

# 《Xor》 解题报告

南京外国语学校 程思元

October 25, 2022

## Contents

|          |               |          |
|----------|---------------|----------|
| <b>1</b> | <b>题目大意</b>   | <b>2</b> |
| <b>2</b> | <b>数据范围</b>   | <b>2</b> |
| <b>3</b> | <b>解题过程</b>   | <b>2</b> |
| 3.1      | 算法一 . . . . . | 2        |
| 3.2      | 算法二 . . . . . | 3        |
| 3.3      | 算法三 . . . . . | 3        |
| 3.4      | 算法四 . . . . . | 3        |
| 3.5      | 算法五 . . . . . | 4        |
| 3.6      | 算法六 . . . . . | 4        |
| 3.7      | 算法七 . . . . . | 6        |
| 3.8      | 算法八 . . . . . | 8        |
| <b>4</b> | <b>命题思路</b>   | <b>9</b> |
| <b>5</b> | <b>参考资料</b>   | <b>9</b> |

## 1 题目大意

给定一个长度为  $n$  的序列  $a_1, a_2, \dots, a_n$ 。你需要进行共  $q$  次下面两种操作：

- 1 l r：将  $a_{[l,r]}$  替换为它的异或差分。形式化地说，令  $b_i = a_i \text{ xor } a_{i-1} (l < i \leq r)$ ，然后对于每个  $l < i \leq r$ ，将  $a_i$  替换为  $b_i$ 。
- 2 pos：查询  $a_{pos}$  的值。

操作执行完后，你还需要回答最终的  $a$  序列。

## 2 数据范围

对于所有数据，保证  $1 \leq n \leq 2.5 \times 10^5, 1 \leq q \leq 10^5, 0 \leq a_i < 2^{30}, 1 \leq l \leq r \leq n, 1 \leq pos \leq n$ 。

| 子任务编号 | $n \leq$          | $q \leq$        | 特殊性质 | 分值 | 子任务依赖               |
|-------|-------------------|-----------------|------|----|---------------------|
| 1     | $2 \times 10^3$   | $2 \times 10^3$ | 无    | 8  | 无                   |
| 2     | $2.5 \times 10^5$ | $10^5$          | A    | 4  | 无                   |
| 3     | $2.5 \times 10^5$ | $10^5$          | B    | 7  | 无                   |
| 4     | $2.5 \times 10^5$ | $10^5$          | CD   | 13 | 2                   |
| 5     | $2.5 \times 10^5$ | $10^5$          | DE   | 12 | 无                   |
| 6     | $2.5 \times 10^5$ | $10^5$          | D    | 16 | 5                   |
| 7     | $2.5 \times 10^5$ | $10^5$          | E    | 11 | 5                   |
| 8     | $2.5 \times 10^5$ | $10^5$          | 无    | 29 | 1, 2, 3, 4, 5, 6, 7 |

特殊性质 A： $\forall i \geq 2, a_i = 0$ 。

特殊性质 B： $0 \leq a_i \leq 1$ 。

特殊性质 C：记序列  $a$  中非零位置个数为  $c$ ，则  $c \leq 100$ 。

特殊性质 D：操作 1 满足  $l = 1, r = n$ 。

特殊性质 E：没有操作 2。

## 3 解题过程

在本文中，我们用  $(x, y)$  表示网格从下往上数第  $x$  行，从左往右数第  $y$  列的格子。

### 3.1 算法一

直接模拟每次操作 1 即可。

时间复杂度为  $\Theta(nq)$ ，可以通过子任务 1，期望得分 8。

### 3.2 算法二

考虑特殊性质 B。我们使用 bitset 维护  $a$  序列。每次相当于把  $a$  序列的一个子段左移一位再和  $a$  序列进行异或。这里主要问题是取出这个子段，这里有两种解决方法。一种是直接手写 bitset 然后只对相关的子段进行操作。另一种是设一个阈值  $B$ ，然后对每个  $k$  预处理出只有前  $k$  个位置是 1 的 bitset，这里的  $B$  是空间和时间的平衡。

时间复杂度为  $\Theta(\frac{nq}{\omega})$ ，可以通过子任务 3，结合算法一期望得分 15。

### 3.3 算法三

考虑特殊性质 A。由于  $a$  序列中只有 0 和  $a_1$ ，所以无论如何操作都只会产生 0 和  $a_1$ 。我们继续使用 bitset 维护  $a$  序列，bitset 中的一个位置为 1 当且仅当  $a$  序列中这个位置为  $a_1$ 。于是直接套用算法 2 即可。

