move 命题报告

长沙市雅礼中学 袁宇韬

题目大意

- 在一个无限长的 01 序列中有 s 个位置为 1。所有 1 的位置为 m 个区间，给出这些区间。

题目大意

- 在一个无限长的 01 序列中有 s 个位置为 1。所有 1 的位置为 m 个区间，给出这些区间。
- 给出 k 次多项式 $f(x)$。对于一个区间，定义移动序列为随机选择一个开始位置，再每次选择下一个位置，将这个位置取反，直到区间中所有数相同时的位置序列。定义这个序列的收益为每相邻两个位置的距离 x 代入 $f(x)$ 的值的和。定义这个区间的权值为这个区间的移动序列的期望收益。

题目大意

- 在一个无限长的 01 序列中有 s 个位置为 1。所有 1 的位置为 m 个区间，给出这些区间。
- 给出 k 次多项式 $f(x)$。对于一个区间，定义移动序列为随机选择一个开始位置，再每次选择下一个位置，将这个位置取反，直到区间中所有数相同时的位置序列。定义这个序列的收益为每相邻两个位置的距离 x 代入 $f(x)$ 的值的和。定义这个区间的权值为这个区间的移动序列的期望收益。
- 有 q 个询问，每次询问一个长度为 n 的区间中将所有数任意排列后最多能增加多少权值，并按照能够使权值最大的方案排列，有多种方案时选择 1 的位置依次尽量靠前的。答案模 1004535809 输出。

题目大意

- 在一个无限长的 01 序列中有 s 个位置为 1。所有 1 的位置为 m 个区间，给出这些区间。
- 给出 k 次多项式 $f(x)$。对于一个区间，定义移动序列为随机选择一个开始位置，再每次选择下一个位置，将这个位置取反，直到区间中所有数相同时的位置序列。定义这个序列的收益为每相邻两个位置的距离 x 代入 $f(x)$ 的值的和。定义这个区间的权值为这个区间的移动序列的期望收益。
- 有 q 个询问，每次询问一个长度为 n 的区间中将所有数任意排列后最多能增加多少权值，并按照能够使权值最大的方案排列，有多种方案时选择 1 的位置依次尽量靠前的。答案模 1004535809 输出。
- $n \le 10^9$，$k \le 250000$，$m \le 20000$，$q \le 15000$，$s \le 5 \times 10^7$。

题目大意

- 如果 $f(x) = x$，则区间 10101 的权值为 $\frac{2113}{80}$，而所有长度为 5 的有三个 1 的区间中权值最大的为 11001 或 10011，权值为 $\frac{4227}{160}$。

题目大意

- 如果 $f(x) = x$，则区间 10101 的权值为 $\frac{2113}{80}$，而所有长度为 5 的有三个 1 的区间中权值最大的为 11001 或 10011，权值为 $\frac{4227}{160}$。
- 如果询问区间中的序列为 10101，则应该将这个区间变为 11001，并输出 $\frac{1}{160}$。

状态压缩

- 显然这道题的关键是求出区间的权值。

状态压缩

- 显然这道题的关键是求出区间的权值。
- 使用状态压缩，用 $f_{i,t}$ 表示当前位置为 i，当前区间状态为 t 开始期望得到的收益。为了方便起见，t 用二进制表示。容易得到转移方程，用 $O(n^3 8^n)$ 时间解方程得到答案。

状态压缩

- 显然这道题的关键是求出区间的权值。
- 使用状态压缩，用 $f_{i,t}$ 表示当前位置为 i，当前区间状态为 t 开始期望得到的收益。为了方便起见，t 用二进制表示。容易得到转移方程，用 $O(n^3 8^n)$ 时间解方程得到答案。
- 期望得分：10 分。

