浅谈一类基于概率的约瑟夫问题

杭州第二中学 叶卓睿

摘 要

“一类基于概率的约瑟夫问题”是一类有趣的概率期望问题，在此之前，这类问题只零零散散地出现过两三次。

笔者对这类问题进行了深入研究，最终得到了较为系统的成果，本文将通过三道例题展示这类题目的各种技巧和思路，并对这些处理手段进行对比和分析，希望读者以后遇到这类问题时有迹可循。

另外，本文也提供了一种很好的出题方向：对一个模型进行深入研究，对相关问题进行对比分析，从而获得新的问题和思路。

1 前言

本文介绍了处理“一类基于概率的约瑟夫问题”这类题目的各种技巧和思路，并对这些处理手段进行对比和分析。在此之前，这类问题只零零散散地出现过两三次。笔者对这类问题进行了深入研究，最终得到了较为系统的成果，这些在文中将会以三道例题的形式进行展示，希望读者以后遇到这类问题时有迹可循。

本文第二节将介绍这类问题的定义，第三节将给出这类问题的通用观察和结论，第四、五、六节将通过三个例题对这类问题的解决办法进行较为全面的展示，第七节将对上述三个例题的解法和思路进行比较和分析。

2 定义

给一个 1 到 n 依次连接的环，有个指针从 1 开始移动，每次指针所在位置有 $p(p > 0)$ 的概率消失掉，然后指针向右移动，游戏在所有位置都消失时结束。

在此基础上需要求解一些概率期望问题。

为了方便描述，下文约定 $q = 1 - p$ 。

3 一些观察

引理 1：一个被指针经过 i 次的点，存活概率为 q^i 。

证明. 转化问题，认为一个点消失后不会真的从环上消失，而是打上一个“删除”标记。那么一个点存活当且仅当每次经过都没有打“删除”标记，故存活概率为 q^i 。 □

引理 2：当指针落在位置 c 时，$[1, c-1]$ 中每个点已经消失的概率相同，$[c+1, n]$ 中每个点已经消失的概率也相同。

证明. 注意到此时 $[1,c-1]$ 中每个点经过次数相同，由引理 1，它们的存活概率相同。$[c+1,n]$ 也同理。 □

4 一个简单的问题

例题 1. 迫真大游戏 [1]

给一个 1 到 n 的环，有个指针从 1 开始移动，每次指针所在位置有 p 的概率消失掉，然后指针向右移动。求每个点是最后一个消失的概率。

$n \le 2 \cdot 10^5$

4.1 分析问题

我们关心的位置未必是起点，考虑先让指针转几步使得它落在我们关心的位置上。

令 f_n 为环大小为 n 时，1 号点最后消失的概率。尝试先求出 f_n 。

4.2 F 的求法

枚举第一个分身在第 $i+1$ 轮消失，计算其概率，可以得到

$$
\begin{aligned}
f_n &= \sum_{i=0}^{\infty} pq^i(1-q^i)^{n-1} \\
&= \sum_{i=0}^{\infty} pq^i \sum_{j=0}^{n-1} (-1)^j (1-p)^{ij} \binom{n-1}{j} \\
&= p \sum_{j=0}^{n-1} (-1)^j \binom{n-1}{j} \sum_{i=0}^{\infty} q^{(j+1)i} \\
&= p \sum_{j=0}^{n-1} (-1)^j \binom{n-1}{j} \frac{1}{1-q^{j+1}}
\end{aligned}
$$

[1] CometOJ Contest 4, Problem E

将组合数用阶乘展开后这是个卷积的形式，可以 FFT 解决。

FFT 解决卷积问题的具体细节可以参考毛啸在 2016 年信息学奥林匹克中国国家队候选队论文中所写的《再探快速傅里叶变换》，这里不再赘述。

4.3 求解原问题

通过枚举指针移动过程中消失的位置个数，我们得到

$$ans_k = \sum_{i=0}^{k-1} \binom{k-1}{i} p^i q^{k-1-i} f_{n-i}$$

这也可以化为卷积的形式。综上所述，原问题可以通过 2 次 FFT 解决，总时间复杂度为 $O(n \log n)$ 。

5 对例题一进行改编

例题 2. Game on a Circle [2]

