move 解题报告

长沙市雅礼中学 袁宇韬

1 试题大意

给定一个 k 次多项式 $f(x)$ 。定义在长度为 n 的 01 序列上进行一次操作为：首先等概率随机选择一个位置作为当前位置 x，然后每次等概率随机选择一个位置 y，将第 y 位翻转（0 变为 1，1 变为 0），并将当前位置变为 y，同时得到 $f(|x-y|)$ 的收益，直到序列中所有数相同时结束。注意如果一开始序列中所有数相同，则会直接结束。

在一个无限长的 01 序列中有一些位置为 1，这些 1 的位置以区间给出，剩下位置全部为 0。每次询问会给定其中一个长度为 n 的区间，你需要将其中的数重新排列，使得你在这个区间上进行一次操作的期望收益最大，并求出重新排列后你的期望收益增加了多少。如果有多种方案使得期望收益最大，将其中 1 的位置从小到大排序，选择字典序最小的方案。即优先选择第一个 1 最靠左的，相同时选择第二个 1 最靠左的，等等。注意你并不会进行操作，只会将数重新排列。

答案在模 1004535809 意义下输出。

对于前 4 个测试点，n 分别为 5，6，8，9，$k=2000$，$m=2000$，$q=1500$ 。

对于接下来 3 个测试点，n 分别为 ≤ 50，≤ 500，≤ 5000，k 分别为 2000，100000，250000。

对于接下来 3 个测试点，n 分别为 ≤ 50000，≤ 500000，≤ 5000000，k 分别为 2000，100000，250000，$m=200$，$q=150$ 。

对于接下来 4 个测试点，$n \leq 50000000$，k 为 100000 和 250000 的测试点各占一半。

对于接下来 2 个测试点，$k=100000$ 。

对于接下来 4 个测试点，没有特殊性质。

对于所有测试点，$n \le 10^9$ ，$k \le 250000$ ，初始区间个数 ≤ 20000 ，询问个数 ≤ 15000 。保证输入所有区间的长度和不超过 50000000。

2 算法介绍

2.1 算法一

对于一个长度为 n 的 01 序列和一个开始位置，考虑怎样求出期望收益。可以将 01 序列和当前位置记为一个状态 $f_{i,j}$ ，表示当前序列为 i 的二进制表示，当前位置为 j 。容易列出转移方程。进行高斯消元可以解出答案。可以用实数计算比较大小，再在取模意义下计算输出答案。这样可以在 $O(n^3 8^n)$ 的时间内预处理出答案。

期望得分：10。

2.2 算法二

可以将每一步移动分为两部分，首先随机选择一个位置并得到这一步的收益，再移动当前位置并计算剩下的答案。由期望的可加性得到这两部分答案可以分别计算。得到

$$f_{i,j} = g_i + h_j (i \neq 0, i \neq 2^n - 1)$$
$$g_i = \frac{1}{n} \sum_{j=0}^{n-1} f_{i \operatorname{xor} 2^j, j}$$

其中 h_j 为从 j 位置开始随机选择一个位置移动的期望收益，可以在 $O(nk)$ 时间内计算。

这样可以将所有 f 用 g 代入，得到 2^n 个状态。可以在 $O(8^n)$ 的时间内预处理出答案。

期望得分：20。

2.3 算法三

由于方程中出现的常数项只有 h_j ，一定存在常数 $c_{i,j}$ 满足

$$g_i = \sum_{j=0}^{n-1} c_{i,j} h_j$$

其中 $c_{i,j}$ 与 h_j 无关。$c_{i,j}$ 表示从 i 状态开始时，h_j 被计算的期望次数。

令 $cnt(i)$ 为 i 状态中 1 的个数，$bit(i,j)$ 为 i 状态中第 j 位的值。由对称性，可以得到对于所有满足 $cnt(i) = cnt(i')$ ，$bit(i,j) = bit(i',j')$ 的 i, j, i', j' ，一定满足 $c_{i,j} = c_{i',j'}$ 。这样可以用 $a_{i,x}$ 表示满足 $bit(i,j) = x$ 的 j 对应的 $c_{i,j}$ 。

