

组合对象符号化

× 义 ×

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
习题集 0

题单

习题集 1

定义
大纲
题单

习题集 2

定义
大纲
题单

omega. 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，高维多项式乘法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，高维多项式乘法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。你可以认为这个课件是类似指南针的东西，当你被那些花里胡哨而且不精确的口语描述搞懵的时候可以靠它重回正道。

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，高维多项式乘法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。你可以认为这个课件是类似指南针的东西，当你被那些花里胡哨而且不精确的口语描述搞懵的时候可以靠它重回正道。不得不说不说最经典的实例就是 `luoguP4500` 的题解区

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

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

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。

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

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
- 不必考虑敛散性

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
- 不必考虑敛散性

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**的集合 $\{(\beta, \gamma) | \beta \in \mathcal{B}, \gamma \in \mathcal{C}\}$, 大小函数定义为 $|\alpha|_{\mathcal{A}} = |\beta|_{\mathcal{B}} + |\gamma|_{\mathcal{C}}$ 。

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，把两个序列拼起来也就当然是一个长度为两个序列之和的序列。

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，把两个序列拼起来也就当然是一个长度为两个序列之和的序列。

此时，

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

也就是普通卷积，从而有

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

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

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

注意这种从“有序对”“有序列表”“无序列表”考虑组合对象的视角非常重要。

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

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

注意这种从“有序对”“有序列表”“无序列表”考虑组合对象的视角非常重要。

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：

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

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。可以看作贴了两种不同的标签。

我们有

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

构造组合类·hint

我们指出，如果你想编一个自己的生成函数，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。

其实线性和积性在数数的其他地方也经常见到。比如 DFT 和 FWT。

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**, 当且仅当: $|\beta_1| = |\beta_2|$, 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**, 当且仅当: $|\beta_1| = |\beta_2|$, 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; 且如果 $\alpha_1 \neq \alpha_2$, 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**, 当且仅当: $|\beta_1| = |\beta_2|$, 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; 且如果 $\alpha_1 \neq \alpha_2$, 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)。

$|\alpha|_{\mathcal{A}/G}$ 定义为任意 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 $1..i$ 的排列对应的置换构成的群。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

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

其中 \mathbf{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

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

其中 \mathbf{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

$CYC(\{\circ, \bullet\})$ 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

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

其中 \mathbf{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

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

其中 \mathbf{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：只需要枚举每个元素用了几次。

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：只需要枚举每个元素用了几次。

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 $\mathcal{M} = \{1, 2, \dots, m\}$ 。

再取一个组合类 \mathcal{B} ，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

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 $\mathcal{M} = \{1, 2, \dots, m\}$ 。

再取一个组合类 \mathcal{B} ，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也可以看作染色。

Pólya 定理

两个**染色** $\phi_1, \phi_2 \in \mathcal{B}^M$ 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 \mathcal{B}^M/G 。

Pólya 定理

两个**染色** $\phi_1, \phi_2 \in \mathcal{B}^{\mathcal{M}}$ 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 $\mathcal{B}^{\mathcal{M}}/G$ 。

\mathcal{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 $w(\phi) := \prod_{k \in \mathcal{M}} w(\phi(k))$, 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** $\phi_1, \phi_2 \in \mathcal{B}^{\mathcal{M}}$ 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 $\mathcal{B}^{\mathcal{M}}/G$ 。

\mathcal{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 $w(\phi) := \prod_{k \in \mathcal{M}} w(\phi(k))$, 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 \mathbf{R} , 我们还有

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

Pólya 定理·特殊情形 2

如果取 G 为任意置换的集合 \mathbf{R} , 我们还有

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

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！

ZJOI2018 树

心肺停止

简要题意：rand 出 k 棵有标号有根，大小为 n 的树，问 rand 出来的有根树全部同构的概率。（具体生成方法是对每个节点选一个标号在它之前的节点作为父亲）