给一个 1 到 n 的环，有个指针从 1 开始移动，每次指针所在位置有 p 的概率消失掉，然后指针向右移动。对于每个 i ，求 c 号点是第 i 个消失的概率。

$n \leq 10^6$

5.1 生成函数方法

我们试试直接用生成函数推式子。

令 a_i 为 c 号点是第 $i+1$ 个消失的概率，我们希望求出 $A(x) = \sum a_i x^i$ 。则有

$$\begin{aligned}
A(x) &= \sum_{t=0}^{\infty} q^t p \left(q^{t+1} + (1-q^{t+1})x\right)^{c-1} (q^t + (1-q^t)x)^{n-c} \\
&= \sum_{t=0}^{\infty} q^t p \left(q^{t+1}(1-x) + x\right)^{c-1} (q^t(1-x) + x)^{n-c} \\
&= p \sum_i \sum_j \binom{c-1}{i} \binom{n-c}{j} \sum_{t=0}^{\infty} q^i (1-x)^{i+j} x^{n-1-i-j} q^{t(1+i+j)} \\
&= p \sum_i \sum_j \binom{c-1}{i} \binom{n-c}{j} q^i (1-x)^{i+j} x^{n-1-i-j} \frac{1}{1-q^{1+i+j}}
\end{aligned}$$

[2] 2020 Multi-University Training Contest 3, Problem K

观察式子，可以记 $f_k = \sum\limits_{i+j=k} \binom{c-1}{i}\binom{n-c}{j} q^i$ ，则有

$$A(x) = p \sum_i \frac{f_i}{1-q^{i+1}} (1-x)^i x^{n-1-i}$$

由于 c 为常数，f_n 是一个卷积的形式，可以 FFT 。现在考虑求解 $A(x)$ ，继续推导可知

$$[x^{n-i+j-1}]A(x) = p \sum_i \sum_j \binom{i}{j} (-1)^j \frac{f_i}{1-q^{i+1}}$$

这也是卷积的形式，问题即可使用 2 次 FFT 得到解决。

5.2 优化

事实上我们可以做得更快。记 $F(x)$ 为 f_i 对应的普通型生成函数，我们发现 $F(x) = (1+qx)^{c-1}(1+x)^{n-c}$ 。

通过对生成函数求导，可以得到

$$F'(x) = (c-1)q\frac{F(x)}{1+qx} + (n-c)\frac{F(x)}{1+x}$$

对比系数，得到 f_i 的递推式：

$$f_{i+1} = \frac{((c-1)q + n - c - (q+1)i)\, f_i + q(n-i) f_{i-1}}{i+1}$$

综上所述，我们只需要 1 次 FFT 就解决了这个问题，时间复杂度 $O(n\log n)$ 。

5.3 其它做法

事实上，使用 4.1 中提到的拆解过程的办法，可以做到 $O(n\log^2 n)$ ，由于复杂度较劣以及篇幅所限，这里不进行展示。

6 更复杂的问题

例题 3. Eat Cards, Have Fun [3]

给一个 1 到 n 的环，第 i 个点上有数字 A_i ，有个指针从 1 开始移动，每次指针所在位置有 p 的概率消失掉然后加入序列 b 的末尾，然后指针向右移动。求排列 b 在所有 $[1,n]$ 的排列中，按字典序排序后的序号期望。

$n \le 300$

[3] 2018 Multi-University Training Contest 4, Problem H

6.1 官方题解做法

6.1.1 初步分析

考虑利用期望的线性性拆贡献。记排列为 P_n ，则这个排列的序号为 $Ans = \sum_i (n-i)! \sum_{j>i} [P_j < P_i] + 1$ 。

贡献还可以通过枚举位置 i ，以及在 i 消失之前 $[1, i-1]$ 共消失了 a 个，$[i+1, n]$ 共消失了 b 个，$[1, i-1]$ 共消失了 c 个小于 P_i 的，$[i+1, n]$ 共消失了 d 个小于 P_i 的来计算。

