Numbers 解题报告

成都外国语学校 张隽恺

2021.10.2

1 题目大意

给定 n 以及 n 组正整数 (l_i, v_i)。有 n 个整数 $x_1, x_2, ..., x_n$，初始时这些整数都为 0。循环进行如下操作直到停止：

随机选择一个 $[1, n]$ 间的正整数 k，其中选择 t 的概率为 $p_t = \frac{v_t}{\sum_{i=1}^{n} v_i}$。然后将 x_k 变为 $(x_k + 1) \bmod l_k$。接下来，记录序列 $(x_1, x_2, ..., x_n)$。如果记录了 $(0, 0, ..., 0)$ 则停止操作。

求停止时，记录过的不同序列种类数的期望，对给定质数 mod 取模。保证答案在模 mod 意义下有定义且对于每个 l_i，模 mod 意义下的 l_i 阶原根 d_i 存在且给出。

同时，保证对于任意一组满足 $\forall 1 \le i \le n, 0 \le x_i < l_i$ 的整数序列 $(x_1, ..., x_n)$，满足 $\sum_{i=1}^{n} p_i a_i^{x_i} \equiv 1 (\mathrm{mod} mod)$ 当且仅当所有 $x_i = 0$。

保证 v 随机生成，且答案在模 mod 意义下有定义。在满足题目所述条件的情况下，运算中出现值在模 mod 意义下无法表示的概率足够小，可以认为不用考虑出现值在模 mod 意义下无法表示的情况。

2 数据范围与限制

记 $m = \prod_{i=1}^{n} l_i$。

对于所有数据，$1 \le n, m \le 5 \times 10^5, 2 \le l_i, 1 \le v_i \le 10^6, 1 \le d_i \le mod - 1, 9 \times 10^8 \le mod \le 1.05 \times 10^9$

子任务编号	子任务分值	特殊限制
1	2	$n = 1$
2	8	$m \le 8$
3	10	$m \le 100$
4	8	$n = 2, mod = 998244353, m \le 2^{10}$
5	8	$l_i = 2, m \le 2^{10}$
6	12	$m \le 2^{10}$
7	8	$m \le 2^{13}$
8	6	$n = 2, mod = 998244353$
9	8	$mod = 998244353$
10	12	$l_i = 2$
11	12	$l_i \le 50$
12	6	无特殊限制

时间限制：4 秒

空间限制：1024 MB

3 解题过程

3.1 算法 1

对于 $n = 1$ 的情况，每一步操作都只会操作 x_1，x_1 会从 0 依次增加到 $l_1 - 1$，然后变为 0 停止。此时答案为 l_1。

复杂度 $O(1)$，可以通过子任务 1，期望得分 2 分。

3.2 算法 2

设 $f_{S,T}$ 表示当前 $(x_1, ..., x_n)$ 的值为 S，之前所有记录过的 $(x_1, ..., x_n)$ 的集合为 T ，继续操作直到停止时种类数的期望值。

对于一个 $f_{S,T}$，如果 $S = (0, 0, ..., 0)$，则 $f_{S,T} = |T|$，否则枚举下一次选择的数的情况，可以得到一个关于 f 的方程。

对每个 $f_{S,T}$ 都考虑一次，可以得到一个 f 的方程组，然后可以高斯消元求出 f。

不同的 S 共有 m 个,因此不同的 T 有 2^m 个,该算法的复杂度为 $O(m^3 2^{3m})$。可能的状态数少于 $m \times 2^m$ 的一半，因此该算法可以通过子任务 2，结合算法 1 期望得分 10 分。

3.3 算法 3

根据期望线性性，记录过的序列 $(x_1, ..., x_n)$ 的种类数的期望，等价于每一种序列 $(x_1, ..., x_n)$ 被记录的概率的和。

考虑计算一个序列 $(y_1, ..., y_n)$ 不被记录的概率。$(0, ..., 0)$ 一定会被记录，只需要计算 $(y_1, ..., y_n)$ 不为 0 的情况。设 $f_{(z_1,...,z_n)}$ 表示当前 $(x_1, ..., x_n)$ 的值为 $(z_1, ..., z_n)$ 且之前没有记录 $(y_1, ..., y_n)$，继续操作停止时不记录 $(y_1, ..., y_n)$ 的概率。