考虑所有 $i = 2^j - 1$ ，其中 $1 < j < n-1$ ，将 a 代入方程可以得到

$$\sum_{k=0}^{j-1} a_{j,1} h_k + \sum_{k=j}^{n-1} a_{j,0} h_k = \frac{1}{n}\Big(\sum_{k=0}^{j-1}\big(h_k + \sum_{p<j,p\neq k} a_{j-1,1} h_p + a_{j-1,0} h_k + \sum_{p=j}^{n-1} a_{j-1,0} h_p\big) + \sum_{k=j}^{n-1}\big(h_k + \sum_{p=0}^{j-1} a_{j+1,1} h_p + a_{j+1,1} h_k + \sum_{p\geq j,p\neq k} a_{j+1,0} h_p\big)\Big)$$

$$\sum_{k=0}^{j-1} a_{j,1} h_k + \sum_{k=j}^{n-1} a_{j,0} h_k = \frac{1}{n}\Big(\sum_{k=0}^{j-1}((j-1)a_{j-1,1} + a_{j-1,0} + (n-j)a_{j+1,1} + 1)h_k + \sum_{k=j}^{n-1}((n-j-1)a_{j+1,0} + a_{j+1,1} + j a_{j-1,0} + 1)h_k\Big)$$

这样可以得到

$$a_{j,1} = \frac{1}{n}((j-1)a_{j-1,1} + a_{j-1,0} + (n-j)a_{j+1,1} + 1)$$

$$a_{j,0} = \frac{1}{n}((n-j-1)a_{j+1,0} + a_{j+1,1} + j a_{j-1,0} + 1)$$

当 $j = 1$ 或 $j = n-1$ 时可以类似得到

$$a_{1,1} = \frac{1}{n}(n-1)a_{2,1}$$

$$a_{1,0} = \frac{1}{n}((n-2)a_{2,0} + a_{2,1} + 1)$$

$$a_{n-1,1} = \frac{1}{n}((n-2)a_{n-2,1} + a_{n-2,0} + 1)$$

$$a_{n-1,0} = \frac{1}{n}(n-1)a_{n-2,0}$$

注意到在重新排列一个区间的数时，$cnt(i)$ 一定保持不变，其中 i 为这个区间的状态。这样相当于选择一定数量的 0 变为 1，使得期望收益最大。

令 $d_j = a_{j,1} - a_{j,0}$ ，则在 $1 < j < n-1$ 时有

$$\begin{aligned} na_{j,1} - na_{j,0} = (j-1)a_{j-1,1} + a_{j-1,0} + (n-j)a_{j+1,1} + 1 - \\ (n-j-1)a_{j+1,0} - a_{j+1,1} - ja_{j-1,0} - 1 \end{aligned}$$

$$nd_j = (n-j-1)d_{j+1} + (j-1)d_{j-1}$$

类似得到

$$nd_1 = (n-2)d_2 - 1$$

$$nd_{n-1} = (n-2)d_{n-2} + 1$$

这样可以在 $O(n)$ 时间内解出所有 d_j 。同时通过观察可以发现当 $j < \frac{n}{2}$ 时 $d_j < 0$ ，当 $j = \frac{n}{2}$ 时 $d_j = 0$ ，当 $j > \frac{n}{2}$ 时 $d_j > 0$ 。由于将第 k 位从 0 变为 1 会增加 $d_j h_k$ 的期望收益，一定会选择 h_k 最大或最小的一些 k ，或选择任意 k 是等价的。

注意到 $h_k = \frac{1}{n}(\sum_{p=0}^{k-1} f(p) + \sum_{p=1}^{n-k} f(p))$ ，由于 $f(x)$ 一定递增，可以得到 h_k 当 $k < \frac{n}{2}$ 时递减，当 $k > \frac{n}{2}$ 时递增。这样可以求出应该怎样重新排列使得期望收益最大。同时，可以发现重新排列后只会有最多两个区间为 1。可以用 map 或平衡树维护所有为 1 的区间。