也就是对于每个无标号有根树，我们可以在它的节点上连续 k 次分配标号，求所有无标号有根树的分配标号方案数之和。

ZJOI2018 树·解

设对树 t 的方案数为 $w(t)$ 。首先考虑两个等价类的笛卡尔积。

ZJOI2018 树·解

设对树 t 的方案数为 $w(t)$ 。首先考虑两个等价类的笛卡尔积。我们有

$$|(t_1, t_2)| = |t_1| + |t_2|$$

$$w((t_1, t_2)) = w(t_1)w(t_2) \binom{|t_1| + |t_2|}{|t_1|}^k$$

ZJOI2018 树·解

设对树 t 的方案数为 $w(t)$ 。首先考虑两个等价类的笛卡尔积。我们有

$$|(t_1, t_2)| = |t_1| + |t_2|$$

$$w((t_1, t_2)) = w(t_1)w(t_2) \binom{|t_1| + |t_2|}{|t_1|}^k$$

这提示我们搞出一个这样一个生成函数：

记 $T(\mathcal{A}; z) = \sum_{a \in \mathcal{A}} \frac{w(a)z^{|a|}}{|a|!^k}$ 。不难验证它有线性和积性。

ZJOI2018 树·解

那么我们分析一棵树的构造。注意到去掉根的操作比较特殊，它并不是 $\times 2$ ，因为根的标号必须是 1。其实应该是直接把计数序列往下移一位。不妨记为

$$\mathcal{T}^\square = \text{MSET}(\mathcal{T})$$

ZJOI2018 树·解

那么我们分析一棵树的构造。注意到去掉根的操作比较特殊，它并不是 $\times 2$ ，因为根的标号必须是 1。其实应该是直接把计数序列往下移一位。不妨记为

$$\mathcal{T}^\square = \text{MSET}(\mathcal{T})$$

注意我们分析的是 \mathcal{T} 的结构，这和我们使用什么生成函数**无关**，我们只是知道某些结构和某些生成函数有直接对应罢了。**不要把构造和生成函数混为一谈。**

ZJOI2018 树·解

那么我们分析一棵树的构造。注意到去掉根的操作比较特殊，它并不是 $\times z$ ，因为根的标号必须是 1。其实应该是直接把计数序列往下移一位。不妨记为

$$\mathcal{T}^\square = \text{MSET}(\mathcal{T})$$

注意我们分析的是 \mathcal{T} 的结构，这和我们使用什么生成函数**无关**，我们只是知道某些结构和某些生成函数有直接对应罢了。**不要把构造和生成函数混为一谈。**

那么问题就来了，“MSET” 现在应该对应怎样的生成函数运算？

ZJOI2018 树·解

正常的 MSET 是:

$$\prod_{a \in \mathcal{A}} \frac{1}{1 - z^{|a|}}$$

ZJOI2018 树·解

正常的 MSET 是:

$$\prod_{a \in \mathcal{A}} \frac{1}{1 - z^{|a|}}$$

仔细考虑一下当我们选了 ℓ 个同一对象 a 会发生什么。如果我们把它们看作不同并分配标号 (即 $w^\ell(a)$), 任何一种分配方式一定会被恰好重复数 $(\ell!)^k$ 次 (这些相同对象可以任意换, 故每一次分配标号都是 $\ell!$)。

于是此处的 MSET 应改为

$$\prod_{a \in \mathcal{A}} \sum_{i=0}^{\infty} \frac{w^i(a) z^{i|a|}}{(|a|!)^i (i!)^k}$$

下面记后面的那个 Σ 为 $\text{xexp} \left(\frac{w(a) z^{|a|}}{|a|!} \right)$ 。

ZJOI2018 树·解

那么接下来的任务就变成求解 $\prod_{a \in \mathcal{A}} \text{xexp } T(\{a\})$ 。具体来说要尽力把它变成只与 $T(\mathcal{A})$ 有关的形式，这样就得到一个只含有 $T(\mathcal{A})$ 的方程。

ZJOI2018 树·解

那么接下来的任务就变成求解 $\prod_{a \in \mathcal{A}} \text{xexp } T(\{a\})$ 。具体来说要尽力把它变成只与 $T(\mathcal{A})$ 有关的形式，这样就得到一个只含有 $T(\mathcal{A})$ 的方程。

