

「茧」解题报告

大连市第二十四中学 李嘉佑

♪² 题目大意

有若干排棋子, 各排所含棋子数已知。

Lily 与 Kaguya 轮流进行如下操作, Lily 先手:

- 选择非空的一排棋子 (所含棋子数记为 x);
选择一个正整数 y ($1 \leq y \leq \sqrt[k]{x}$), 从中移除 y 枚棋子。

无法操作者负。换句话说, 移除最后一枚棋子的人获胜。

多次询问一个局面下谁有必胜策略。

对局面的描述形如: 给定 $n, a_1 \dots n$, 表示有 $\sum a_i$ 排棋子, 各排所含棋子数分别为 $1 \dots a_1, 1 \dots a_2, \dots, 1 \dots a_n$ 。

♪² 数据范围

对于所有测试数据保证: $1 \leq k \leq 5, 1 \leq n, \sum n \leq 10^5, 1 \leq a_i < 2^{60}$ 。

每个子任务的具体限制见下表:

子任务编号	n	k	a_i	特殊性质	分值
1	$\sum n \leq 10^5$	$k = 1$	$a_i < 2^{60}$		5
2	$\sum n \leq 10^5$	$2 \leq k \leq 5$	$a_i < 2^{16}$		5
3	$\sum n \leq 10^5$	$2 \leq k \leq 5$	$a_i < 2^{22}$		10
4	$\sum n \leq 10^3, n = 2$	$2 \leq k \leq 3$	$a_i < 2^{32}$	A	10
5	$\sum n \leq 10^3$	$2 \leq k \leq 5$	$a_i < 2^{32}$	A	10
6	$\sum n \leq 10^5$	$k = 2$	$a_i < 2^{60}$	A	10
7	$\sum n \leq 10^5$	$k = 3$	$a_i < 2^{60}$	A	10
8	$\sum n \leq 10^3$	$k = 2$	$a_i < 2^{32}$		10
9	$\sum n \leq 10^3$	$k = 3$	$a_i < 2^{32}$		10
10	$\sum n \leq 10^5$	$1 \leq k \leq 5$	$a_i < 2^{60}$		20

特殊性质 A: 保证 $2 \mid n$, 且 $a_{2k-1} = a_{2k} - 1$ ($1 \leq k \leq \frac{n}{2}$)。

♪² 解题过程

本部分中, 约定:

- $\oplus$ 表示按位异或运算;
- V 表示 $\max a_i$;
- $g(x)$ 表示一排 x 个棋子的 Grundy value;
- $h(x)$ 表示 $g(x)$ 的前缀异或和, 即 $h(x) = \bigoplus_{i=1}^x g(i)$ 。

根据 Sprague-Grundy theorem, 先手必胜当且仅当 $\bigoplus_{i=1}^n h(a_i) \neq 0$ 。

于是, 问题转化为快速求出 g 的前缀异或和, 即快速对 h 进行单点求值。

♪³ 算法 1

显然, $g(x) = x \ (k = 1)$ 。

$$\bigoplus_{i=0}^3 4n + i = 0 \quad (n \in \mathbb{N})$$

$$\therefore h(x) = \bigoplus_{i=0}^{x \bmod 4} x - i \quad (k = 1)$$

按上式计算即可, 时间复杂度 $O(\sum n)$ 。

期望通过 Subtask 1, 得分 5。

下面的分析中将默认 $k \neq 1$ 以规避一些特殊情况。

♪³ 算法 2

$$g(x) = \text{mex}\{g(x - d) \mid 1 \leq d \leq \sqrt[k]{x}\}$$

模拟上式, 计算出 $g(1) \dots g(V)$, 再预处理出前缀异或和即可。

时间复杂度 $O((k + V)\sqrt[k]{V} + \sum n)$ 。

结合算法 1, 期望通过 Subtask 1, 2, 得分 10。

♪³ 算法 3

观察算法 2 得出的结果, 可以注意到:

$$g(x) = \begin{cases} \sqrt[k]{x} & \sqrt[k]{x} \in \mathbb{Z} \\ g(x - \lceil \sqrt[k]{x} \rceil) & \sqrt[k]{x} \notin \mathbb{Z} \end{cases} \quad (1)$$

下面将归纳地给出证明。显然上式对 $g(0), g(1)$ 成立。

设 $t > 1$, 若上式对所有 $g(y)$ ($0 \leq y < t$) 成立, 尝试证明上式对 $g(t)$ 成立。

记 $S(x) = \{g(y) \mid x - \sqrt[k]{x} \leq y < x\}$ ($x \in \mathbb{N}^*$), 记 $r = \lceil \sqrt[k]{t} \rceil$ 。

根据 Sprague-Grundy theorem, $g(x) = \text{mex } S(x)$ 。

设 p, q 满足 $t - r \leq p < q < t$, 记 $T = \{g(y) \mid t - r \leq y < t\}$ 。

由于 $t - q = t - 1 - q + 1 \geq \sqrt[k]{t-1} - \sqrt[k]{q} + 1 \geq \lceil \sqrt[k]{t} \rceil - \sqrt[k]{q}$,