如果 $(z_1,...,z_n)=(y_1,...,y_n)$，则 $f_{(z_1,...,z_n)}=0$。如果 $(z_1,...,z_n)=(0,...,0)$，则看作 $f_{(z_1,...,z_n)}=1$。否则枚举下一步可能的操作得到方程。这样可以得到关于 f 的方程组，从而解出 f。由于初始在 $(0,...,0)$ 时不会停止但方程中看作 $f_{(0,...,0)}=1$，计算概率时可以枚举第一步操作的情况并求和概率。

该算法需要进行 $O(m)$ 次高斯消元，复杂度为 $O(m^4)$。结合算法 1 可以通过子任务 $2,3$，期望得分 20 分。

3.4 算法 4

为了便于描述接下来的算法，定义 $(a_1,a_2,...,a_n)+(b_1,b_2,...,b_n)=((a_1+b_1)\bmod l_1,...,(a_n+b_n)\bmod l_n)$。下文中所有形如 $\sum_{(b_1,...,b_n)}$ 的求和默认包含 $\forall i,0\le b_i<l_i$ 的限制。

记 $T_i=(\delta_{i1},...,\delta_{in})$，其中 δ_{ij} 在 $i=j$ 时为 1，否则为 0，即 T_i 的第 i 个元素为 1 ，其余元素为 0。则算法 3 中关于 f 的方程组为：

$$f_{(z_1,...,z_n)}=\begin{cases}1 & (z_1,...,z_n)=(0,...,0)\\ 0 & (z_1,...,z_n)=(y_1,...,y_n)\\ \sum_{i=1}^{n}p_i*f_{(z_1,...,z_n)+T_i} & otherwise\end{cases}$$

定义 $g_{(a_1,...,a_n)}$ 为：

$$g_{(a_1,...,a_n)}=\begin{cases}p_i & (\exists i,(a_1,...,a_n)=T_i)\\ 0 & otherwise\end{cases}$$

则 f 的方程中的第三种情况可以看成一种卷积运算。定义：

$$(f+g)_{(a_1,...,a_n)}=f_{(a_1,...,a_n)}+g_{(a_1,...,a_n)}$$
$$(f\times g)_{(a_1,...,a_n)}=\sum_{(b_1,...,b_n)+(c_1,...,c_n)=(a_1,...,a_n)}f_{(b_1,...,b_n)}g_{(c_1,...,c_n)}$$

考虑将前两种情况看成在第三种情况的方程的一侧加上一个常数。设 $x=1-$

$\sum_{i=1}^{n} p_i * f_{(0,...,0)+T_i}, y = -\sum_{i=1}^{n} p_i * f_{(y_1,...,y_n)+T_i}$。定义 $h_{(a_1,...,a_n)}$ 为：

$$h_{(a_1,...,a_n)} = \begin{cases} x & (a_1,...,a_n) = (0,...,0) \\ y & (a_1,...,a_n) = (y_1,...,y_n) \\ 0 & otherwise \end{cases}$$

则方程组可以看成 $f = f \times g + h$。同时，因为 $\sum_{i=1}^{n} p_i * f_{(0,...,0)+T_i}$ 为从 $(0,...,0)$ 开始，停止时不记录 $(y_1,...,y_n)$ 的概率，因此 x 即为记录 $(y_1,...,y_n)$ 的概率。

在 $d_i = 2$ 时，$f \times g$ 即为异或卷积。由 FWT 的性质，定义：

$$FWT(f)_{(a_1,...,a_n)} = \sum_{(b_1,...,b_n)} f_{(b_1,...,b_n)} \prod_{i=1}^{n} (-1)^{a_i b_i}$$

则 $FWT(f)_{(a_1,...,a_n)} = FWT(f)_{(a_1,...,a_n)} * FWT(g)_{(a_1,...,a_n)} + FWT(h)_{(a_1,...,a_n)}$。

考虑 $(a_1,...,a_n) = (0,...,0)$ 的情况，此时 $FWT(g)_{(0,...,0)} = \sum_{i=1}^{n} p_i = 1$，因此可以得到 $FWT(h)_{(0,...,0)} = 0$。又因为 $FWT(h)_{0,...,0} = x + y$，因此 $y = -x$。

对于 $(a_1,...,a_n) \neq (0,...,0)$ 的情况，$FWT(g)_{(a_1,...,a_n)} = \sum_{i=1}^{n} p_i (-1)^{a_i} < 1$。因为 h 中只有 $x, -x, 0$，因此存在常数 c 使得 $FWT(h)_{(a_1,...,a_n)} = cx$，进一步可以得到存在常数 c 使得 $FWT(f)_{(a_1,...,a_n)} = cx$。