一个自然的想法是，乘法不好处理，那么就取 \ln 再加起来：

$$\exp \sum_{a \in \mathcal{A}} \ln \circ \text{xexp } T(\{a\})$$

那么任务就变为求解 $\ln \circ \text{xexp}$ 。

ZJOI2018 树·解

$$\ln \circ \text{xexp } T(\{a\}) = \ln \sum_{j=0}^{\infty} \frac{w^{jk}(a) x^{|a|j}}{(|a|!)^{jk} (j!)^k}$$

ZJOI2018 树·解

$$\ln \circ \text{xexp } T(\{a\}) = \ln \sum_{j=0}^{\infty} \frac{w^{jk}(a)x^{|a|j}}{(|a|!)^{jk}(j!)^k}$$

事实上可以预处理。下面不妨记

$$\ln \sum_{j=0}^{\infty} \frac{x^j}{(j!)^k} = \sum_{j=0}^{\infty} f_j x^j$$

那么之前的式子就是将 $x = w^k x^{|a|}$ 代入其中。

ZJOI2018 树·解

下面为了简便，把 $\frac{w(a)}{(|a|!)^k}$ 记为 w 。

$$\sum_{a \in \mathcal{A}} \ln \circ \text{xexp } T(\{a\}) = \sum_{a \in \mathcal{A}} \sum_{j=0}^{\infty} f_j (w^k x^{|a|})^j = \sum_{j=0}^{\infty} f_j \sum_{a \in \mathcal{A}} (w^k x^{|a|})^j$$

ZJOI2018 树·解

下面为了简便，把 $\frac{w(a)}{(|a|!)^k}$ 记为 w 。

$$\sum_{a \in \mathcal{A}} \ln \circ \text{xexp } T(\{a\}) = \sum_{a \in \mathcal{A}} \sum_{j=0}^{\infty} f_j (w^k x^{|a|})^j = \sum_{j=0}^{\infty} f_j \sum_{a \in \mathcal{A}} (w^k x^{|a|})^j$$

你会发现这个东西怎么也没法和 $T(\mathcal{A})$ 联系在一起。我们失败了。

ZJOI2018 树·解

下面为了简便，把 $\frac{w(a)}{(|a|!)^k}$ 记为 w 。

$$\sum_{a \in \mathcal{A}} \ln \circ \text{xexp } T(\{a\}) = \sum_{a \in \mathcal{A}} \sum_{j=0}^{\infty} f_j(w^k x^{|a|})^j = \sum_{j=0}^{\infty} f_j \sum_{a \in \mathcal{A}} (w^k x^{|a|})^j$$

你会发现这个东西怎么也没法和 $T(\mathcal{A})$ 联系在一起。我们失败了。不过，虽然这东西暂时和 $T(\mathcal{A})$ 无关，不过我们可以试试强行让它和 $T(\mathcal{A})$ 有关。

ZJOI2018 树·解

下面这一步是最为关键的一步，我们把 $T(\mathcal{A})$ 改成二元生成函数

$$\begin{aligned} T(\mathcal{A}; x, y) &= \sum_{a \in \mathcal{A}} x^{|a|} \sum_{i=0}^{\infty} \frac{w^i(a) y^i}{(|a|!)^i} \\ &= \sum_{a \in \mathcal{A}} \frac{x^{|a|}}{1 - w(a)y/(|a|!)} \end{aligned}$$

说人话就是保存 k 为任意值时的答案。 y 的次数就是 k 的值。其中要注意 $w^i(a)$ 其实指的是 $k = i$ 时的分配标号方案数，不过它也的确显然就是 $k = 1$ 时的 i 次方，所以我们就这么记算了。

可见笛卡尔积现在对应 “ x 上的卷积， y 上的点积”。继续考虑 $x \exp$ 。

ZJOI2018 树·解

不能死板地使用 xexp : k 都不一样了, 自然也不能用之前的定义。易知应重新定义为

$$\text{xexp} : T(\{a\}; x, y) = \frac{x^{|a|}}{1 - w(a)y/(|a|!)} \mapsto \sum_{i=0}^{\infty} y^i \sum_{j=0}^{\infty} \frac{w^{ij}(a)x^{|a|j}}{(|a|!)^{ij}(j!)^i}$$

