用生成函数刻画随机过程停时

叶开

杭州学军中学教育集团文渊中学

2024 年 1 月

目录

前言

信息学竞赛中，我们总是会见到许多关于随机过程停时的问题。

前言

信息学竞赛中，我们总是会见到许多关于随机过程停时的问题。我们知道，生成函数是一个很有力的工具，我们能否使用生成函数来刻画随机过程的停时呢？

引入

首先引入一个问题：

Gachapon

现有一随机数生成器，每次生成 $1 \le j \le n$ 的概率为 $A_j / \sum_{i=1}^{n} A_i$，使用其不断独立随机生成。
当对任意 $1 \le j \le n$ 都有 j 被生成了至少 B_j 次时过程停时。
求期望停时。
$A_i, B_i \in \mathbb{N}^+$，$1 \le n, \sum A_i, \sum B_i \le 400$。

引入

这类问题（而不单是这道题目）的常见思路？

引入

这类问题（而不单是这道题目）的常见思路?

- min-max 容斥。

引入

这类问题（而不单是这道题目）的常见思路?

- min-max 容斥。
- 生成函数。

引入

这类问题（而不单是这道题目）的常见思路?

- min-max 容斥。
- 生成函数。
- 势函数与鞅的停时定理。

引入

这类问题（而不单是这道题目）的常见思路？

- min-max 容斥。
- 生成函数。
- 势函数与鞅的停时定理。

接下来着重探讨第二种做法，并给出一类较为通用的方法。

引入

这类问题（而不单是这道题目）的常见思路？

- min-max 容斥。
- 生成函数。
- 势函数与鞅的停时定理。

接下来着重探讨第二种做法，并给出一类较为通用的方法。

由于时间限制，概率论的基本概念不在这里赘述，默认大家都已掌握。

用 PGF 刻画停时

设对于一个总会发生停时的过程，$F(z)=\sum_j P(j\text{ 时刻停时})z^j$，那么总有：

用 PGF 刻画停时

设对于一个总会发生停时的过程，$F(z) = \sum_j P(j\ \text{时刻停时}) z^j$，
那么总有：
结论 1：$F(1) = 1$。

用 PGF 刻画停时

设对于一个总会发生停时的过程，$F(z) = \sum_j P(j\text{ 时刻停时})z^j$，
那么总有：
结论 1：$F(1) = 1$。
结论 2：期望停时为 $F'(1)$。

用 PGF 刻画停时

设对于一个总会发生停时的过程，$F(z) = \sum_j P(j\text{ 时刻停时})z^j$，
那么总有：
结论 1：$F(1) = 1$。
结论 2：期望停时为 $F'(1)$。
理由显然。

用 PGF 刻画停时

设对于一个总会发生停时的过程，$F(z) = \sum_j P(j\text{ 时刻停时})z^j$，
那么总有：
结论 1：$F(1) = 1$。
结论 2：期望停时为 $F'(1)$。
理由显然。
因此问题往往就被转化为刻画出 $F(z)$ 的信息。

用 EGF 刻画时间轴

由于 EGF 乘法能够分配标号，使用 EGF 刻画时间轴往往会比较容易。

用 EGF 刻画时间轴

由于 EGF 乘法能够分配标号，使用 EGF 刻画时间轴往往会比较容易。
但是问题来了：使用 PGF 刻画停时信息时往往需要 OGF 形式，怎么办呢？

Laplace 变换

我们规定 $\hat{F}(z) = \sum_j A_j z^j / j!$ 的 Laplace 变换结果为

$$F(z) = \mathcal{L}\hat{F}(z) = \sum_j A_j z^j$$

其中 $\mathcal{L}$ 称为 Laplace 变换算子。

Laplace 变换

我们规定 $\hat{F}(z) = \sum_j A_j z^j / j!$ 的 Laplace 变换结果为

$$F(z) = \mathcal{L}\hat{F}(z) = \sum_j A_j z^j$$

其中 $\mathcal{L}$ 称为 Laplace 变换算子。
容易发现这实现了从 EGF 到 OGF 的变换。

Laplace 变换

结论 1：Laplace 变换算子是线性算子。

Laplace 变换

结论 1：Laplace 变换算子是线性算子。
结论 2：$\mathcal{L}z^a e^{bz} = a!z^a/(1-bz)^{a+1}$。

Laplace 变换

结论 1：Laplace 变换算子是线性算子。
结论 2：$\mathcal{L}z^a e^{bz} = a!z^a/(1-bz)^{a+1}$。
因此对于形如 $z^a e^{bz}$ 的线性组合的 EGF，我们总可以将其改写成 OGF 形式。