此时可以用 x 表示所有满足 $(a_1,...,a_n) \neq (0,...,0)$ 的 $FWT(f)_{(a_1,...,a_n)}$，但无法确定 $FWT(f)_{(0,...,0)}$ 的值。

但因为 $f_{(0,...,0)} = 1, f_{(y_1,...,y_n)} = 0$，由 FWT 的性质，有：

$$f_{(a_1,...,a_n)} = \frac{1}{2^n} \sum_{(b_1,...,b_n)} FWT(f)_{(b_1,...,b_n)} \prod_{i=1}^{n} (-1)^{a_i b_i}$$

$$f_{(0,...,0)} - f_{(y_1,...,y_n)} = \frac{1}{2^n} \sum_{(b_1,...,b_n)} FWT(f)_{(b_1,...,b_n)} (1 - \prod_{i=1}^{n} (-1)^{y_i b_i})$$

此时 $FWT(f)_{(0,...,0)}$ 的系数为 0，因此可以得到形如 $cx = 1$ 的方程，进而解出 x。

对于一个 $(y_1,...,y_n)$ 进行上述过程的复杂度为 $O(m \log m)$，因此该算法的

复杂度为 $O(m^2 \log m)$。可以通过子任务 5 ，结合算法 1 与算法 3 期望得分 28 分。

3.5 算法 5

FWT 相当于每一维上进行长度为 2 的 DFT，因此容易将算法 4 的操作扩展到每一维大小不为 2 的情况。

定义：

$$DFT(f)_{(a_1,\ldots,a_n)} = \sum_{(b_1,\ldots,b_n)} f_{(b_1,\ldots,b_n)} \prod_{i=1}^{n} \omega_{l_i}^{a_i b_i}$$

通过与 FWT 类似的方式，可以证明 $DFT(f)$ 满足：

$$DFT(f)_{(a_1,\ldots,a_n)} = DFT(f)_{(a_1,\ldots,a_n)} * DFT(g)_{(a_1,\ldots,a_n)} + DFT(h)_{(a_1,\ldots,a_n)}$$

同时根据 DFT 的性质，逆变换为：

$$f_{(a_1,\ldots,a_n)} = \frac{1}{\prod_{i=1}^{n} l_i} \sum_{(b_1,\ldots,b_n)} DFT(f)_{(b_1,\ldots,b_n)} \prod_{i=1}^{n} \omega_{l_i}^{-a_i b_i}$$

因此使用与算法 4 相同的做法即可对于每一个 $(y_1, ..., y_n)$ 得到记录 $(y_1, ..., y_n)$ 的概率。

此时算法的复杂度瓶颈在于对于 g, h 求出 $DFT(g), DFT(h)$。因为 g 只在所有 T_i 位置值非零，h 只在两个位置值非零。因此容易在 $O(nm) = O(m \log m)$ 的复杂度内求出。

该算法复杂度为 $O(m^2 \log m)$，可以通过子任务 $2 \sim 6$，结合算法 1 期望得分 48 分。

通过一些实现上的优化（例如按照顺序枚举 $(a_1, ..., a_n)$ 每一维的值，在枚举的过程中计算 DFT 的值），可以将复杂度优化至 $O(m^2)$，这样可以通过子任务 $2 \sim 7$，结合算法 1 期望得分 56 分。

3.6 算法 6

考虑直接解出上一个算法中的 x，对于 $(y_1, ..., y_n)$ 可以得到：

$$DFT(g)_{(a_1,...,a_n)} = \sum_{i=1}^{n} p_i \omega_{l_i}^{a_i}$$

$$DFT(h)_{(a_1,...,a_n)} = (1 - \prod_{i=1}^{n} \omega_{l_i}^{a_i y_i})x$$

因此可以得到：

$$DFT(f)_{(a_1,...,a_n)} = \frac{1 - \prod_{i=1}^{n} \omega_{l_i}^{a_i y_i}}{1 - \sum_{i=1}^{n} p_i \omega_{l_i}^{a_i}} x((a_1, ..., a_n) \neq (0, ..., 0))$$

$$1 = f_{(0,...,0)} - f_{(a_1,...,a_n)} = \frac{1}{\prod_{i=1}^{n} l_i} \sum_{(b_1,...,b_n)} DFT(f)_{(b_1,...,b_n)} (1 - \prod_{i=1}^{n} \omega_{l_i}^{-b_i y_i})$$