注意整个 xexp 完全是我们按想要的组合意义编出来的。没有像 $1 + \frac{x}{(1!)^k} + \frac{x^2}{(2!)^k} + \dots$ 那样的幂级数表达式也很正常。最后回到 $\ln \circ \text{xexp}$ 。

ZJOI2018 树·解

前面的推导都相同，直到我们刚失败时的那一步：

$$[y^k] \sum_{a \in \mathcal{A}} \ln \circ \text{xexp } T(\{a\}) = \sum_{j=0}^{\infty} f_j \sum_{a \in \mathcal{A}} (w^k x^{|a|})^j$$

注意到 $[y^k]T(\mathcal{A}) = \sum_{a \in \mathcal{A}} w^k x^{|a|}$ ，故
 $\sum_{a \in \mathcal{A}} w^{jk} x^{j|a|} = [y^{jk}]T(\mathcal{A}; x^j, y)$ 。

ZJOI2018 树·解

前面的推导都相同，直到我们刚失败时的那一步：

$$[y^k] \sum_{a \in \mathcal{A}} \ln \circ \text{xexp } T(\{a\}) = \sum_{j=0}^{\infty} f_j \sum_{a \in \mathcal{A}} (w^k x^{|a|})^j$$

注意到 $[y^k]T(\mathcal{A}) = \sum_{a \in \mathcal{A}} w^k x^{|a|}$ ，故
 $\sum_{a \in \mathcal{A}} w^{jk} x^{j|a|} = [y^{jk}]T(\mathcal{A}; x^j, y)$ 。

则有

$$[y^k] \sum_{a \in \mathcal{A}} \ln \circ \text{xexp } T(\{a\}) = \sum_{j=0}^{\infty} f_j [y^{jk}]T(\mathcal{A}; x^j, y)$$

ZJOI2018 树·结语

这个东西就很好处理了，因为和今天主题关系不大我们直接 skip。

ZJOI2018 树·结语

这个东西就很好处理了，因为和今天主题关系不大我们直接 skip。

可以发现刚刚的过程大部分都只是无脑化式子，如果你熟练的话可以直接秒掉，可见这个做法的优越性。

ZJOI2018 树·结语

这个东西就很好处理了，因为和今天主题关系不大我们直接 skip。

可以发现刚刚的过程大部分都只是无脑化式子，如果你熟练的话可以直接秒掉，可见这个做法的优越性。

有兴趣的同学可以去看看其他题解有多阴间。

无标号荒漠计数

心肺停止

简要题意：求 n 个节点的无标号荒漠的数量。无标号荒漠是无标号无根**边**仙人掌（所有边至多在一个环上）构成的森林。

荒漠的问题直接 MSET 一下即可。也存在一个方法实现从无根到有根的转化但是在仙人掌上可能极为毒瘤。我们直接考虑无标号有根仙人掌就好了。

无标号荒漠计数·解

分类讨论根的儿子和根的关系：以一条边相连，或者同时在一个环上。在同一个环上时，由于根是固定的，所以置换只有两种：恒等和翻转。应用 Pólya 定理

$$\begin{aligned}\mathcal{C} &= \mathcal{Z} \times \text{MSET} \left(\mathcal{C} + \frac{1}{2} \sum_{i=2}^{\infty} \mathcal{C}^i + \frac{1}{2} \sum_{i=1}^{\infty} (\Delta \mathcal{C}^2)^i \times (1 + \mathcal{C}) \right) \\ &= \mathcal{Z} \times \text{MSET} \left(\frac{\mathcal{C}}{2(1-\mathcal{C})} + \frac{\mathcal{C} + \Delta \mathcal{C}^2}{2(1-\Delta \mathcal{C}^2)} \right)\end{aligned}$$