故 $q - p \leq q - (t - r) = r - (t - q) \leq r - r + \sqrt[k]{q} = \sqrt[k]{q}$ 。

于是, $\forall t - r \leq p < q < t, g(p) \in S(q), g(p) \neq g(q)$ 。

由于 (1) 式对所有 $g(y)$ ($0 \leq y < t$) 成立, 故 $\forall y < t, g(y) < r$ 。

故 $T = \{g(y) \mid t - r \leq y < t\} = [0, r) \cap \mathbb{Z}$ 。

若 $\sqrt[k]{t} \in \mathbb{Z}$, 则 $S(t) = T, g(t) = \text{mex } S(t) = r$;

若 $\sqrt[k]{t} \notin \mathbb{Z}$, 则 $S(t) = T \setminus \{g(t - r)\}, g(t) = \text{mex } S(t) = g(t - r)$ 。

根据数学归纳法, (1) 式得证。

递推出 $g(1) \dots g(V)$, 再预处理出前缀异或和即可。

时间复杂度 $O(k\sqrt[k]{V} + V + \sum n)$ 。

结合算法 1, 期望通过 Subtask 1, 2, 3, 得分 20。

♪³ 算法 4

性质 A 是将求 g 的前缀异或和变为对 g 进行单点求值。

设 $g(r, d) = g(r^k + d)$ ($0 \leq d < (r+1)^k - r^k$)。

根据算法 3 中 (1) 式, $g(r, d) = g(r, d \bmod (r+1))$ 。

(如无特殊说明, 下文中 $g(r, *)$ 的循环节均指长为 $r+1$ 的循环节。)

递归计算, 不难得到一个时间复杂度 $O(k\sqrt[k]{V} \sum n)$ 的解法。

结合算法 1, 3, 期望通过 Subtask 1 - 5, 得分 40。

♪³ 算法 5

由于 $g(r, d) = g(r, d \bmod (r+1))$, 故只需要讨论 $0 \leq d \leq r$ 的情况。

$$\begin{aligned} r^k - (r-1)^k &\equiv (-1)^{k+1} \pmod{r} \\ \therefore g(r, d) &= \begin{cases} r & d = 0 \\ g(r-1, (d-1 + (-1)^{k+1}) \bmod r) & 1 \leq d \leq r \end{cases} \end{aligned}$$

即:

$$g(r, d) = \begin{cases} r & d = 0 \\ g(r-1, d \bmod r) & 2 \nmid k, 1 \leq d \leq r \\ g(r-1, r-1) & 2 \mid k, d = 1 \\ g(r-1, d-2) & 2 \mid k, 2 \leq d \leq r \end{cases}$$

递归下去, 有:

$$g(r, d) = \begin{cases} r & d = 0 \\ d-1 & 2 \nmid k, 1 \leq d \leq r \\ g(r-1, r-1) & 2 \mid k, d = 1 \\ g(r - \lfloor \frac{d}{2} \rfloor, d \bmod 2) & 2 \mid k, 2 \leq d \leq r \end{cases}$$

按上式递归的话, 求出某个特定的 $g(r, d)$ 的时间复杂度为 $O(k \log V)$ 。

于是可得到一个时间复杂度 $O((k \log V) \sum n)$ 的解法。

尝试继续递归下去, 有 $(\alpha(x))$ 表示 x 二进制后导零的数目):

$$\begin{aligned} g(r, 1) &= g(r-1, r-1) = g(\lfloor \frac{r}{2} \rfloor, (r+1) \bmod 2) \\ &= g(\frac{r}{2^{\alpha(r)}}, 1) = g(\lfloor \frac{r}{2^{\alpha(r)+1}} \rfloor, 0) = \lfloor \frac{r}{2^{\alpha(r)+1}} \rfloor \end{aligned}$$

故:

$$g(r, d) = \begin{cases} r & d = 0 \\ d-1 & 2 \nmid k, 1 \leq d \leq r \\ \lfloor \frac{r}{2^{\alpha(r)+1}} \rfloor & 2 \mid k, d = 1 \\ g(r - \lfloor \frac{d}{2} \rfloor, d \bmod 2) & 2 \mid k, 2 \leq d \leq r \end{cases} \quad (2)$$

按上式计算的话, 求出某个特定的 $g(r, d)$ 的时间复杂度为 $O(1)$ 。

于是可得到一个时间复杂度 $O(k \sum n)$ 的解法。

结合算法 1, 3, 期望通过 Subtask 1 - 7, 得分 60。

3 算法 6

记 $s(r, d) = \sum_{i=0}^d g(r, i)$, $s(r) = s(r, (r+1)^k - r^k - 1)$, $\sigma(r) = \sum_{i=0}^r s(i)$ 。

记 $h_0(r) = \bigoplus_{i=0}^r g(i, 0) = \bigoplus_{i=0}^r i$, $h_1(r) = \bigoplus_{i=1}^r g(i, 1) = \bigoplus_{i=0}^{r-1} g(i, i)$ 。