$$1 = \frac{1}{\prod_{i=1}^{n} l_i} \sum_{(b_1,...,b_n) \neq (0,...,0)} \frac{(1 - \prod_{i=1}^{n} \omega_{l_i}^{b_i y_i})(1 - \prod_{i=1}^{n} \omega_{l_i}^{-b_i y_i})}{1 - \sum_{i=1}^{n} p_i \omega_{l_i}^{b_i}} x$$

$$\frac{1}{x} = \frac{1}{\prod_{i=1}^{n} l_i} \sum_{(b_1,...,b_n) \neq (0,...,0)} \frac{(1 - \prod_{i=1}^{n} \omega_{l_i}^{b_i y_i}) + (1 - \prod_{i=1}^{n} \omega_{l_i}^{-b_i y_i})}{1 - \sum_{i=1}^{n} p_i \omega_{l_i}^{b_i}}$$

定义 $-(b_1, ..., b_n) = ((l_1 - b_1) \bmod l_1, ..., (l_n - b_n) \bmod l_n)$，记：

$$\begin{aligned} s_{(a_1,...,a_n)} &= \sum_{(b_1,...,b_n) \neq (0,...,0)} \frac{1 - \prod_{i=1}^{n} \omega_{l_i}^{b_i a_i}}{1 - \sum_{i=1}^{n} p_i \omega_{l_i}^{b_i}} \\ &= \sum_{(b_1,...,b_n) \neq (0,...,0)} \frac{1}{1 - \sum_{i=1}^{n} p_i \omega_{l_i}^{b_i}} - \sum_{(b_1,...,b_n) \neq (0,...,0)} \frac{\prod_{i=1}^{n} \omega_{l_i}^{b_i a_i}}{1 - \sum_{i=1}^{n} p_i \omega_{l_i}^{b_i}} \end{aligned}$$

则答案等于：

$$1 + \prod_{i=1}^{n} l_i \sum_{(a_1,...,a_n) \neq (0,...,0)} \frac{1}{s_{(a_1,...,a_n)} + s_{-(a_1,...,a_n)}}$$

因此求出 s 后即可 $O(m \log m)$ 得到答案。考虑求 s 的部分，其中第一部分

$$\sum_{(b_1,\ldots,b_n)\neq(0,\ldots,0)} \frac{1}{1-\sum_{i=1}^{n} p_i\omega_{l_i}^{b_i}}$$

为与 $(a_1,...,a_n)$ 无关的定值，可以 $O(m\log m)$ 求出。对于第二部分，定义：

$$t_{(a_1,\ldots,a_n)} = \frac{1}{1-\sum_{i=1}^{n} p_i\omega_{l_i}^{b_i}}$$

则第二部分等于 $DFT(t)$。

在 $l_i = 2$ 时，DFT 等价于 FWT，因此可以做到 $O(m\log m)$ 复杂度，可以通过子任务 $5, 10$，结合算法 $1, 5$ 期望得分 68 分。

在 $l_i \geq 2$ 时，可以沿用 FWT 的思想，依次对每一维做变换。一维上的变换相当于求 $t'_x = \sum_{y=0}^{l_i-1} t_y\omega_{l_i}^{xy}$，直接枚举 y 求和，总复杂度为 $O(m*(\sum l_i))$，可以通过子任务 $2\sim 7, 10, 11$，结合算法 1 期望得分 80 分。

3.7 算法 7

算法 6 的复杂度瓶颈在于 $O(l_i^2)$ 求出所有的 $t'_x = \sum_{y=0}^{l_i-1} t_y\omega_{l_i}^{xy}$，即任意长度的 DFT。

使用 Bluestein 算法即可在 $O(l_i\log l_i)$ 的复杂度内求出。因此可以将算法 6 的复杂度优化至 $O(m\log m)$。

对于 $mod = 998244353$ 的情况，可以使用 NTT 实现 Bluestein 中的卷积，这样可以通过子任务 $4, 8, 9$，结合算法 $1, 6$ 期望得分 94 分。

