题解

Happy · Lovely · Everyday!

首先，由于是 01 序列，所以 $\operatorname{popc}(x)=x$。而且异或运算具有结合律，所以合并的顺序是无关紧要的。这样，原问题就变为：

> 将该序列划分为若干个连续段，将每一段的异或和拼成一个新的序列，求本质不同的序列数量。

那么假设新序列为 $b_{1\sim k}$，a 的前缀异或和序列为 a'（定义 $a'_0=0$），那么 b 合法的条件即为：

> 存在一个序列 $p_{0\sim k}$，且 $p_0=0,p_k=n$，满足 $\forall i,a'_{p_{i-1}}\oplus a'_{p_i}=b_i$。

对 b 序列也做一个前缀异或和得到 b'，就可以发现 b 合法的条件就变成了 b' 是 a' 的子序列，且 $b'_n=a'_n$。因为一个 b' 唯一对应了一个 b，所以我们只需要统计 b' 的数量。

只需要求 $a'_{1\sim n-1}$ 的本质不同子序列个数即可，容易做到 $\mathcal{O}(n)$。

敬启，致那时的我

首先我们发现斐波那契数列没有什么好的性质，那么我们不妨直接维护其转移矩阵。那么我们设 F_k 为转移矩阵的 2^k 次幂，则特殊性质 C 相当于，在 $F_0,F_1,\dots,F_{n-1}$ 这 n 个矩阵中选择 k 个的乘积之和。

这个问题是容易的，注意到矩阵有分配律，所以我们只需要设 $f_{i,j}$ 表示前 i 个矩阵选了 j 个的乘积之和，直接转移即可，答案即为 $f_{n,k}$ 的 $(0,0)$ 位置，时间复杂度为 $\mathcal{O}(nk)$。

而原问题也可以类似的解决，只需要记录多一维表示是否超过前 i 位即可。可能要带一个 2^5 的常数，所以只开了 2×10^3。

upd：此题已被 deepseek R1 击杀。

Lead to shine more

第一档直接模拟即可。

第二档，观察样例可以发现答案为 $\mathcal{O}(\frac{n}{\log n})$ 级别，使用 $\mathcal{O}(1)$ 的 `__builtin_popcount()` 即可。实际上 $\mathcal{O}(\log n)$ 求 popc 可能也能通过。

第三四档，看到 10^{12}，考虑类似 $\mathcal{O}(\sqrt{n})$ 的时间复杂度。将高的 $\frac{k}{2}$ 位和低的 $\frac{k}{2}$ 位分开考虑，那么低的位进到高的位的次数是 $\mathcal{O}(\sqrt{n})$ 的，而高位对低位的影响只是 x 的初值，和每次操作额外加上的一个值。所以我们可以对每一对 a,b 暴力预处理出：x 初始为 a，每次操作额外加上 b，第一次向前进位时所需的次数，以及进位以后余下的值。a,b 都是 $\mathcal{O}(\log n)$ 级别，而每次预处理的时间复杂度是 $\mathcal{O}(\frac{\sqrt{n}}{b})$，所以预处理的时间复杂度为 $\mathcal{O}(\sqrt{n}\log n)$，查询的时间复杂度为 $\mathcal{O}(\sqrt{n})$。分块打表可以通过第三档。

考虑继续拓展第三档的思想。我们将 x 从高位到低位一位位变为 1。类似的，预处理出 $f_{i,a,b},g_{i,a,b}$ 表示 x 初始为 a，每次操作额外加上 b，第一次向第 i 位进位时所需的次数与进位以后余下的值即可。这样就无法暴力预处理了，但是考虑给第 i 位进位相当于给第 $i-1$ 位进两次位，所以可以分解成两个 $i-1$ 的子问题。那么我们得到转移方程：

$$lst=f_{i-1,a,b}$$

$$f_{i,a,b}=f_{i-1,lst,b+1}$$

$$g_{i,a,b}=g_{i-1,a,b}+g_{i-1,lst,b+1}$$

由于每次 x 最多增加 $\lfloor \log n \rfloor$，所以我们不处理 $i < \log \log n$ 的 $f_{i,*}, g_{i,*}$，当考虑到小于 8 的 i 时直接暴力。

预处理时间复杂度为 $\mathcal{O}(\log^3 n)$，查询时间复杂度为 $\mathcal{O}(\log n)$。拓展不到位的第三四档（比如拆成三个 $\frac{k}{3}$ 位或者更多）可以通过第五档。

永恒的盛宴

显然每只虫子最多只能吃到一只虫子，那么我们考虑分别计算每一只虫子能够吃到虫子的概率。

我们递归的刻画出虫子 u 能够吃到虫子的条件：

- u 的行动晚于 a_u。
- 在所有其他满足 $a_v = a_u$ 且行动晚于 a_u 的 v 中，a_u 的行动是最早的。
- v 不能够吃到虫子。

然后其实只要第一二个条件有一个不满足我们就不需要关心第三个条件了。

考虑枚举一条链 $b_0 \to b_1 \to \cdots \to b_k$（$b_0 = u$），且 $b_0, b_1, \ldots, b_{k-1}$ 都满足前两个条件，但是 b_k 不满足。这样的话，如果 k 为奇数，那么 u 就能吃到虫子。我们对于这样的概率求和即可。进一步的，我们不强制令 b_k 不满足，转而用 $b_0, b_1, \ldots, b_{k-1}$ 都满足的方案减去 $b_0, b_1, \ldots, b_k$ 都满足的方案。那么其实总答案就是奇数长度的链的概率减去偶数长度的链的概率。

由于我们有第一个条件，所以我们只需要考虑简单链。如果一条链形成了一个完整的 ρ 形，可能需要对交点特殊处理，但是这并不重要。

考虑计算这样的概率。问题可以抽象成类似这样：

> 有 $m+1$ 种颜色的小球，颜色 0 的小球有 $m+1$ 个，颜色 i（$i > 0$）的小球有 c_i 个，求出任意排列小球使得对于每一个 i（$1 \le i \le m$），第 $i-1$ 个颜色 0 的小球与第 i 个颜色 0 的小球之间都没有第 i 种颜色的小球的概率。

这是容易计算的，一个做法可以直接强制钦定第 $i-1$ 个颜色 0 的小球与第 i 个颜色 0 的小球之间放了 d_i 个第 i 种颜色的小球，并乘上容斥系数 $(-1)^{\sum d_i}$，然后剩下的小球没有限制，可以直接插板，我们把 $\sum d_i$ 记到 DP 状态里即可。

还有一种做法是从单个颜色 0 的小球开始，每一次插入一个颜色 0 的小球与 c_i 个颜色 i 的小球。转移时枚举这 c_i 个颜色为 i 的小球有多少插在左边有多少插在右边。注意到我们只关心最后一个颜色 0 的小球的位置，在 DP 状态中记录这个东西即可。转移需要前缀和优化才能与前一个做法有一样的复杂度。

我们枚举每一个链头，然后在链上边走边 DP，把 DP 的结果加到答案里面。DP 单次转移是 $\mathcal{O}(nc_i)$ 的，而对于一个链头，我们经过的点的 c_i 之和是 $\mathcal{O}(n)$ 的，那么可以分析得到总时间复杂度是 $\mathcal{O}(n^3)$。