beta_1), (\beta_3, \beta_1, \beta_2), (\beta_3, \beta_2, \beta_1)$ 等价。
这是整个 $(\beta_1, \beta_2, \beta_3)$ 所属的等价类。

再比如, $(\beta_1, \beta_1, \beta_2)$ 和 $(\beta_1, \beta_2, \beta_1), (\beta_2, \beta_1, \beta_1)$ 等价。这是整个
 $(\beta_1, \beta_1, \beta_2)$ 所属的等价类。

想必精通群论的您早就知道了

构造组合类 · Multiset 构造

有结论：

$$\mathcal{A} = \text{MSET}(\mathcal{B}) \Rightarrow A(z) = \begin{cases} \prod_{i=1}^{\infty} (1 - x^i)^{-B_i} \\ \exp \left(\sum_{i=1}^{\infty} \frac{B(z^i)}{i} \right) \end{cases}$$

其中 $\exp(A)$ 应理解为 $\sum_{i=0}^{\infty} \frac{A^i}{i!}$ 。

构造组合类 · Multiset 构造

有结论：

$$\mathcal{A} = \text{MSET}(\mathcal{B}) \Rightarrow A(z) = \begin{cases} \prod_{i=1}^{\infty} (1 - x^i)^{-B_i} \\ \exp \left(\sum_{i=1}^{\infty} \frac{B(z^i)}{i} \right) \end{cases}$$

其中 $\exp(A)$ 应理解为 $\sum_{i=0}^{\infty} \frac{A^i}{i!}$ 。

第一行从组合意义考虑显然：无序列表可以看成 sort 过的列表，而一个自然的排序是按大小为第一关键字，然后给相同大小的 β 分配 $1..B_i$ 的标号，作为第二关键字。

构造组合类 · Multiset 构造

有结论：

$$\mathcal{A} = \text{MSET}(\mathcal{B}) \Rightarrow A(z) = \left\{ \begin{array}{l} \prod_{i=1}^{\infty} (1 - x^i)^{-B_i} \\ \exp \left(\sum_{i=1}^{\infty} \frac{B(z^i)}{i} \right) \end{array} \right.$$

其中 $\exp(A)$ 应理解为 $\sum_{i=0}^{\infty} \frac{A^i}{i!}$ 。

第一行从组合意义考虑显然：无序列表可以看成 sort 过的列表，而一个自然的排序是按大小为第一关键字，然后给相同大小的 β 分配 $1..B_i$ 的标号，作为第二关键字。

第二行？

构造组合类 · Multiset 构造 · 证明

首先枚举儿子数 i 。考虑应用 burnside 引理，显然一个置换 f 的不动点只与其循环拆分（有序，显然循环间可以通过最前一个元素的位置分出顺序） $\{a_i\}$ 有关。具体来说是

$$\sum_i \frac{1}{i!} \sum_f \prod F(x^{a_i})$$

解释一下上式：循环内所有组的“染色”方案必须完全一样，选择大小均为 s 方案数就只有 $[x^s]F$ ，但是“占地”却是 $a_i s$ ，故有上式。事实上可以直接应用下面会讲到的 Pólya 定理。

构造组合类 · Multiset 构造 · 证明

思考有多少个循环能被循环拆分 $\{a_i\}$ 描述。首先决定每个标号属于哪个循环，然后循环内可以圆排列。

而这样搞会破坏循环原本的自然顺序（自然顺序即最前一个元素的位置）（一个拆分会被恰好计算循环数阶乘次），所以还要再除以循环数的阶乘。

$$\sum_i \frac{1}{i!} \sum_{l, \sum a_j = i} \frac{1}{l!} \binom{i}{a_1, a_2, \dots, a_l} \prod_{j=1}^l F(x^{a_j})(a_j - 1)!$$
$$\sum_i \sum_{l, \sum a_j = i} \frac{1}{l!} \prod_{j=1}^l \frac{F(x^{a_j})}{a_j}$$

构造组合类 · Multiset 构造 · 证明

我们发现现在后面这个部分可以写成 \exp 的形式。设

$$G(x, y) = \sum_{i=1}^{\infty} \frac{F(x^i)y}{i}$$

则我们得到

$$\sum_i [y^i] e^G$$

直接带入 $y = 1$ 即可。我们得到

$$\exp \left(\sum_{i=1}^{\infty} \frac{F(x^i)}{i} \right)$$

可喜可贺!