对于 p 任意的情况，可以使用任意模数 FFT 的方法。可以通过所有子任务，期望得分 100 分。

由于 p 随机生成，可以看作 t 随机生成，且题目保证 t 的分母在模意义下不为 0。可以近似看作 s 随机生成。若出现 $\frac{1}{s_{(a_1,\ldots,a_n)}+s_{-(a_1,\ldots,a_n)}}$ 的分母模意义下为 0 的情况，则只有当所有分母模意义为 0 的部分相加，分子为 mod 的倍数时，答案可能在模意义下有意义。因此这种情况出现的概率近似于不大于存在一个分母模意义为 0 的概率乘上 $\frac{1}{mod}$，即不大于 $\frac{m}{mod^2}$，在本题的范围下，这个概率小于 10^{-10}，因而可以不考虑这种情况。

3.8 算法 8

以下为笔者的一个从生成函数角度出发的思路，该思路得到了与之前的算法相同的答案。但笔者认为该思路可能存在不严谨，因此在这里仅供参考。

为了避免一些问题，所有求 $F(1)$ 的步骤可以看成求 $\lim_{x\to 1^-} F(x)$。

设 $h_{i,j,x}$ 表示对第 i 个数进行 x 次操作后等于 j 概率乘上 p_i^x，$H_{i,j}(x)=\sum_{l\geq 0}\frac{h_{i,j,l}}{l!}x^l$，则：

$$H_{i,j}(x)=\sum_{k\geq 0}[k \bmod v_i=j]\frac{(\frac{p_i}{\sum p_i})^k}{k!}=\frac{1}{v_i}\sum_{k=0}^{v_i-1}\omega_{v_i}^{-jk}e^{\omega_{v_i}^k\frac{p_i}{\sum p_i}x}$$

设 $f_{i,(a_1,...,a_n)}$ 表示操作 i 次后，所有数字为 $(a_1,...,a_n)$ 的概率。$F_{(a_1,...,a_n)}(x)=\sum_{i\geq 0}f_{i,(a_1,...,a_n)}x^i, G_{(a_1,...,a_n)}=\sum_{i\geq 0}\frac{f_{i,(a_1,...,a_n)}}{i!}x^i$。则可以得到：

$$G_{(c_1,...,c_n)}(x)=\frac{1}{\prod_{i=1}^n v_i}\prod_{i=1}^n H_{i,c_i}(x)$$

$$=\frac{1}{\prod_{i=1}^n v_i}\sum_{(a_1,...,a_n)}(\prod_{i=1}^n\omega_{v_i}^{-a_ic_i})*e^{\sum_{i=1}^n\omega_{v_i}^{a_i}\frac{p_i}{\sum p_i}x}$$

$$F_{(c_1,...,c_n)}(x)=\frac{1}{\prod_{i=1}^n v_i}\sum_{(a_1,...,a_n)}\frac{\prod_{i=1}^n\omega_{v_i}^{-a_ic_i}}{1-(\sum_{i=1}^n\omega_{v_i}^{a_i}\frac{p_i}{\sum p_i})x}$$

在接下来计算贡献的部分中，可以认为 $(a_1,...,a_n)\neq(0,...,0)$。

设 $q_{i,(a_1,...,a_n)}$ 表示操作 i 次后当前的所有数字为 $(a_1,...,a_n)$，且在此之前的每次操作后数都不为 $(a_1,...,a_n)$ 的概率，设 $Q_{(a_1,...,a_n)}(x)=\sum_{i\geq 0}q_{i,(a_1,...,a_n)}x^i$。

考虑容斥，枚举最后一次出现 $(a_1,...,a_n)$ 位置减去，则可以得到：

$$Q_{(a_1,...,a_n)}(x)=F_{(a_1,...,a_n)}(x)-Q_{a_1,...,a_n}(x)*(F_{(0,...,0)}-1)$$

$$Q_{(a_1,...,a_n)}(x)=\frac{F_{(a_1,...,a_n)}(x)}{F_{(0,...,0)}(x)}$$

再设 $s_{i,(a_1,...,a_n)}$ 表示操作 i 次后当前所有数字为 $(0,...,0)$，且过程中每次操作后数都不为 $(a_1,...,a_n)$ 的概率。设 $S_{(a_1,...,a_n)}(x)=\sum_{i\geq 0}s_{i,(a_1,...,a_n)}x^i$。

同样考虑容斥，枚举最后一次出现 $(a_1,...,a_n)$ 位置减去，对于之后的部分考虑反向的操作，可以得到：

$$S_{(a_1,...,a_n)}(x)=F_{(0,...,0)}(x)-F_{(a_1,...,a_n)}(x)*Q_{-(a_1,...,a_n)}(x)$$