这样可以做到时间复杂度 $O(nk + (m+q)\log(m+q))$ 。

期望得分：30–40。

2.4 算法四

注意到统计答案时需要用到的 h_k 只有一些 $\sum_{k=l}^{r} h_k$ 形式的求和。这样可以求出 h_k 的前缀和，再对所有询问的区间端点求值。

由于 $h_k = \frac{1}{n}(\sum_{p=0}^{k-1} f(p) + \sum_{p=1}^{n-k} f(p))$ ，而两部分都是 $f(x)$ 的前缀和，则 h_k 同样为多项式。这样可以先求出 $f(x)$ 的前缀和，得到 h_k ，再求出 h_k 的前缀和。

求多项式的前缀和可以先求出 Bernoulli 数，再用一次卷积得到。用 Bernoulli 数的指数生成函数 $\frac{x}{e^x-1} = \sum_{m=0}^{\infty} B_m \frac{x^m}{m!}$ ，可以用一次求逆元操作得到 Bernoulli 数。关于多项式求逆的做法这里不再详细介绍。

这样可以将求 h_k 的过程优化到 $O(k\log k+\min(n,m+q)k)$ 。总时间复杂度为 $O(n+k\log k+\min(n,m+q)k+(m+q)\log(m+q))$ 。

期望得分：45–50。

2.5 算法五

对多个 x 求出 $f(x)$ 的值可以用多项式除法优化。由于 $f(x_0)=f(x) \mod (x-x_0)$ ，可以分治进行。如果需要对 $x=x_0,x_1,\ldots,x_{m-1}$ 求值，可以先求出 $f(x) \mod \prod_{i=0}^{\lfloor\frac{m}{2}\rfloor}(x-x_i)$ 和 $f(x) \mod \prod_{i=\lfloor\frac{m}{2}\rfloor+1}^{m-1}(x-x_i)$ ，再递归进行。

这样可以将求 h_k 的过程优化到 $O(k\log k+(m+q)\log^2(m+q))$ 。总时间复杂度为 $O(n+k\log k+(m+q)\log^2(m+q))$ 。

期望得分：50–70。

2.6 算法六

考虑关于所有 d_j 的方程。可以观察得到

$$d_j=\frac{\sum_{k=0}^{j-1}\binom{n-1}{k}-2^{n-2}}{\binom{n-2}{j-1}(n-1)2^{n-2}}$$

容易用数学归纳法证明正确性。注意到可以用这个式子在 $O(s)$ 时间内求出所有 d_j ，$1\le j\le s$ 。但是直接实现会超过空间限制。

将公式变形得到

$$d_j=\frac{\sum_{k=0}^{j-1}\frac{(n-1)!}{(n-k-1)!}\frac{(j-1)!}{k!}-2^{n-2}(j-1)!}{\frac{(n-1)!}{(n-j-1)!}2^{n-2}}$$

分子部分可以用 $O(1)$ 空间递推得到。分母部分需要预处理 $\frac{(n-1)!}{(n-j-1)!}$ 的逆元，可以只求出关于需要求出的 j 的项，这样只需要 $O(m+q)$ 的空间。

最后时间复杂度为 $O(s+k\log k+(m+q)\log^2(m+q))$ ，空间复杂度为 $O(k\log k+m+q)$ 。

期望得分：80–100。

3 试题考点与难点

考点：概率与期望问题的求解，多项式求逆，多项式除法，Bernoulli 数的性质。

难点：公式推导以及算法实现能力。

4 区分度设计

前 20 分为容易得到的部分分。其中根据算法效率分为两个 10 分段。

接下来 40 分需要 $O(n)$ 解出期望问题。这一段根据多项式处理算法的效率分为 15 分，15 分，10 分三段。

接下来 40 分需要 $O(s)$ 解出期望问题。这一段设计了有一定梯度的数据，用来区分算法实现常数不同的程序。

5 得分情况估计

有 0–1 名选手获得 60–100 分。

有 2–3 名选手获得 40–60 分。

有 3–5 名选手获得 30–40 分。

剩余选手大部分获得 10–20 分。