

《中奖率》命题报告

山东省平邑第一中学 唐绍轩

目录

1	题面	2
1.1	题目描述	2
1.2	限制与约定	2
2	题解	3
2.1	算法一	3
2.2	算法二	3
2.3	算法三	4
2.4	算法四	4
2.5	算法五	5
2.6	算法六	6
2.7	算法七	6
3	总结	7
4	参考资料	7

1 题面

1.1 题目描述

小 T 在玩一款游戏，游戏中有彩票。彩票只有中奖与不中奖两种情况。

小 T 玩了一段时间后发现，如果将中奖看作 1，不中奖看作 0 的话，那么一直买彩票后生成的无限长的 01 序列 T 有一个长为 n 的循环节 S 。具体来讲， T 满足 $T_i = S_{((i-1) \bmod n)+1}$ 。

定义 f_i 为只考虑前 i 次买彩票时的中奖率，更具体地，若前 i 个数中有 a_i 个 1，那么 $f_i = \frac{a_i}{i}$ 。小 T 想知道中奖率何时比较高，于是他会有 q 次询问，具体形式如下：

1. 给定整数 k ，设 f_w 为序列 f 中第 k 大的值，求 w 。
2. 给定整数 k ，求出 f_k 的排名，若排名为无穷大，输出 `inf`。

注意：我们称 f_a 比 f_b 大当且仅当 $f_a > f_b$ 或 $f_a = f_b \wedge a < b$ 。可以证明在这样的定义下，序列 f 中第 k 大的值是存在且唯一的。

1.2 限制与约定

对于所有测试数据， $1 \leq n \leq 10^5, 1 \leq k \leq 10^{10000}, 1 \leq q \leq 20$ 。

子任务编号	$n \leq$	$k \leq$	特殊性质	分值
1	10^5	10^{10000}	$S_1 = S_2 = \dots = S_{n-1} = 0, S_n = 1$	1
2	10	1000	-	9
3	10^5	10^9		9
4		10^{10000}	$op = 2$	13
5			$S_1 = 1, S_2 = S_3 = \dots = S_n = 0$	20
6	200		-	18
7	10^5			30

2 题解

设 g_k 表示序列 f 的第 k 大的值。

首先考虑 $i, i+n$ 两个位置，它们之间恰好相差一个循环节，所以有 $f_{i+n} = \frac{a_i+a_n}{i+n}$ ，而 $f_i = \frac{a_i}{i}$ ，可以发现 $f_i > f_{i+n}$ 当且仅当 $f_i > f_n = \frac{a_n}{n}$ 。

将序列 f 按照下标 $\bmod n$ 划分成 n 个序列。若某个序列 $\{f_{i+kn}\}, k \in \mathbb{N}, 1 \leq i \leq n$ 满足 $f_i > f_n$ ，那么这个序列单调递减且满足 $f_{i+kn} > f_n$ 。若 $f_i < f_n$ ，同理推出 $f_{i+kn} < f_n$ 。

所以有以下两种情形：

1. 前 n 个数中没有数大于 f_n ，那么此时的最大值 f_i 满足 $f_i = f_n$ ，而有无限个数 $f_n, f_{2n}, f_{3n}, \dots$ 的值都等于 f_n ，所以对于所有 j ， g_j 都与 f_n 相等。

2. 前 n 个数中存在某个 f_i 满足 $f_i > f_n$ ，那么 $\{f_{i+kn}\}$ 这个序列的所有元素均大于 f_n ，所以对于所有 j ， g_j 大于 f_n ，只需考虑首项大于 f_n 的序列。

情形 1 较为简单，设 $f_{k_1}, f_{k_2}, f_{k_3}, \dots, f_{k_n}, 1 \leq k_1 < k_2 < k_3 < \dots < k_m = n$ 为所有前 n 个数中等于 f_n 的数，那么容易发现序列 g 为

$$f_{k_1}, f_{k_2}, f_{k_3}, \dots, f_{k_n}, f_{k_1+n}, f_{k_2+n}, \dots, f_{k_n+n}, f_{k_1+2n}, \dots$$

所以可以使用常数次运算完成询问 1 和询问 2，时间复杂度 $O(n + q \log k)$ 。

接下来只考虑情形 2。

2.1 算法一

序列 S 的构造保证了它属于情形 1，按上述算法操作即可。

时间复杂度 $O(n + q \log k)$ ，可以通过子任务 1，期望得分 1。

2.2 算法二

考虑直接截取前若干项进行排序，再去做其余的操作。通过打表发现截取的项数不需要太多就可以获得正确答案。

事实上，通过之后的性质分析，截取的项数在 $O(n^2 k)$ 级别就可以确保正确。

时间复杂度 $O((n^2 k + q) \log nk)$ ，可以通过子任务 2，期望得分 9，结合算法一期望得分 10。

2.3 算法三

首先考虑如何查询某个数 f_u 的排名。

首先若 $f_u \leq f_n$ 那么自然有无限组解，下设 $f_u > f_n$ 。只需计算每个满足 $f_i > f_n$ 的序列 $\{f_{i+kn}\}$ 中有多少个数比 f_u 大即可，由于序列单调递减，可以直接使用二分解决。但由于其形式简单，可以直接解不等式 $f_{i+kn} \geq f_u$ ，具体如下：

$$\frac{a_i + ka_n}{i + kn} \geq \frac{a_u}{u}$$

整理可得

$$(na_u - ua_n)k \leq ua_i - ia_u$$

记 $L = na_u - ua_n$, $R = ua_i - ia_u$ ，式子变为 $Lk \leq R$ 。不难发现 $L > 0$ ，以下分类讨论 R 的正负。