记 $dp_{i,a,b}$ 为在 i 消失之前 $[1, i-1]$ 共消失了 a 个，$[i+1, n]$ 共消失了 b 个的概率，l_i 为 $[1, i-1]$ 中小于 P_i 的个数，r_i 为 $[i+1, n]$ 中小于 P_i 的个数，则有

$$Ans = \sum_i \sum_a \sum_b \sum_c \sum_d (n-a-b-1)!(A_i - c - d - 1)\binom{l_i}{c}\binom{i-1-l_i}{a-c}\binom{r_i}{d}\binom{n-i-r_i}{b-d} dp_{i,a,b} + 1$$

6.1.2 求解 dp 数组

首先考虑如何计算 $dp_{i,a,b}$ 。

$$\begin{aligned} dp_{i,a,b} &= \sum_{t=0}^{\infty} q^t p (q^{t+1})^{i-1-a} (1-q^{t+1})^a (q^t)^{n-i-b} (1-q^t)^b \\ &= \sum_{j=0}^{a} \sum_{k=0}^{b} \binom{a}{j}\binom{b}{k} (-1)^{j+k} q^{i-1-a+j} \sum_{t=0}^{\infty} q^{t(n+j+k-a-b)} \\ &= \sum_{j=0}^{a} \sum_{k=0}^{b} \binom{a}{j}\binom{b}{k} (-1)^{j+k} \frac{q^{i-1-a+j}}{1-q^{n+j+k-a-b}} \end{aligned}$$

暴力计算时间复杂度为 $O(n^5)$ ，需要优化。

优化计算 定义

$$h_{1,a,b} = \sum_{j=0}^{a} \sum_{k=0}^{b} \binom{a}{j}\binom{b}{k} (-1)^{j+k} \frac{q^{a+j}}{1-q^{n+j+k-a-b}}$$

则有

$$dp_{i,a,b} = p q^{i-1} h_{1,a,b}$$

定义

$$h_{2,a,b} = \sum_k \binom{b}{k} (-1)^k \frac{1}{1-q^{n-a-b+k}}$$

则有

$$h_{1,a,b} = \sum_j \binom{a}{j} (-1)^j q^{j-a} h_{2,a-j,b}$$

那么，我们可以分别计算出 $h_{2,a,b}, h_{1,a,b}, dp_{i,a,b}$ ，这部分时间复杂度优化至了 $O(n^3)$ 。

6.1.3 原问题的求解

回到答案式子

$$Ans=\sum_i\sum_a\sum_b\sum_c\sum_d(n-a-b-1)!(A_i-c-d-1)\binom{l_i}{c}\binom{i-1-l_i}{a-c}\binom{r_i}{d}\binom{n-i-r_i}{b-d}dp_{i,a,b}+1$$

暴力计算仍然是 $O(n^5)$ 的。

优化计算 接下来我们定义一些辅助函数加速计算：

$$f_{i,0,a}=\sum_c\binom{l_i}{c}\binom{i-1-l_i}{a-c}$$

$$f_{i,1,a}=\sum_c\binom{l_i}{c}\binom{i-1-l_i}{a-c}c$$

$$g_{i,0,b}=\sum_d\binom{r_i}{b}\binom{n-i-r_i}{b-d}$$

$$g_{i,1,b}=\sum_d\binom{r_i}{b}\binom{n-i-r_i}{b-d}d$$

然后将 $A_i-c-d-1$ 拆为 $(A_i-1)-c-d$ 三部分分别算贡献，可以得到：

$$Ans=\sum_i\sum_a\sum_b\left((A_i-1)f_{i,0,a}g_{i,0,b}-f_{i,1,a}g_{i,0,b}-f_{i,0,a}g_{i,1,b}\right)(n-a-b-1)!dp_{i,a,b}+1$$

时间复杂度 $O(n^3)$ 。

6.2 新的思考角度

6.2.1 初步思路

根据期望的线性性，贡献可以拆成 $\sum_i\sum_j\sum_t[A_i>A_j]Pr(i\text{先于}j\wedge i\text{在时刻}t\text{消失})(n-t)!$ 。
由引理 2，我们发现固定 i,t 后，贡献只需与 j,i 的相对大小有关。
那么思路就很明显了，分 $i>j$ 和 $i<j$ 两类情况进行讨论。