例题

以上内容已经可以解决一大批题目了。
我们先看一下引言中的题目。

Gachapon

现有一随机数生成器，每次生成 $1 \leq j \leq n$ 的概率为 $A_j / \sum_{i=1}^{n} A_i$，使用其不断独立随机生成。
当对任意 $1 \leq j \leq n$ 都有 j 被生成了至少 B_j 次时过程停时。
求期望停时。
$A_i, B_i \in \mathbb{N}^+$，$1 \leq n, \sum A_i, \sum B_i \leq 400$。

Gachapon

直接做不好做，不妨枚举最后一步生成的数，最后把 1 的贡献加回去即可。

Gachapon

直接做不好做，不妨枚举最后一步生成的数，最后把 1 的贡献加回去即可。
设 $S = \sum_i A_i$，那么立刻就有答案为

$$1 + (\mathcal{L}(\sum_{i=1}^{n} \frac{(A_i z/S)^{B_i - 1}}{(B_i - 1)!} \prod_{j \neq i} (\exp(A_j z/S) - \sum_{k < B_j} \frac{(A_j z/S)^k}{k!})))' \circ 1$$

Gachapon

直接做不好做，不妨枚举最后一步生成的数，最后把 1 的贡献加回去即可。

设 $S = \sum_i A_i$，那么立刻就有答案为

$$1 + (\mathcal{L}(\sum_{i=1}^{n} \frac{(A_i z/S)^{B_i - 1}}{(B_i - 1)!} \prod_{j \neq i} (\exp(A_j z/S) - \sum_{k<B_j} \frac{(A_j z/S)^k}{k!})))' \circ 1$$

设 $u = \exp(z/S)$，容易把内部的式子经过暴力乘法表示成关于 u, z 的多项式，复杂度不会超过 $O(v^3)$（其中 $v = \max\{S, \sum B\}$）。

Gachapon

直接做不好做，不妨枚举最后一步生成的数，最后把 1 的贡献加回去即可。

设 $S = \sum_i A_i$，那么立刻就有答案为

$$1 + (\mathcal{L}(\sum_{i=1}^{n} \frac{(A_i z/S)^{B_i - 1}}{(B_i - 1)!} \prod_{j \neq i} (\exp(A_j z/S) - \sum_{k<B_j} \frac{(A_j z/S)^k}{k!})))' \circ 1$$

设 $u = \exp(z/S)$，容易把内部的式子经过暴力乘法表示成关于 u, z 的多项式，复杂度不会超过 $O(v^3)$（其中 $v = \max\{S, \sum B\}$）。

最后直接把每个 $z^a u^b$ 对答案的贡献单独计算一遍即可。

例题

事实上可以发现，这样不仅能求出停时，还能求出以其中某些事件发生停时的概率。
求出对应的 PGF 后直接代入 1 即为答案。

猎人杀

现在有 n 个人，每人有一个权值 w_i。
每次会在剩下的人集合中，以 w_i 比例的概率独立随机选择第 i 个人让其离开，重复直到所有人都离开。
求最后离开的人是第一个人的概率，对 998244353 取模。
$w_i \in \mathbb{N}^+$，$1 \le \sum_{i=1}^{n} w_i \le 10^5$。

猎人杀

由于概率会变化，看上去这并不好直接做。

猎人杀

由于概率会变化，看上去这并不好直接做。
但是注意到，如果允许选择一个已经离开的人再次离开，那么这个就变成了和上一题类似的问题，只是 $B_i = 1$。

猎人杀

由于概率会变化，看上去这并不好直接做。
但是注意到，如果允许选择一个已经离开的人再次离开，那么这个就变成了和上一题类似的问题，只是 $B_i = 1$。
这样，令 $S = \sum w_i$，就有答案为

$$\frac{w_1}{S}(\mathcal{L}(\prod_{j=2}^{n}(\exp(w_j z/S) - 1))) \circ 1$$

猎人杀

由于概率会变化，看上去这并不好直接做。
但是注意到，如果允许选择一个已经离开的人再次离开，那么这个就变成了和上一题类似的问题，只是 $B_i = 1$。
这样，令 $S = \sum w_i$，就有答案为

$$\frac{w_1}{S}(\mathcal{L}(\prod_{j=2}^{n}(\exp(w_j z/S) - 1))) \circ 1$$

设 $u = \exp(z/S)$，使用分治卷积算法在 $O(S \log^2 S)$ 时间内计算出关于 u 的多项式后即可对每个 u^a 算出对答案的贡献，进而求出答案。