$h_0(r)$ 可类似算法 1 地 $O(1)$ 求出。

$h_1(r) = \bigoplus_{i=1}^r g(i, 1)$ 可通过枚举 i 的二进制后导零数目 $O(\log V)$ 地求出。

将 $1 \dots x$ 拆分为两部分, 有:

$$h(x) = s(\lfloor \sqrt[k]{x} \rfloor, x - \lfloor \sqrt[k]{x} \rfloor^k) \oplus \sigma(\lfloor \sqrt[k]{x} \rfloor - 1)$$

于是, 问题转化为快速求出 $s(r, d)$ 与 $\sigma(r)$ 。

由于 $g(r, *)$ 有长为 $r+1$ 的循环节, 故 $s(r, *)$ 有长为 $2r+2$ 的循环节。

于是 $s(r, d) = s(r, d \bmod (2r+2))$, 直接求出涉及到的所有 $g(r, i)$ 即可 $O(r)$ 地求出某个特定的 $s(r, d)$ 。

下面尝试分析 $s(r)$ 的性质。

由于 $(r+1)^k - r^k \equiv (-1)^{k+1} \pmod{r+1}$, 故 $g(r, *)$ 形如若干完整的循环节多一项或少一项。

首先考虑 $g(r, *)$ 的完整循环节数目的奇偶性, 略去简单且与本题无关的推导过程, 有:

$$\frac{(r+1)^k - r^k - (-1)^{k+1}}{r+1} \equiv kr \pmod{2}$$

由于 $g(r, *)$ 有长为 $r+1$ 的循环节, 不难得出:

- 若 $g(r, *)$ 形如若干完整的循环节多了一项 (即 $2 \nmid k$), 则多的那项必然等于 $g(r, 0)$;
- 若 $g(r, *)$ 形如若干完整的循环节少了一项 (即 $2 \mid k$), 则少的那项必然等于 $g(r, r)$;

于是, 有:

$$s(r) = \begin{cases} g(r, r) & 2 \mid k \\ g(r, 0) & 2 \nmid k, 2 \mid r \\ g(r, 0) \oplus h_0(r) & 2 \nmid k, 2 \nmid r \end{cases}$$

即:

$$s(r) = \begin{cases} \lfloor \frac{r+1}{2^{\alpha(r+1)+1}} \rfloor & 2 \mid k \\ r & 2 \nmid k, 2 \mid r \\ r \oplus [r \bmod 4 = 1] & 2 \nmid k, 2 \nmid r \end{cases} \quad (3)$$

于是可 $O(1)$ 地求出某个特定的 $s(r)$, 即可 $O(r)$ 地求出某个特定的 $\sigma(r)$ 。

至此, 可以 $O((k + \sqrt[k]{V}) \sum n)$ 的时间复杂度解决原问题。

结合算法 1, 3, 期望通过 Subtask 1 - 5, 8 - 9, 得分 60。

结合算法 1, 3, 5, 期望通过 Subtask 1 - 9, 得分 80。

♪³ 算法 7

在算法 6 的基础上进行优化。

首先, 考虑更快速地求出 $s(r, d)$ 。在算法 6 中已经发现 $s(r, d)$ 有长度为 $2r+2$ 的循环节。

考虑将 $r < d < 2r+2$ 的情况转化为 $0 \leq d \leq r$ 的情况。进一步利用循环节, 可以得到:

$$s(r, d) = h_0(r) \oplus s(r, d - r - 1) \quad (r < d < 2r + 2)$$

于是, 只需要考虑 $0 \leq d \leq r$ 的情况。此时, 考虑按算法 5 中的 (2) 式递归一步, 有:

$$s(r, d) = \begin{cases} h_0(r) \oplus h_0(r - \lfloor \frac{d}{2} \rfloor - 1) \oplus h_1(r) \oplus h_1(r - \lfloor \frac{d+1}{2} \rfloor) & 2 \mid k, 0 \leq d \leq r \\ r & 2 \nmid k, d = 0 \\ r \oplus h_0(d - 1) & 2 \nmid k, 1 \leq d \leq r \end{cases}$$

于是, 可 $O(\log V)$ 地求出某个特定的 $s(r, d)$ 。

其次, 考虑更快速地求出 $\sigma(r)$ 。用算法 6 中 (3) 式展开得:

$$\sigma(r) = \begin{cases} h_1(r+1) & 2 \mid k \\ h_0(r) \oplus [1 \leq r \bmod 8 \leq 4] & 2 \nmid k \end{cases}$$

于是, 可 $O(\log V)$ 地求出某个特定的 $\sigma(r)$ 。

至此, 可以 $O((k + \log V) \sum n)$ 的时间复杂度解决原问题。

结合算法 1, 期望通过 Subtask 1 - 10, 得分 100。

♪² 参考资料

无。

♪² 致谢

- 感谢王孜研同学在本题命制过程中提出的诸多富有价值的意见。