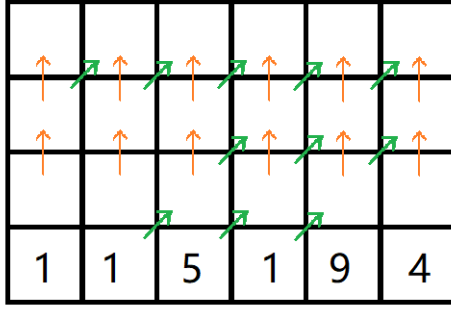
时间复杂度为  $\Theta(\frac{nq}{\omega})$ ，可以通过子任务 2，结合算法一、二期望得分 19。

### 3.4 算法四

上面的做法都比较无脑，接下来我们要对问题进行一些分析。抛开看上去是数据结构的外壳，考虑操作实际上干了什么。令  $q_1$  为操作 1 的数量，我们构造一个  $(q_1 + 1) \times n$  的网格，行从下往上编号为  $0, 1, \dots, q_1$ ，列从左往右编号为  $1, 2, \dots, n$ 。我们在第 0 行的第  $i$  列填入  $a_i$ ，并用第  $i$  行表示进行了  $i$  次操作后的  $a$  序列。例如下图为样例 1。

|   |   |   |   |   |   |
|---|---|---|---|---|---|
|   |   |   |   |   |   |
|   |   |   |   |   |   |
|   |   |   |   |   |   |
| 1 | 1 | 5 | 1 | 9 | 4 |

然后我们考虑操作对  $a$  序列的影响。首先，对于一个位置  $i$ ，无论它是否在操作区间内，新的序列  $a'_i$  的值都要异或上  $a_i$ ，所以我们画一条有向边  $(t, i) \rightarrow (t + 1, i)$ ，我们称之为基础边。其次，对于一个位置  $i$ ，若  $l \leq i < r$ ，则新的序列中  $a'_{i+1}$  的值也要异或上  $a_i$ ，所以对于在区间中的  $i$ ，我们连一条有向边  $(t, i) \rightarrow (t + 1, i + 1)$ ，我们称之为操作边。例如下图为样例 1，我们用橙色的边表示基础边，用绿色的边表示操作边。



我们定义一条路径的权值是该路径起点的权值，那么我们每次查询的就是所有以第 0 行格子为起点，以查询的格子为终点的所有路径的权值异或和。我们定义一个不在第 0 行的格子的权值为查询这个格子的答案。

有了对题意的转化，接下来我们考虑如何解决这个问题。

首先考虑性质 D，性质 D 是一个非常好的性质，它在这张网格图上的体现是每一行的所有位置都存在操作边。

对于有这个性质的网格图，我们可以简单计算固定起点和终点后，这个起点对这个终点是否会产生贡献（也就是路径个数的奇偶性）。设起点为  $(x_s, y_s)$ ，终点为  $(x_t, y_t)$ ，则路径的个数为  $\binom{y_t - y_s}{x_t - x_s}$ 。

当同时满足性质 C 时，第 0 行只有最多 100 个起点。对每次查询，我们可以暴力枚举起点计算贡献。

时间复杂度为  $\Theta((n + q)c)$ ，可以通过子任务 1, 2, 4，结合算法二期望得分 32。

### 3.5 算法五

我们继续讨论满足性质 D 的问题。当同时满足性质 E 时，我们只要求出最上面一行的格子里的数即可。考虑化简我们确定某对起点终点是否产生贡献的式子  $\binom{y_t - y_s}{x_t - x_s}$ ，根据 lucas 定理，这个式子为奇数当且仅当在二进制表示下  $(x_t - x_s)$  中 1 的位置集合是  $(y_t - y_s)$  的子集。在满足性质 DE 的问题中， $(x_t - x_s)$  是定值，为  $q_1$ ，故  $a_i$  对  $ans_j$  有贡献当且仅当在二进制表示下  $(j - i)$  中 1 的位置集合是  $q_1$  的子集，也就是我们可以给  $i$  加上若干个  $2^k$ （其中  $q_1$  的第  $k$  位是 1）。于是我们可以使用类似数位 dp 的方法，记  $f_{i,j}$  表示转移了前  $i$  位后位置  $j$  的答案。第一维可以用滚动数组优化掉。