猎人杀

由于概率会变化，看上去这并不好直接做。
但是注意到，如果允许选择一个已经离开的人再次离开，那么这个就变成了和上一题类似的问题，只是 $B_i = 1$。
这样，令 $S = \sum w_i$，就有答案为

$$\frac{w_1}{S}(\mathcal{L}(\prod_{j=2}^{n}(\exp(w_j z/S) - 1))) \circ 1$$

设 $u = \exp(z/S)$，使用分治卷积算法在 $O(S \log^2 S)$ 时间内计算出关于 u 的多项式后即可对每个 u^a 算出对答案的贡献，进而求出答案。
使用 $f = \exp \ln f$（其中 $f = \prod_{j \neq 1}(1 - u^{w_j})$）的技巧可以进一步做到 $O(S \log S)$，此处略去。

通用测评号

现在有 n 个盒子，每个盒子最多放 a 个球，初始均为空。
不断独立均匀随机选取一个不满的盒子放入一个球，直到所有盒子内的球数均 $\geq b$ 发生停时，问最后期望有多少盒子被填满，对 998244353 取模。
$1 \leq n \leq 250$，$1 \leq b < a \leq 250$。

通用测评号

先特判掉 $n = 1$ 的情况：此时答案显然为 0，无须考虑。

通用测评号

先特判掉 $n = 1$ 的情况：此时答案显然为 0，无须考虑。
注意到答案即为每个盒子被填满的概率和，也即 n 乘上单个盒子被填满的概率，因此只用求出单个盒子被填满的概率。

通用测评号

先特判掉 $n=1$ 的情况：此时答案显然为 0，无须考虑。
注意到答案即为每个盒子被填满的概率和，也即 n 乘上单个盒子被填满的概率，因此只用求出单个盒子被填满的概率。
容易发现假如允许继续往已经被填满的盒子里填，答案是不变的。枚举最后一步选择的是哪个盒子，就得到概率为

$$\frac{n-1}{n}(\mathcal{L}_z((\exp t - \sum_{j=0}^{a-1}\frac{t^j}{j!})(\exp t - \sum_{j=0}^{b-1}\frac{t^j}{j!})^{n-2}\frac{t^{b-1}}{(b-1)!}))_z(1)$$

其中 $t=z/n$。

通用测评号

设 $u = \exp t$，那么发现核心就是要快速求出

$$(u - \sum_{j=0}^{a-1} \frac{t^j}{j!})(u - \sum_{j=0}^{b-1} \frac{t^j}{j!})^{n-2} \frac{t^{b-1}}{(b-1)!}$$

通用测评号

设 $u = \exp t$，那么发现核心就是要快速求出

$$(u - \sum_{j=0}^{a-1} \frac{t^j}{j!})(u - \sum_{j=0}^{b-1} \frac{t^j}{j!})^{n-2} \frac{t^{b-1}}{(b-1)!}$$

容易使用 ODE 在 $O(n^2 a)$ 的复杂度下求出，设为 $\sum_{i,j} F_{i,j} t^i u^j$，那么容易得到概率即为

$$(n-1) \sum_{i,j} \frac{i! F_{i,j}}{(n-j)^{i+1}}$$

更多例题

在解决问题时，有时还要结合一类特殊的符号化方法。这样还可以解决更多题目，譬如下面这题。

Slime and Biscuits

n 个人，第 i 个人有 a_i 个球。
设 $S = \sum a_i$，每次会在 S 个球中独立均匀随机选择一个并传给任意一个不是当前人的人，直到所有球都归一人所有后过程停时。
求期望停时对 998244353 取模后的值。
$a_i \in \mathbb{N}$，$1 \le n \le 10^5$，$1 \le S \le 3 \times 10^5$。

本题使用这类方法，可以通过一些繁复的技巧做到 $O(n+S)$。

更多例题

在解决问题时，有时还要结合一类特殊的符号化方法。这样还可以解决更多题目，譬如下面这题。

Slime and Biscuits

n 个人，第 i 个人有 a_i 个球。
设 $S=\sum a_i$，每次会在 S 个球中独立均匀随机选择一个并传给任意一个不是当前人的人，直到所有球都归一人所有后过程停时。
求期望停时对 998244353 取模后的值。
$a_i \in \mathbb{N}$，$1 \le n \le 10^5$，$1 \le S \le 3 \times 10^5$。

本题使用这类方法，可以通过一些繁复的技巧做到 $O(n+S)$。但由于时间所限，在这里不进行展开了，感兴趣的同学可以阅读我的集训队论文。

结尾

Thank you!