6.2.2 动态规划算法

接下来我们先讨论 $i>j$ 的情况，可以设计 dp：$f_{n,i}$ 表示长度为 n 的环，i 先于 $i-1$ 消失的所有情况下 $(n-t)!$ 的期望。

转移如下：

$$f_{n,i} = \begin{cases} qf_{n,n} + p(n-1)! & i = 1 \\ qf_{n,i-1} & i = 2 \\ qf_{n,i-1} + pf_{n-1,i-1} & i > 2 \end{cases}$$

按 n 从小到大进行求解，对于每个 n 直接高斯消元可以 $O(n^3)$ ，总复杂度 $O(n^4)$ 。

事实上对于每个 n ，dp 转移是环上高斯消元的形式。而环上高斯消元是可以做到 $O(n)$ 的，具体做法是：设 $x = f_{n,n}$ ，那么对于 $i \in [1, n]$ 可以分别把 $f_{n,i}$ 表示成 $k_i x + b_i$ 的形式，从而得到一个一元一次方程，解出 x 后即可计算出 $f_{n,i}$ 。

另一种情况 $i < j$ 类似，篇幅所限这里不再赘述。

这样时间复杂度为 $O(n^2)$ ，实现难度较低。

6.3 进一步优化

dp 转移式子最麻烦的一点在于有环，我们尝试先求解第一列。

6.3.1 dp 数组第一列的求解

记 $f_n = dp_{n,1}$ 。枚举有多少个是在 1 之后消失的：

$$\begin{aligned} f_n &= \sum_{i=0}^{n-2} \binom{n-2}{i} \sum_{t=0}^{\infty} q^t p q^t (q^t)^i (1-q^t)^{n-2-i} (i+1)! \\ &= \sum_{i=0}^{n-2} \binom{n-2}{i} \sum_{j=0}^{n-2-i} \binom{n-2-i}{j} (-1)^j p(i+1)! \sum_{t=0}^{\infty} q^{t(2+i+j)} \\ &= p \sum_{i=0}^{n-2} \binom{n-2}{i} \sum_{j=0}^{n-2-i} \binom{n-2-i}{j} (-1)^j (i+1)! \frac{1}{1-q^{2+i+j}} \\ &= p \sum_{i=0}^{n-2} \sum_{j=0}^{n-2-i} \binom{n-2}{i+j} \binom{i+j}{i} (-1)^j (i+1)! \frac{1}{1-q^{2+i+j}} \end{aligned}$$

优化计算 为加速计算，我们令

$$\begin{aligned}g_k &= \sum_{i+j=k}\binom{i+j}{i}(i+1)!(-1)^j\\ &= \sum_{i+j=k}\frac{(i+j)!(i+1)(-1)^j}{j!}\end{aligned}$$

这是一个卷积的形式，可以 FFT 解决。

于是有

$$\begin{aligned}f_n &= p\sum_k \frac{1}{1-q^{2+k}}\binom{n-2}{k}g_k\\ &= p\sum_k\left(\frac{1}{1-q^{2+k}}g_k\right)\binom{n-2}{k}\end{aligned}$$

将组合数用阶乘展开后，这也是一个卷积的形式，可以 FFT 解决。

因此，我们可以 $O(n\log n)$ 地求出 $dp_{i,1}$ 。

6.3.2 dp 数组第 n 行的求解

回顾 dp 转移方程

$$f_{n,i} = \begin{cases} qf_{n,n} + p(n-1)! & i = 1\\ qf_{n,i-1} & i = 2\\ qf_{n,i-1} + pf_{n-1,i-1} & i > 2\end{cases}$$