时间复杂度为  $\Theta(n \log q + q)$ ，可以通过子任务 5，结合算法二、四期望得分 44。

### 3.6 算法六

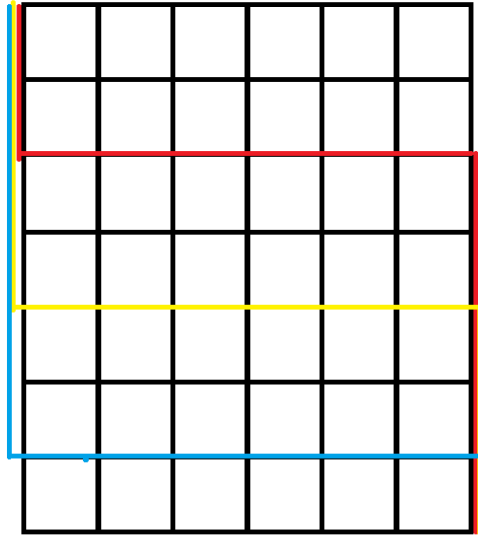
现在只满足性质 D 了，考虑如何处理在网格中间的查询。我们发现对于那个式子  $\binom{y_t - y_s}{x_t - x_s}$ ，它是奇数的一个必要条件是  $x_t - x_s \leq y_t - y_s$ ，这启发我们在  $y_t - y_s$  较小的时候暴力枚举起点。在此之前，我们先证明一个引理。

**引理 1.** 在这张网格图上，我们从左上角的点出发，画一条只能向右或向下的轮廓线，轮廓线只能在网格图的顶点处更改方向，最终到达右下角的点。对于两个格子  $s, t$ ，若  $s$  在轮廓线的左下方， $t$  在轮廓线的右上方，则所有  $s$  到  $t$  的路径跨过了这条轮廓线恰好一次。

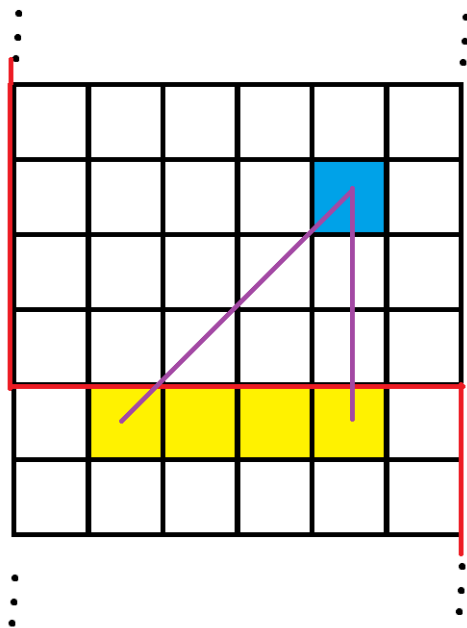
*Proof.* 对于每个  $0 \leq i \leq q_1$ , 我们记  $L_i$  表示第  $i$  行有  $L_i$  个格子在轮廓线的左边, 则  $L_i$  单调不升。由于经过网格图中的两类边都会使  $y$  增加 1, 所以从  $S$  到  $T$  的路径必然在第  $i$  行对应一个唯一的  $x$  坐标  $P_i$ , 而  $P_i$  单调不降。又因为  $P_{y_s} = x_s \leq L_{y_s}, P_{y_t} = x_t \geq L_{y_t}$ , 所以必然存在恰好一个  $i$  使得  $P_i \leq L_i, P_{i+1} > L_{i+1}$ 。□

在算法四中, 我们定义了一条路径的权值是该路径起点的权值, 一个不在第 0 行的格子的权值是所有以第 0 行的点为起点, 以这个格子为终点的路径的权值的异或和。这里我们再定义一条边的权值是其起点的权值。所以若第 0 行的所有格子在轮廓线的左下方, 某个格子  $t$  在轮廓线的右上方, 那么所有以这条轮廓线上的**边**为起点以  $t$  为终点的路径的权值异或和也等于  $t$  的权值。

我们考虑设阈值  $B$ , 从  $y = 0$  和  $y = 1$  的分界线开始每隔  $B$  个格子画一条轮廓线, 如下图 ( $B = 2$ ):



对于每次查询的格子  $t$ , 我们找到其下方离它最近的轮廓线, 显然这条轮廓线和它的距离不超过  $B$ 。我们枚举正好在这条轮廓线下方的一排格子作为起点。由于  $\binom{y_t - y_s}{x_t - x_s}$  是奇数的一个必要条件是  $x_t - x_s \leq y_t - y_s$ , 只需要枚举最多  $B$  个起点。如下图, 蓝色的格子是我们查询的格子, 那么只有黄色的格子是有用的。



于是我们只需要对每条轮廓线，求出正好在这条线下方的一排格子的权值。这可以使用算法五解决。

时间复杂度为  $\Theta(qB + \frac{nq}{B} \log q)$ ， $B = \Theta(\sqrt{n \log q})$  时有最优时间复杂度  $\Theta(q\sqrt{n \log q})$ 。可以通过子任务 4, 5, 6，结合算法二、四期望得分 60。