构造组合类 · example 5

分析无标号有根树的构造。注意允许递归调用自身。

构造组合类 · example 5

分析无标号有根树的构造。注意允许递归调用自身。

一棵树可以表示为根和其儿子列表（由于无标号，我们不能区分儿子的排列顺序，所以这个列表是一个无序列表）组成的有序对（根和儿子还是分清的，毕竟是有根树）。从而我们可以说

$$\mathcal{T} = \mathcal{Z} \times \text{MSET}(\mathcal{T})$$

Pólya 定理

重要的计数工具。

Pólya 定理

重要的计数工具。

但是 Olwiki 上的那个太逊了，我们将要加强它。

Pólya 定理

重要的计数工具。

但是 Olwiki 上的那个太逊了，我们将要加强它。

考虑广义的**染色**。考虑一个大小为 m 的有限集合 \mathcal{M} 和一个置换群 G 作用在其上。不妨认为 $\mathcal{M} = \{1, 2, \dots, m\}$ 。

令 B 是一个组合类， \mathcal{M} 是一个有限集合，上有一个置换群 G 。我们将要研究集合 $B^{\mathcal{M}}$ ，即所有 $\mathcal{M} \rightarrow B$ 的映射，我们称这种映射为**染色**。

染色这个名字的来源是 B 是颜色集合的情形。想必精通群论的您早就知道了

Pólya 定理

重要的计数工具。

但是 Olwiki 上的那个太逊了，我们将要加强它。

考虑广义的**染色**。考虑一个大小为 m 的有限集合 \mathcal{M} 和一个置换群 G 作用在其上。不妨认为 $\mathcal{M} = \{1, 2, \dots, m\}$ 。

令 \mathcal{B} 是一个组合类， \mathcal{M} 是一个有限集合，上有一个置换群 G 。我们将要研究集合 $\mathcal{B}^{\mathcal{M}}$ ，即所有 $\mathcal{M} \rightarrow \mathcal{B}$ 的映射，我们称这种映射为**染色**。

染色这个名字的来源是 \mathcal{B} 是颜色集合的情形。想必精通群论的您早就知道了

之前说的有序列表其实就是染色。

Pólya 定理

两个染色 $\phi_1, \phi_2 \in B^M$ 被认为本质相同, 如果存在映射 $g \in G, \phi_1 \circ g = \phi_2$ 。从而我们染色的等价类集合可以称为 B^M/G 。

Pólya 定理

两个染色 $\phi_1, \phi_2 \in B^M$ 被认为本质相同, 如果存在映射 $g \in G, \phi_1 \circ g = \phi_2$ 。从而我们染色的等价类集合可以称为 B^M/G 。

B 中的每一个元素 β 都可以有权重 $w(\beta)$ 。一个映射 ϕ 的权重 $w(\phi) := \prod_{k \in M} w(\phi(k))$, 显然同一个等价类中的映射权重都一样, 不然不会等价, 称为这个等价类的权重。

Pólya 定理

两个染色 $\phi_1, \phi_2 \in B^M$ 被认为本质相同, 如果存在映射 $g \in G, \phi_1 \circ g = \phi_2$ 。从而我们染色的等价类集合可以称为 B^M/G 。

B 中的每一个元素 β 都可以有权重 $w(\beta)$ 。一个映射 ϕ 的权重 $w(\phi) := \prod_{k \in M} w(\phi(k))$, 显然同一个等价类中的映射权重都一样, 不然不会等价, 称为这个等价类的权重。

注意这个权值可以是一些奇怪的东西, 比如一个幂级数。

Pólya 定理

Pólya 定理的内容如下:

Pólya 定理

Pólya 定理的内容如下:

$$\sum_{\phi \in (\mathcal{B}^M/G)} w(\phi) = Z \left(G; \sum_{\beta \in \mathcal{B}} w(\beta), \dots, \sum_{\beta \in \mathcal{B}} w^m(\beta) \right)$$

其中

$$Z(G; x_1, x_2, \dots, x_m) = \frac{1}{|G|} \sum_{g \in G} x_1^{j_1(g)} x_2^{j_2(g)} \dots x_m^{j_m(g)}$$