1. 若 $R < 0$ ，那自然无解。

2. 若 $R \geq 0$ ，考虑有 $k \leq \frac{R}{L}$ ，由于 k 是自然数，那么解数为 $\lfloor \frac{R}{L} \rfloor + 1$ 。

若不等式能取等，还需判断这两个数在序列中的先后关系。

那么我们有了查询排名的方法以后，查询第 k 大可以自然想到使用二分。这里分母最大为 10^9 ，那么两个分数之差最小为 10^{-18} ，较为精细地使用 `long double` 的实现可以通过。

时间复杂度为 $O(qn \log k)$ ，可以通过子任务 2 和子任务 3，期望得分 18，结合算法一期望得分 19。

2.4 算法四

考虑如何快速查询排名。

现在需要快速计算一些分数的下取整之和，继续推导式子，将 u 重写为 $v + tn$ ，那么 $a_u = a_v + ta_n$ ，所以有

$$L = na_u - ua_n = n(a_v + ta_n) - (v + tn)a_n = na_v - va_n$$

所以 L 是在 $O(n^2)$ 范围的，无需高精度，并且注意到 L 与 i 无关，故所有分数的分母都相等。如果能将分子都快速加起来再除以分母，那么对高精度的操作次数就能接受了。

首先下取整完再加与加完再下取整这两件事情实际上只差了个进位，换句话说，我们有

$$\left\lfloor \frac{a}{p} \right\rfloor + \left\lfloor \frac{b}{p} \right\rfloor = \left\lfloor \frac{a+b}{p} \right\rfloor - \left\lfloor \frac{(a \bmod p) + (b \bmod p)}{p} \right\rfloor$$

并且这个形式可以推广到 n 元。

注意到 R 是 u 和 a_u 的线性组合，于是分子的加法是 $O(n + \log k)$ 的，而 $R \bmod L$ 可以通过预处理 $u \bmod L$ 与 $a_u \bmod L$ 快速算出。

但这里有个比较致命的细节，就是若存在相等，如何快速判断先后关系。但实际上只需判断 $na_u - ua_n$ 与 $na_i - ia_n$ 的大小关系即可，证明详见后文。

时间复杂度 $O(q(n + \log k))$ ，可以通过子任务 4，期望得分 13，结合算法一和算法三期望得分 32。

2.5 算法五

这里指定了 S 的取法，考虑此时第 k 大有什么规律。

记 h_k 为满足 f_{h_k} 为第 k 大的数。对 h_k 打表可以发现它 $\bmod n$ 有一个长为 $\frac{n(n-1)}{2}$ 的循环节。

有了循环节，我们就可以将 k 缩减到循环节长度的量级了，此时再使用算法三即可求出答案。可是这里的精度要求为 10^{-20} 量级，如何有理有据地进行二分呢？

答案是在分数上二分，我们考虑 Stern-Brocot Tree，由于它具有搜索树性质，那么可以像在平衡树上二分一样对其二分。但是这样做的复杂度并没有保证，原因是一个分母为 n 的分数在 Stern-Brocot Tree 上的深度可以达到 $O(n)$ 。

使用倍增优化，来使连续的向左跳或者向右跳更快，具体来说记录一个参数 t ，每次尝试向左/右跳 2^t 步，成功了就直接跳到对应位置且将 t 自增 1，否则将其自减 1，等到 $t = 0$ 时改为普通的二分。

优化后的复杂度为 $O(\log n)$ ，其中 n 为要找的分数的分母大小，证明考虑归纳，归纳的命题为当向左总共跳了 t 次（这里的次数指倍增跳的成功次数）时，分母至少为 $2^{t/2}$ ，若该命题成立则复杂度显然正确。

开始时往左跳了 0 步，所在位置为 $\frac{1}{1}$ ，命题成立。

否则考虑最后一步，设当前的位置的分数为 $\frac{a}{b}$ ，它当前的前驱（只考虑深度不大于它的分数）为 $\frac{c}{d}$ ，若最后一步为往右跳直接删掉该步，由归纳显然成立。