状态压缩

- 用 g_t 表示当前状态为 t，第一步已经移动过的期望收益。用 h_i 表示从位置 i 开始移动一步的期望收益。

$$f_{i,t} = g_t + h_i \quad (t \neq 0, t \neq 2^n - 1)$$

$$g_t = \frac{1}{n} \sum_{i=0}^{n-1} f_{i, t \operatorname{xor} 2^i}$$

状态压缩

- 用 g_t 表示当前状态为 t，第一步已经移动过的期望收益。用 h_i 表示从位置 i 开始移动一步的期望收益。

$$f_{i,t} = g_t + h_i \quad (t \neq 0, t \neq 2^n - 1)$$

$$g_t = \frac{1}{n}\sum_{i=0}^{n-1} f_{i,t \operatorname{xor} 2^i}$$

- 将所有 $f_{i,t}$ 消去，只有 $O(2^n)$ 个变量，可以在 $O(8^n)$ 时间内解方程得到答案。

状态压缩

- 用 g_t 表示当前状态为 t，第一步已经移动过的期望收益。用 h_i 表示从位置 i 开始移动一步的期望收益。

$$f_{i,t} = g_t + h_i \quad (t \neq 0, t \neq 2^n - 1)$$

$$g_t = \frac{1}{n} \sum_{i=0}^{n-1} f_{i, t \operatorname{xor} 2^i}$$

- 将所有 $f_{i,t}$ 消去，只有 $O(2^n)$ 个变量，可以在 $O(8^n)$ 时间内解方程得到答案。
- 期望得分：20 分。

优化

- 令 $c_{i,t}$ 为从 g_t 的状态开始 h_i 被计算的期望次数。则 $g_t = \sum_{i=0}^{n-1} c_{i,t} h_i$。

优化

- 令 $c_{i,t}$ 为从 g_t 的状态开始 h_i 被计算的期望次数。则 $g_t=\sum_{i=0}^{n-1} c_{i,t}h_i$。
- 由于不同位置之间有对称性，$c_{i,t}$ 只与 t 中的 1 的个数与第 i 位的值有关。可以用 $a_{i,x}$ 表示在有 i 个 1 的状态 t 中值为 x 的位置 k 的 $c_{k,t}$。

优化

- 令 $c_{i,t}$ 为从 g_t 的状态开始 h_i 被计算的期望次数。则 $g_t = \sum_{i=0}^{n-1} c_{i,t} h_i$。
- 由于不同位置之间有对称性，$c_{i,t}$ 只与 t 中的 1 的个数与第 i 位的值有关。可以用 $a_{i,x}$ 表示在有 i 个 1 的状态 t 中值为 x 的位置 k 的 $c_{k,t}$。
- 考虑有 i 个 1，$n-i$ 个 0 的状态 t，将 g_t 的转移方程用 a 代入。

优化

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

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

$$(1 < i < n-1)$$

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

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

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

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

优化

- 由于 a_i 只与 a_{i-1} 和 a_{i+1} 有关，可以在 $O(n)$ 时间内解出方程。

优化

- 由于 a_i 只与 a_{i-1} 和 a_{i+1} 有关，可以在 $O(n)$ 时间内解出方程。
- 容易观察发现当 $i < \frac{n}{2}$ 时 $a_{i,1} < a_{i,0}$，当 $i = \frac{n}{2}$ 时 $a_{i,1} = a_{i,0}$，当 $i > \frac{n}{2}$ 时 $a_{i,1} > a_{i,0}$。由 $a_{i,1}$ 和 $a_{i,0}$ 的大小关系可以得到区间权值最大时 1 应该在 h_k 最大或最小的位置。

优化

- 由于 a_i 只与 a_{i-1} 和 a_{i+1} 有关，可以在 $O(n)$ 时间内解出方程。
- 容易观察发现当 $i < \frac{n}{2}$ 时 $a_{i,1} < a_{i,0}$，当 $i = \frac{n}{2}$ 时 $a_{i,1} = a_{i,0}$，当 $i > \frac{n}{2}$ 时 $a_{i,1} > a_{i,0}$。由 $a_{i,1}$ 和 $a_{i,0}$ 的大小关系可以得到区间权值最大时 1 应该在 h_k 最大或最小的位置。
- 由于 $h_k = \frac{1}{n}\left(\sum_{i=1}^{x} f(i) + \sum_{i=0}^{n-x-1} f(i)\right)$，而 $f(i)$ 单调递增，则 h_k 关于 $k = \frac{n}{2}$ 对称，且先递减再递增。

优化

- 由于 a_i 只与 a_{i-1} 和 a_{i+1} 有关，可以在 $O(n)$ 时间内解出方程。
- 容易观察发现当 $i < \frac{n}{2}$ 时 $a_{i,1} < a_{i,0}$，当 $i = \frac{n}{2}$ 时 $a_{i,1} = a_{i,0}$，当 $i > \frac{n}{2}$ 时 $a_{i,1} > a_{i,0}$。由 $a_{i,1}$ 和 $a_{i,0}$ 的大小关系可以得到区间权值最大时 1 应该在 h_k 最大或最小的位置。
- 由于 $h_k = \frac{1}{n}\left(\sum_{i=1}^{x} f(i) + \sum_{i=0}^{n-x-1} f(i)\right)$，而 $f(i)$ 单调递增，则 h_k 关于 $k = \frac{n}{2}$ 对称，且先递减再递增。
- 这样每次询问时 1 的最优位置一定是一个或两个区间。用平衡树维护 1 区间的端点，每次询问时对询问覆盖的所有 1 区间计算答案。总共的区间个数为 $O(m+q)$ 的。