其中 $j_1(g)$ 表示 g 中大小为 1 的循环个数。下面是一个 Olwiki 上的 Pólya 定理，以示对比。

$$|X/G| = \frac{1}{|G|} \sum_{g \in G} |B|^{c(g)}, c(g) = \sum j(g)$$

Pólya 定理 · 特殊情形 1

如果取 $w(\beta) = z^{|\beta|}$, 我们还有

$$\sum_{\phi \in (\mathcal{B}^M/G)} z^{|\phi|} = Z(G; B(z), \dots, B(z^m))$$

Pólya 定理 · 特殊情形 1

如果取 $w(\beta) = z^{|\beta|}$ ，我们还有

$$\sum_{\phi \in (\mathcal{B}^M/G)} z^{|\phi|} = Z(G; B(z), \dots, B(z^m))$$

很自然，带进去就完事了。

Pólya 定理 · 特殊情形 2

如果取 G 为任意置换的集合 R ，我们还有

$$\sum_{i=0}^{\infty} \sum_{\phi \in (B^{[1..i]}/R_i)} w(\phi) = \exp \left(\sum_{i=1}^{\infty} \frac{\sum_{\beta \in B} w^i(\beta)}{i} \right)$$

Pólya 定理 · 特殊情形 2

如果取 G 为任意置换的集合 R ，我们还有

$$\sum_{i=0}^{\infty} \sum_{\phi \in (B^{[1..i]}/R_i)} w(\phi) = \exp \left(\sum_{i=1}^{\infty} \frac{\sum_{\beta \in B} w^i(\beta)}{i} \right)$$

证明可参考之前 MSET 的部分，容易扩展到此处。

Pólya 定理 · 特殊情形 2

如果取 G 为任意置换的集合 R ，我们还有

$$\sum_{i=0}^{\infty} \sum_{\phi \in (B^{[1..i]}/R_i)} w(\phi) = \exp \left(\sum_{i=1}^{\infty} \frac{\sum_{\beta \in B} w^i(\beta)}{i} \right)$$

证明可参考之前 MSET 的部分，容易扩展到此处。

顺带一提 MSET 构造同时是上面的两个特殊情形（

例题和习题

好了，你已经掌握数数的基本方法了，让我们来做一些题目练习一下吧！

hidden

hidden

结语

其实就是从头给大家介绍了生成函数。不过介绍的角度可能大家不太熟悉。

结语

其实就是从头给大家介绍了生成函数。不过介绍的角度可能大家不太熟悉。

虽然例题看起来很鬼畜，但在这个新的符号化体系的视角下它们的确是比较基本的。

El: 我的题提交数怎么翻了一倍

结语

其实就是从头给大家介绍了生成函数。不过介绍的角度可能大家不太熟悉。

虽然例题看起来很鬼畜，但在这个新的符号化体系的视角下它们的确是比较基本的。

El: ~~我的题提交数怎么翻了一倍~~

El: ~~这个 IP 段怎么这么多恶意提交啊，叫 LOJ 管理封了吧~~

结语

其实就是从头给大家介绍了生成函数。不过介绍的角度可能大家不太熟悉。

虽然例题看起来很鬼畜，但在这个新的符号化体系的视角下它们的确是比较基本的。

~~EI: 我的题提交数怎么翻了一倍~~

~~EI: 这个 IP 段怎么这么多恶意提交啊，叫 LOJ 管理封了吧~~

理论上任何生成函数题都可以拿符号化方法分析，除非你搞出一个比它还 nb 的方法……

结语

其实就是从头给大家介绍了生成函数。不过介绍的角度可能大家不太熟悉。

虽然例题看起来很鬼畜，但在这个新的符号化体系的视角下它们的确是比较基本的。

~~El: 我的题提交数怎么翻了一倍~~

~~El: 这个 IP 段怎么这么多恶意提交啊，叫 LOJ 管理封了吧~~

理论上任何生成函数题都可以拿符号化方法分析，除非你搞出一个比它还 nb 的方法……

如果还希望更多资料的话可以看看[这本书](#)。（[另一个下载链接](#)）

感谢

很惭愧，只做了一点微小的贡献

感谢

很惭愧，只做了一点微小的贡献

感谢学军中学提供的交流机会和平台

感谢

很惭愧，只做了一点微小的贡献

感谢学军中学提供的交流机会和平台

感谢 zb 和 wjh 的宽容审核

感谢

很惭愧，只做了一点微小的贡献

感谢学军中学提供的交流机会和平台

感谢 zb 和 wjh 的宽容审核

希望大家看了这个之后可以爱上数数，吊打 \times 义 \times !