反之，设从最后一步往前数，连续往左跳的次数是 L 次，那么 $\frac{c}{d}$ 一定在跳的路径中被经过，且它向右跳一次，再向左跳 L 次后即为 $\frac{a}{b}$ 。此处有一个小小的反例为 $\frac{c}{d} = \frac{0}{1}$ ，但其依然满足我们下面的论述。

那么 $\frac{c}{d}$ 的路径中有 $t - L$ 次向左跳，于是 $d \geq 2^{\frac{t-L}{2}}$ ，而由于它连续向左跳了 L 次，那么由倍增以及 Stern-Brocot Tree 的性质，分母至少加了 $2^{\frac{L}{2}}$ 次 d ，所以 $b \geq 2^{\frac{L}{2}}d \geq 2^{\frac{t}{2}}$ ，证毕。

时间复杂度 $O(q(n + \log k))$ ，可以通过子任务 5，期望得分 20，结合算法一、算法三和算法四期望得分 52。

2.6 算法六

考虑推广算法五中发现的循环节性质。

打表发现， $h_k \bmod n$ 永远有一个较短的循环节，但是我们没法直接求出循环节。

考虑暴力，直接截取前若干项，排序后求出 h_k ，再使用 KMP 求出其循环节即可，通过之后的性质可以发现只需截取 $O(n^3)$ 项即可给出正确答案。

时间复杂度 $O(n^3 \log n + q(n + \log k))$ ，可以通过子任务 2 和子任务 6，期望得分 27，结合算法一、算法三、算法四和算法五期望得分 70。

2.7 算法七

需要快速求出循环节以及每个序列在循环节中出现的次数。

转换视角看问题，考虑设 z_k 为复数 $k + a_k i$ ，那么 z_k 对应的辐角 θ_k 满足 $\tan \theta_k = f_k$ 。于是，构建 h_k 的方法如下：

先将 $z_1, z_2, \dots, z_n$ 放入复平面，每次取出一个辐角最大的 z_k ，若有多个取模长最长的，将其变为 $z_{k+n} = z_k + z_n$ 后放回平面。那么第 j 次取出的复数的下标即为 h_j 。

首先对所有 z_k 做变换 $z_k \rightarrow z_k \overline{z_n}$ ，可以发现这个变换保辐角和模长的序。这样一来， z_n 变为整数 $a_n^2 + n^2$ ，便于分析。

接着考虑以原点为中心，将所有复数投影到 $\text{Im } z = 1$ 这条直线上，也即 $a + bi \rightarrow \frac{a}{b} + i$ ，设 z_k 投影后为 $x_k + i$ 。那么求 h_k 的方式变为取 x_k 最小的值，且将其加上 $z_n(\text{Im } z_k)^{-1}$ 。于是对于 x_k ，它被选中 $\text{Im } z_k$ 次之后就会变成 $x_k + z_n$ ，

而且, 我们有 $\operatorname{Re} z_k = nk + a_n a_k < n^2 + a_n^2 = z_n$, 所以开始的 x_k 在 $[0, z_n)$ 之间, 且 x_k 无法再往前移动, 故 z_n 是一个循环节, 且第 k 个序列出现了 $\operatorname{Im} z_k$ 次。

以上只考虑按辐角比较, 那么若辐角相同时又是什么结果呢?

若 z_u 和 z_v 辐角相同, 比较模长时, 我们可以直接对虚部进行比较, 而由于虚部不会变动, 所以相同时的大小关系是确定的, 不会因为操作过程而改变, 至此, 循环节完全确定。

此时我们已经有 $O(n)$ 求循环节的算法了, 时间复杂度 $O(q(n + \log k))$, 期望得分 100。

以下是对之前算法正确性的说明。

注意到 $\operatorname{Im} z_k = na_k - ka_n$, 那么每个数在循环节中的出现次数为 $O(n^2)$, 循环节长度为 $O(n^3)$ 。所以对于算法六只需处理前 $O(n^3)$ 项即可找出循环节。

对于询问1, 容易发现第 k 大的位置不会超过 nk , 因为全局第 k 大可以通过只考虑每个序列的前 k 大得到, 而这些数位于前 nk 个位置。对于询问2, 查询的 k 假设在第 j 个序列, 那么有 $k = j + tn$, 其中 $t = \lfloor \frac{k-1}{n} \rfloor$, 而循环节长度是 $O(n^3)$ 的, 那么它的排名若非正无穷, 则一定为 $O(n^2k)$ 量级。于是, 对于算法二, 只需处理前 $O(n^2k)$ 项即可正确回答询问。

算法四中判断的正是这里虚部的大小。

3 总结

本文从该题中含有的单调性入手, 给出了一个基于二分的解法, 又通过题目的性质, 从特殊到一般地发现了循环节这一特征, 最后的解法中, 巧妙地转换了视角, 从较为几何的角度给出了循环节的本质, 从而快速求解。本题的思维难度较高, 考察了选手发现性质的能力, 是笔者认为不错的题目。

4 参考资料

[1] https://en.wikipedia.org/wiki/Stern-Brocot_tree, Wikipedia