优化

- 由于 a_i 只与 a_{i-1} 和 a_{i+1} 有关，可以在 $O(n)$ 时间内解出方程。
- 容易观察发现当 $i < \frac{n}{2}$ 时 $a_{i,1} < a_{i,0}$，当 $i = \frac{n}{2}$ 时 $a_{i,1} = a_{i,0}$，当 $i > \frac{n}{2}$ 时 $a_{i,1} > a_{i,0}$。由 $a_{i,1}$ 和 $a_{i,0}$ 的大小关系可以得到区间权值最大时 1 应该在 h_k 最大或最小的位置。
- 由于 $h_k = \frac{1}{n}\left(\sum_{i=1}^{x} f(i) + \sum_{i=0}^{n-x-1} f(i)\right)$，而 $f(i)$ 单调递增，则 h_k 关于 $k = \frac{n}{2}$ 对称，且先递减再递增。
- 这样每次询问时 1 的最优位置一定是一个或两个区间。用平衡树维护 1 区间的端点，每次询问时对询问覆盖的所有 1 区间计算答案。总共的区间个数为 $O(m+q)$ 的。
- 时间复杂度 $O(nk + (m+q)\log(m+q))$。

优化

- 由于 a_i 只与 a_{i-1} 和 a_{i+1} 有关，可以在 $O(n)$ 时间内解出方程。
- 容易观察发现当 $i < \frac{n}{2}$ 时 $a_{i,1} < a_{i,0}$，当 $i = \frac{n}{2}$ 时 $a_{i,1} = a_{i,0}$，当 $i > \frac{n}{2}$ 时 $a_{i,1} > a_{i,0}$。由 $a_{i,1}$ 和 $a_{i,0}$ 的大小关系可以得到区间权值最大时 1 应该在 h_k 最大或最小的位置。
- 由于 $h_k = \frac{1}{n}\left(\sum_{i=1}^{x} f(i) + \sum_{i=0}^{n-x-1} f(i)\right)$，而 $f(i)$ 单调递增，则 h_k 关于 $k = \frac{n}{2}$ 对称，且先递减再递增。
- 这样每次询问时 1 的最优位置一定是一个或两个区间。用平衡树维护 1 区间的端点，每次询问时对询问覆盖的所有 1 区间计算答案。总共的区间个数为 $O(m+q)$ 的。
- 时间复杂度 $O(nk + (m+q)\log(m+q))$。
- 期望得分：30–35 分。

使用 Bernoulli 数

- 可以用 Bernoulli 数优化 h_x 的求值。

使用 Bernoulli 数

- 可以用 Bernoulli 数优化 h_x 的求值。
- 求出 h_x 的前缀和关于 x 的多项式，只需要代入所有询问需要的区间端点。

使用 Bernoulli 数

- 可以用 Bernoulli 数优化 h_x 的求值。
- 求出 h_x 的前缀和关于 x 的多项式，只需要代入所有询问需要的区间端点。
- 由于不同区间端点个数是 $O(\min(n, m+q))$ 的，可以在 $O(k\min(n, m+q))$ 时间内求出所有询问需要的 h_x 的和。

使用 Bernoulli 数

- 可以用 Bernoulli 数优化 h_x 的求值。
- 求出 h_x 的前缀和关于 x 的多项式，只需要代入所有询问需要的区间端点。
- 由于不同区间端点个数是 $O(\min(n, m+q))$ 的，可以在 $O(k\min(n, m+q))$ 时间内求出所有询问需要的 h_x 的和。
- 时间复杂度 $O(n + k\min(n, m+q) + (m+q)\log(m+q))$。

使用 Bernoulli 数

- 可以用 Bernoulli 数优化 h_x 的求值。
- 求出 h_x 的前缀和关于 x 的多项式，只需要代入所有询问需要的区间端点。
- 由于不同区间端点个数是 $O(\min(n, m+q))$ 的，可以在 $O(k\min(n, m+q))$ 时间内求出所有询问需要的 h_x 的和。
- 时间复杂度 $O(n + k\min(n, m+q) + (m+q)\log(m+q))$。
- 期望得分：45–50 分。

使用多点求值

- 用多点求值优化代入 h_x 求值的过程。这样可以在 $O(k\log k + (m+q)\log^2(m+q))$ 的时间内求出所有需要的 h_x 的和。

使用多点求值

- 用多点求值优化代入 h_x 求值的过程。这样可以在 $O(k \log k + (m+q)\log^2(m+q))$ 的时间内求出所有需要的 h_x 的和。
- 时间复杂度 $O(n + k \log k + (m+q)\log^2(m+q))$。

使用多点求值

- 用多点求值优化代入 h_x 求值的过程。这样可以在 $O(k \log k + (m+q)\log^2(m+q))$ 的时间内求出所有需要的 h_x 的和。
- 时间复杂度 $O(n + k \log k + (m+q)\log^2(m+q))$。
- 期望得分：60–70 分。