最后设 $t_{i,(a_1,...,a_n)}$ 表示表示操作 i 次后当前所有数字为 $(0,...,0)$，且过程中每次操作后数都不为 $(a_1,...,a_n)$，除去最后一次操作外每次操作后不为 $(0,...,0)$ 的概率，$T_{(a_1,...,a_n)}(x)=\sum_{i\geq 0}t_{i,(a_1,...,a_n)}x^i$。认为 $T_{0,(a_1,...,a_n)}=0$。

由容斥得：

$$T_{(a_1,...,a_n)}(x)=S_{(a_1,...,a_n)}(x)-T_{(a_1,...,a_n)}(x)*(S_{(a_1,...,a_n)}(x)-1)-1$$
$$T_{(a_1,...,a_n)}(x)=1-\frac{1}{S_{(a_1,...,a_n)}(x)}$$

因此停止时没有记录 $(a_1,...,a_n)$ 的概率为 $T_{(a_1,...,a_n)}(1)$。由期望线性性答案等于 $1+\sum_{(a_1,...,a_n)\neq(0,...,0)}\frac{1}{S_{(a_1,...,a_n)}(x)}$。令

$$vl_{(a_1,...,a_n)}=F_{(0,...,0)}(1)-F_{(a_1,...,a_n)}(1)$$
$$=\frac{1}{\prod_{i=1}^n v_i}*\sum_{(b_1,...,b_n)\neq(0,...,0)}\frac{1-\prod_{i=1}^n\omega_{v_i}^{-a_ib_i}}{1-\sum_{i=1}^n\omega_{v_i}^{b_i}\frac{p_i}{\sum p_i}}$$

则有：

$$S_{(a_1,...,a_n)}(x)=F_{(0,...,0)}(x)-F_{(a_1,...,a_n)}(x)*\frac{F_{-(a_1,...,a_n)}(x)}{F_{(0,...,0)}(x)}$$
$$S_{(a_1,...,a_n)}(x)=$$
$$\frac{(F^2_{(0,...,0)}(x)-(F_{(0,...,0)}(x)-vl_{(a_1,...,a_n)})(F_{(0,...,0)}(x)-vl_{-(a_1,...,a_n)}))}{F_{(0,...,0)}(x)}$$
$$S_{(a_1,...,a_n)}(x)=vl_{(a_1,...,a_n)}+vl_{-(a_1,...,a_n)}-\frac{vl_{(a_1,...,a_n)}*vl_{-(a_1,...,a_n)}}{F_{(0,...,0)}(x)}$$

由于 $\lim_{x\to 1^-}F_{(0,...,0)}(x)=\infty$，因此 $\lim_{x\to 1^-}S_{(a_1,...,a_n)}(x)=vl_{(a_1,...,a_n)}+$

$vl_{-(a_1,\ldots,a_n)}$

因此答案 ans 为：

$$ans = 1 + \sum_{(a_1,\ldots,a_n)\neq(0,\ldots,0)} \frac{1}{vl_{(a_1,\ldots,a_n)} + vl_{-(a_1,\ldots,a_n)}}$$

$$ans = 1 + \sum_{(a_1,\ldots,a_n)\neq(0,\ldots,0)} \frac{\prod_{i=1}^{n} l_i}{\sum_{(b_1,\ldots,b_n)\neq(0,\ldots,0)} \frac{(1-\prod_{i=1}^{n}\omega_{l_i}^{a_i b_i})+(1-\prod_{i=1}^{n}\omega_{l_i}^{-a_i b_i})}{1-\sum_{i=1}^{n} p_i \omega_{l_i}^{b_i}}}$$

这与算法 7 的结果相同。

4 总结

本题思维难度适中，代码难度适中，考察选手对部分经典模型的掌握程度。$v_i = 2$ 的情况可以通过方程组与 FWT 间的关联得到关键的性质。而 $v_i > 2$ 的情况可以看做 FWT 的拓展，即高维 DFT，因此可以直接将 $v_i = 2$ 的解法拓展到 v_i 任意的情况。

考虑随机游走过程的概率生成函数，通过期望步数等于 $F'(1)$ 的性质，可以用完全不同的方式得到相同的答案。笔者认为本问题可能存在更多的解决方式，但笔者能力有限，并没有得到更多的解法。

本题为笔者的原创题。感谢彭博参与验题工作。

5 参考资料

[1] 《浅谈 DFT 在信息学竞赛中的应用》，刘承奥，IOI2018 中国国家候选队论文集

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