已知 $dp_{i,1}$ ，可以 $O(n)$ 地计算出 $dp_{i,2}$ ，这样做是为了解决 $i = 2$ 时转移比较特殊的问题。

记 $g_i = dp_{i,2}, h_i = dp_{n,i+2}$ 。计算每个 g_j 对 h_i 的贡献，观察 dp 转移式子可知：

$$h_i = \sum_j g_j\binom{i}{n-j}p^{n-j}q^{i+j-n}$$

将组合数用阶乘展开后，这也是一个卷积的形式，可以 FFT 解出 $f_{n,i}$ 。

6.3.3 回到原问题

考虑求解原问题，$i > j$ 的情况贡献为

$$Ans = \sum_{i>j}[A_i > A_j]f_{n,i}$$

，可以使用树状数组统计 $Ans = \sum_{i>j}[A_i > A_j]$ 。

$i < j$ 的部分使用类似的做法即可做到总时间复杂度 $O(n\log n)$ 。

引理 3：长度为 n 的环，1 先于 2 消失的所有情况下 $(n-t)!$ 的期望与 1 先于 n 相同。

证明. 枚举 1 是在第 $i+1$ 次经过时消失的，那么 $[2, n]$ 的每个点经过次数相同，由引理 1 可知它们消失的概率相同，因此在这个问题中是等价的。 □

由引理 3，$i < j$ 的情况下 dp 数组第一列和 $i > j$ 完全相同，由此可以减小一定的代码量和常数。

7 对比与分析

例题一只需要仔细分析题目，发现过程可以拆成独立的两步，逐个解决即可，算是这类问题中较为基础的情形。

在引理 1 的帮助下，只使用数学推导手段就可以在一些问题（如例题二）上取得不错的效果，其优势是思维难度不高，不过缺点是推导、计算较为繁琐，在一些情况下可能无法进行优化。

例题三很好地体现了代数推导的局限性：不仅繁琐而且无法做到很优的复杂度。笔者通过对问题模型进行观察，得到一个很强的性质，从而得到一种 $O(n^2)$ 的简洁的 dp 做法。借鉴例题一的思想，指针初始位于 1 的情况是容易解决的，此后 dp 的转移无环，因此可以使用常见的生成函数技巧进行优化，从而做到 $O(n \log n)$ 的优秀复杂度，这相对于官方题解给出的 $O(n^3)$ 是一个巨大的飞跃。

8 研究思路与成果

笔者最初在做 2018 年杭电多校联合训练第 4 场 H 题时思考出 $O(n^2)$ 的做法，比官方题解的推式子做法更优秀。这启发我对这一问题模型进行研究。之后我发现这一模型存在一些基本结论和观察，而且不同问题在解法上也有一定的联系。例如 CometOJ 第 4 场的 E 题就是先将指针所在位置移动到 1 ，从而拆解问题，事实上笔者也正是受这个思路启发，才思考出了例题三最后一步的优化。在研究过程中，笔者对例题一进行修改并进行了深入的思考，最终命制了例题二，此题虽和例题一题意比较相似，做法却大相径庭，最后几乎只需要一些推导的技巧就可以解决，这说明代数推导在一些情况下也有用武之地，选手在做这类题目时需要灵活选择解法，不能思维定式化。

顺带一提，例题二作为 2020 年杭电多校联合训练第 3 场 K 题使用，场上共有 9 支队伍通过，可见此题难度适中、区分度良好，同时也说明选手对这类模型还不够熟悉。

9 总结

本文通过三个例题展示了解决“一类基于概率的约瑟夫问题”的各种思路和技巧，对这类问题进行了详细的分析研究，最终都得到了 $O(n \log n)$ 的优秀解法，希望读者遇到这类

问题时有迹可循。

另外，对一个模型进行深入研究，对相关问题进行对比分析也提供了一种很好的出题方向，例如本文例题二就是这样命制的。

最后，希望本文起到抛砖引玉的作用，吸引更多读者来研究这一类问题，或是将本文提及的思想应用到更多其它问题上。

致谢

感谢中国计算机学会提供学习和交流的平台。

感谢杭州第二中学李建老师的关心和指导。

感谢国家集训队教练高闻远的指导。

感谢吴越学长、周欣同学对本文的帮助。

感谢父母对我的关心和照顾。

感谢所有帮助过我的人。

参考文献

[1] Ronald L. Graham, Donald E. Knuth, Oren Patashnik. Concrete Mathematics .

[2] 金策，《生成函数的运算与组合计数问题》，2015 年信息学奥林匹克中国国家队候选队员论文

[3] 毛啸，《再探快速傅里叶变换》，2016 年信息学奥林匹克中国国家队候选队员论文