进一步优化

- 由于询问时只需要用到 $a_{i,1}-a_{i,0}$，可以令 $d_i=a_{i,1}-a_{i,0}$。由 $a_{i,x}$ 的方程可以得到

$$nd_i=(n-i-1)d_{i+1}+(i-1)d_{i-1}\quad(1<i<n-1)$$

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

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

进一步优化

- 由于询问时只需要用到 $a_{i,1} - a_{i,0}$，可以令 $d_i = a_{i,1} - a_{i,0}$。由 $a_{i,x}$ 的方程可以得到

$$\begin{aligned} nd_i &= (n-i-1)d_{i+1} + (i-1)d_{i-1} \quad (1 < i < n-1) \\ nd_1 &= (n-2)d_2 - 1 \\ nd_{n-1} &= (n-2)d_{n-2} + 1 \end{aligned}$$

- 将 d_1 到 d_m 的方程相加，其中 $1 \le m < n-1$，得到

$$\sum_{i=1}^{m} nd_i = \sum_{i=2}^{m-1} nd_i + (n-m)d_m + (n-m-1)d_{m+1} + d_1 - 1$$

进一步优化

- 化简得到

$$(n-m-1)d_{m+1}-md_m=(n-1)d_1+1$$

进一步优化

- 化简得到

$$(n-m-1)d_{m+1} - md_m = (n-1)d_1 + 1$$

- 得到

$$\binom{n-2}{m}d_{m+1} - \binom{n-2}{m-1}d_m = \binom{n-1}{m}(d_1 + \frac{1}{n-1})$$

进一步优化

- 化简得到

$$(n-m-1)d_{m+1}-md_m=(n-1)d_1+1$$

- 得到

$$\binom{n-2}{m}d_{m+1}-\binom{n-2}{m-1}d_m=\binom{n-1}{m}(d_1+\frac{1}{n-1})$$

- 将这个式子对于 $1\le m\le k-1$ 求和，得到

$$\binom{n-2}{k-1}d_k=\sum_{i=0}^{k-1}\binom{n-1}{i}(d_1+\frac{1}{n-1})-\frac{1}{n-1}$$

进一步优化

- 取 $k = n - 1$，得到

$$d_{n-1} = (2^{n-1} - 1)(d_1 + \frac{1}{n-1}) - \frac{1}{n-1}$$

进一步优化

- 取 $k = n - 1$，得到

$$d_{n-1} = (2^{n-1} - 1)(d_1 + \frac{1}{n-1}) - \frac{1}{n-1}$$

- 由对称性，$d_{n-1} = -d_1$，可以得到

$$d_1 = \frac{1 - 2^{n-2}}{(n-1)2^{n-2}}$$

进一步优化

- 取 $k = n - 1$，得到

$$d_{n-1} = (2^{n-1} - 1)(d_1 + \frac{1}{n-1}) - \frac{1}{n-1}$$

- 由对称性，$d_{n-1} = -d_1$，可以得到

$$d_1 = \frac{1 - 2^{n-2}}{(n-1)2^{n-2}}$$

- 于是有

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

进一步优化

- 取 $k = n-1$，得到

$$d_{n-1} = (2^{n-1} - 1)(d_1 + \frac{1}{n-1}) - \frac{1}{n-1}$$

- 由对称性，$d_{n-1} = -d_1$，可以得到

$$d_1 = \frac{1 - 2^{n-2}}{(n-1)2^{n-2}}$$

- 于是有

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

- 这样可以在 $O(k)$ 时间内计算出 d_k。

进一步优化

- 由于 $k \leq s$，可以在 $O(s)$ 时间内预处理所有需要的 d_k。

进一步优化

- 由于 $k \le s$，可以在 $O(s)$ 时间内预处理所有需要的 d_k。
- 为了减少空间使用，可以只预处理需要使用的逆元，这样这一部分的空间复杂度为 $O(q)$。

进一步优化

- 由于 $k \leq s$，可以在 $O(s)$ 时间内预处理所有需要的 d_k。
- 为了减少空间使用，可以只预处理需要使用的逆元，这样这一部分的空间复杂度为 $O(q)$。
- 时间复杂度 $O(s + k \log k + (m + q) \log^2 (m + q))$。

进一步优化

- 由于 $k \le s$，可以在 $O(s)$ 时间内预处理所有需要的 d_k。
- 为了减少空间使用，可以只预处理需要使用的逆元，这样这一部分的空间复杂度为 $O(q)$。
- 时间复杂度 $O(s + k \log k + (m+q) \log^2 (m+q))$。
- 期望得分：90–100 分。

感谢

- 感谢 CCF 提供学习和交流的平台。
- 感谢汪星明老师对我的教导。
- 感谢帮助过我的同学们。
- 感谢大家的聆听。