其中 $\Delta \mathcal{C}^2 = \sum_{t \in \mathcal{C}} t^2$ 。

习题

均不要求通过，会口胡就行。因为一些处理生成函数的操作不是今天讲课的重点。

习题集 0·题单

无标号无根树计数

小 Q 的序列

hint: 上面这题用生成函数硬上可以避免猜阴间组合意义。

其他题好像都和今天的内容没什么关系，就不放进例题了。

习题集 1: 有标号计数·定义

一个大小为 n 的对象被称作**标号的**，如果它的每一个节点（此处一般认为一个对象是一个图，但其实是任意对象也可以，只不过图的例子是最直观的）都附带一个互不相同的正整数标号，恰好是 $[1..n]$ 的一个排列。

习题集 1: 有标号计数·定义

一个大小为 n 的对象被称作**标号的**，如果它的每一个节点（此处一般认为一个对象是一个图，但其实是任意对象也可以，只不过图的例子是最直观的）都附带一个互不相同的正整数标号，恰好是 $[1..n]$ 的一个排列。

一个对象被称为**弱标号的**，如果它几乎是标号的，但是其标号不必恰好是 $[1..n]$ 的一个排列。

习题集 1: 有标号计数·定义

一个大小为 n 的对象被称作**标号的**，如果它的每一个节点（此处一般认为一个对象是一个图，但其实是任意对象也可以，只不过图的例子是最直观的）都附带一个互不相同的正整数标号，恰好是 $[1..n]$ 的一个排列。

一个对象被称为**弱标号的**，如果它几乎是标号的，但是其标号不必恰好是 $[1..n]$ 的一个排列。

一个组合类被称作**标号的**，如果它的元素都是标号的对象。

习题集 1: 有标号计数·定义

一个大小为 n 的对象被称作**标号的**，如果它的每一个节点（此处一般认为一个对象是一个图，但其实是任意对象也可以，只不过图的例子是最直观的）都附带一个互不相同的正整数标号，恰好是 $[1..n]$ 的一个排列。

一个对象被称为**弱标号的**，如果它几乎是标号的，但是其标号不必恰好是 $[1..n]$ 的一个排列。

一个组合类被称作**标号的**，如果它的元素都是标号的对象。

我们可以把一个弱标号的对象排成强标号。比如有一个 α ，它的标号是 $\langle 233, 998244353, 1 \rangle$ ，那么 α 的标号就会被重标号成 $\langle 2, 3, 1 \rangle$ 。这个被重标号的 α 记作 $\rho(\alpha)$ 。其实就是离散化。

习题集 1: 有标号计数·定义

一个计数序列 A_n 的 **EGF** (Exponential GF) 是形式幂级数

$$A(z) = \sum_{i \geq 0} A_n \frac{z^n}{n!}$$

因而对于组合类 \mathcal{A} , 它的 EGF 也是

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

(很熟悉吧?) 我们记

$$A_n = n! [z^n] A(z)$$

\mathcal{E}, \mathcal{Z} 的定义如之前一样。接下来的 GF 默认为 EGF。

习题集 1: 有标号计数·大纲

定义两个组合类的**标号积**为（其实就是所谓“分配标号”的严格说法）

$$\beta \star \gamma := \{(\beta', \gamma') | (\beta', \gamma') \text{ 是强标号的}, \rho(\beta') = \beta, \rho(\gamma') = \gamma\}$$

其大小定义为 $|\beta| + |\gamma|$ 。

习题集 1: 有标号计数·大纲

定义两个组合类的**标号积** (其实就是所谓 “分配标号” 的严格说法)

$$\beta \star \gamma := \{(\beta', \gamma') | (\beta', \gamma') \text{ 是强标号的}, \rho(\beta') = \beta, \rho(\gamma') = \gamma\}$$

其大小定义为 $|\beta| + |\gamma|$ 。

两个组合类的**标号积**定义为

$$\mathcal{B} \star \mathcal{C} := \bigcup_{\beta \in \mathcal{B}, \gamma \in \mathcal{C}} \beta \star \gamma$$

