

Xor Master 解题报告

深圳中学 李可

2023.8

1 试题

1.1 简要题意

有一个长为 n 的序列 a ，和一个初始为空的集合 S_0 。

定义 $xor(S)$ 表示集合 S 中所有元素的异或和，若 S 为空则是 0。

定义 $g(x, S) = \max_{T \subseteq S} (x \oplus xor(T))$ 。

定义 $f(l, r) = g(\oplus_{i=l}^r a_i, S_0)$ ， $\oplus_{i=l}^r a_i$ 表示 $a_l \sim a_r$ 的异或和。

进行 Q 次操作或询问：

1 x v: 将 a_x 异或上 v 。

2 x: 将 x 加入集合 S_0 中。

3 l r: 询问 $\sum_{i=l}^r f(l, i)$ 的值 $\text{mod } 2^{64}$ 。

1.2 输入格式

第一行两个整数 n, Q 。

第二行 n 个非负整数 $a_1 \sim a_n$ 。

接下来 Q 行，每行描述一次操作或询问，格式见题目描述。

1.3 输出格式

对于每次询问输出一行一个整数，表示答案 $\text{mod } 2^{64}$ 的值。

1.4 数据范围

对于 100% 的数据, $1 \leq n \leq 5 \times 10^5$, $1 \leq Q \leq 10^5$, $0 \leq a_i, x, v < 2^m$, m 的限制见下表。

子任务编号	分值	$n \leq$	$Q \leq$	$m =$	特殊性质
1	10	2000	2000	32	无
2	10	5×10^5	10^5	64	A, B
3	15	5×10^5	10^5	64	B
4	15	10^5	10^5	32	B, C
5	20	10^5	10^5	32	无
6	30	5×10^5	10^5	64	无

特殊性质 A: 不存在 1 类操作。

特殊性质 B: 不存在 2 类操作。

特殊性质 C: 对于所有询问, 保证 $l = 1$ 。

2 算法分析

2.1 算法 1

可以通过维护 S_0 的线性基来对一个 x 在 $O(\log V)$ 的时间内求出 $g(x, S)$ 。

暴力回答询问, 时间复杂度 $O(nQ \log V)$ 。

可以通过 subtask1。

2.2 算法 2

对于 subtask2, 不存在任何修改操作。记 s_i 表示 a_i 的前缀异或和, 此时询问相当于直接询问区间 $[l, l], [l, l+1], \dots, [l, r]$ 每个的异或和之和, 也即 $\sum_{i=l}^r s_{l-1} \oplus s_i$ 。

预处理出 $c[i][j]$ 表示 $s_1 \sim s_i$ 中有多少个第 j 位是 1, 查询时枚举二进制位, 考虑有多少个 $s_{l-1} \oplus s_i$ 在这一位是 1, 即可 $O(\log V)$ 回答询问。

时间复杂度 $O((n+Q) \log n)$, 可以通过 subtask2。

2.3 算法 3

对于 subtask3, 不存在操作 2, 因此仍然不需要考虑 S_0 带来的影响。

操作 1 的单点异或在前缀异或和数组 s 中的体现是区间异或一段后缀, 可以使用线段树来维护算法 2 中的 c 数组。

时间复杂度 $O(n \log V + Q \log n \log V)$, 可以通过 subtask3。

2.4 算法 4

对于 subtask4, 由于保证询问的 $l = 1$, 所以询问相当于求 $\sum_{i=1}^r g(s_i, S_0)$, 如果能提前预处理出所有 $g(s_i, S_0)$ 即可 $O(1)$ 回答询问。

操作 2 将一个元素加入线性基, 由线性基的性质可知有效的操作至多 $\log V$ 次, 又因为没有操作 1, 可以在每次线性基改变后重新对每个 s_i 暴力计算 $g(s_i, S_0)$ 的值。

时间复杂度 $O(n \log^2 V + Q)$, 可以通过 subtask4。

2.5 算法 5

在算法 4 的基础上, 使用类似高斯消元的方法将线性基维护成一个行最简形矩阵, 这样当线性基发生变化时, 可以 $O(1)$ 由 $g(s_i, s_0)$ 算出新的 $g(s_i, s'_0)$ 。

时间复杂度 $O(n \log V + Q)$, 可以通过 subtask5。

2.6 算法 6

下记 $g(x, S) = \max_{T \subseteq S} (x \oplus \text{xor}(S))$, $h(x, S) = \min_{T \subseteq S} (x \oplus \text{xor}(S))$ 。

引理 1 $g(x \oplus y, S) = g(x, S) \oplus h(y, S)$

考虑利用线性基计算 $g(x, S), h(y, S)$ 的过程, 从高位往低位扫, 若线性基第 i 位上有数, 则 $g(x, S)$ 的第 i 位肯定会被修改成 1, $h(y, S)$ 的第 i 位被修改成 0, 因此 $g(x, S) \oplus h(y, S)$ 的第 i 位一定是 1, 实际上与计算 $g(x \oplus y, S)$ 的过程一致。

由上述引理可知, $\sum_{i=l}^r f(l, i) = \sum_{i=l}^r h(s_{l-1}, S_0) \oplus g(s_i, S_0)$ 。

用线段树维护 $g(s_i, S_0)$ (每一位上有多少个 1), 1 类操作相当于给一段后缀异或上 $h(v, S_0)$, 2 类操作使用算法 5 的方法或直接暴力重构, 查询到 $h(s_{l-1}, S_0)$ 后线段树可以做到 $O(\log n \log V)$ 回答询问。

时间复杂度 $O(n \log^2 V + Q \log n \log V)$ ，可以通过 subtask6。若实现精细可能可以获得满分。

2.7 算法 7

算法 6 的瓶颈在于每次线性基变化时需要 $O(n \log V)$ 的时间重建线段树。

考虑使用类似 UOJ 671 的优化方法：将线段树一个代表长为 len 区间的节点中的 $\log V$ 个数列成一张表格，每行有 $\log len$ 个数代表每个数的二进制表示。将这个“表格”竖向存储，即用 $\log len$ 个 64 位整数分别存储每一列，pushup 时可以用位运算对 64 位并行计算二进制加法。区间异或与查询也可以较为方便的实现。此时建树的时间复杂度由 $T(n) = 2T(\frac{n}{2}) + \log n$ 可推出是 $O(n)$ 的。

2 类操作使用算法 5 的方法来修改 $g(s_i, S_0)$ ，此时总复杂度变为 $O(n \log V + Q \log n \log V)$ ，可以通过此题。

3 总结

本题思维难度和代码难度适中，考察了线性基和线段树的综合应用，部分分充足且提示性强，即使未能想到关键技巧或结论仍能获得较多分数。

4 参考资料

<https://oiwiki.org/math/linear-algebra/basis/>

<https://uoj.ac/problem/671>