显然此处 $\beta \star \gamma$ 两两不交。

习题集 1: 有标号计数·大纲

- 1. 证明: EGF 是 (在标号积下) 积性的。
- 2. 定义有标号 Sequence 构造, 它应该和 $\frac{1}{1-B(z)}$ 对应。指出它的组合意义。
- 3. 定义 Set 构造, 它应该和 \exp 对应。指出它的组合意义。
- 4. 定义 Pointing 构造, 它应该和求导再乘以 z 对应。指出它的组合意义。
- 5. 定义 Substitution 构造, 它应该和复合对应。指出它的组合意义。指出为什么 $B \circ C$ 中的 B 也是 EGF。

习题集 1: 有标号计数·题单

传 世 经 典

小朋友和二叉树

大朋友和多叉树

有标号荒漠计数

有标号边双连通图计数

有标号点双连通图计数

习题集 1: 有标号计数·题单

传 世 经 典

小朋友和二叉树

大朋友和多叉树

有标号荒漠计数

有标号边双连通图计数

有标号点双连通图计数

口胡一下就好了，后三题我也没写过（

习题集 2: 集合幂级数·定义

我们定义一种新的组合类, 其元素的大小函数是 $[1..n]$ 的一个子集, 当然也可看成一个 $[0, 2^n)$ 的一个二进制数。

习题集 2: 集合幂级数·定义

我们定义一种新的组合类，其元素的大小函数是 $[1..n]$ 的一个子集，当然也可看成一个 $[0, 2^n)$ 的一个二进制数。

考虑子集卷积的普通做法，我们定义一个组合类的**集合占位幂级数**为

$$\sum_{\alpha \in \mathcal{A}} \left(z^{|\alpha|} x^{||\alpha||} + \sum_{i=||\alpha||+1}^{\infty} f_i x^i \right)$$

其中 $||\alpha||$ 表示集合 $|\alpha|$ 的大小。 f 是无意义的占位信息。

习题集 2: 集合幂级数·定义

我们定义一种新的组合类，其元素的大小函数是 $[1..n]$ 的一个子集，当然也可看成一个 $[0, 2^n)$ 的一个二进制数。

考虑子集卷积的普通做法，我们定义一个组合类的**集合占位幂级数**为

$$\sum_{\alpha \in \mathcal{A}} \left(z^{|\alpha|} x^{||\alpha||} + \sum_{i=||\alpha||+1}^{\infty} f_i x^i \right)$$

其中 $||\alpha||$ 表示集合 $|\alpha|$ 的大小。 f 是无意义的占位信息。容易验证集合占位幂级数的或卷积

$$[z^s](\sigma \times \tau) = \sum_{p \cup q = s} [z^p]\sigma \times [z^q]\tau$$

是**子集卷积**。或卷积就比较好考虑了，FWT 即可。

习题集 2: 集合幂级数·大纲

- 1. 定义两个组合类的“笛卡尔积”，它应该和其集合占位幂级数的或卷积对应。

- 2. 定义 composition 构造，它应该和其集合占位幂级数的 exp 对应。exp 中出现的乘法均定义为或卷积。指出 composition 构造的组合意义。

hint: 实现的时候，由于 FWT 的优秀性质，我们可以把原幂级数 FWT 然后对每一位（是个形式幂级数）exp 再 FWT 回去。由于 exp 不是瓶颈所以建议暴力实现。

- 3. 定义 decomposition 构造，它应该和其集合占位幂级数的 ln 对应。指出 decomposition 构造的组合意义。

- 4. 定义 modified box 构造

$\square_i \mathcal{A} := \{\alpha \mid \alpha \in \mathcal{A}, i \in |\alpha|\}, |\alpha|_{\square_i \mathcal{A}} := |\alpha|_{\mathcal{A}} / \{i\}$ 。指出它的组合意义。

习题集 2: 集合幂级数·题单

欧拉生成子图

Tutte 多项式

生成仙人掌

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。这就是符号化的意义所在啊！

EI：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。这就是符号化的意义所在啊！

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。这就是符号化的意义所在啊！

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~~

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 x义x!