### 3.7 算法七

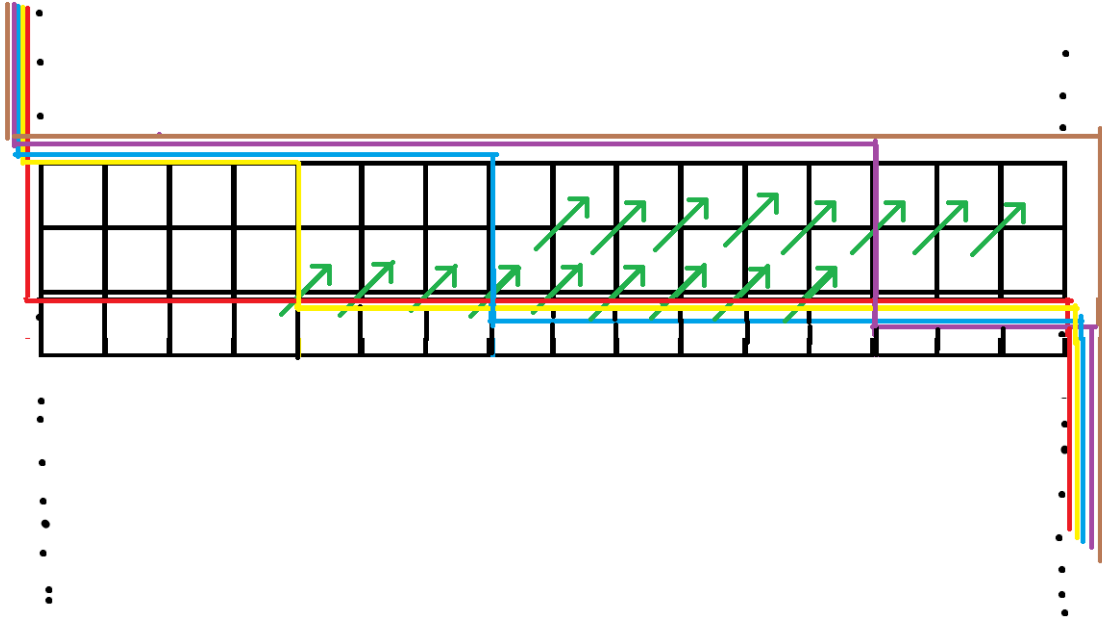
考虑不满足性质 D 的一般情况。此时操作边不一定存在了，就不容易直接计算两个格子之间的贡献。

考虑如果  $n$  远大于  $q$  会发生什么。此时可以看作是每个操作的两个端点共  $2q$  个关键点把长度为  $n$  的序列划分成了  $O(q)$  个段，每个段内每一列的操作边存在性相同。

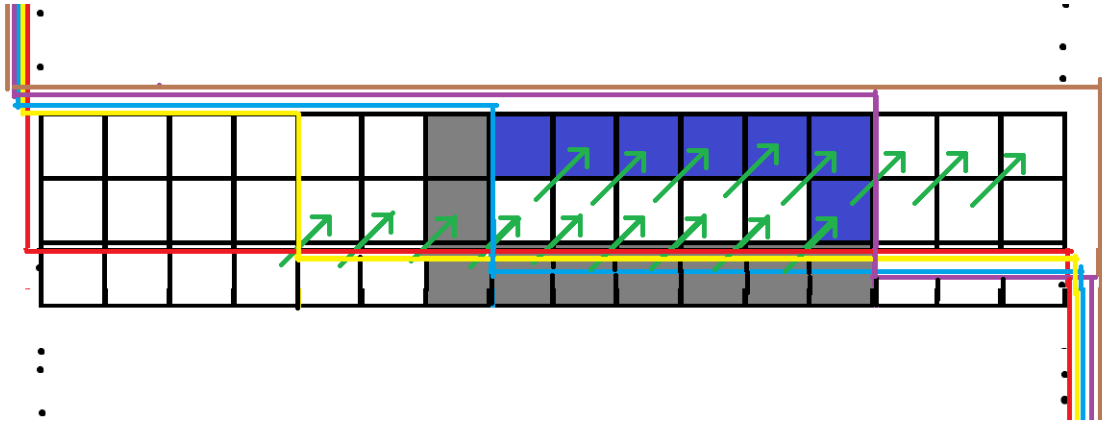
我们继续使用算法六的思想，设阈值  $B$  并画轮廓线。如果我们把相邻两条轮廓线之间的部分看作一个子问题，这个子问题对应的子网格就被竖着划分成了  $O(B)$  个段。

先考虑性质 E 也就是没有查询操作的情况。那么我们只需要根据下面这条轮廓线下方格子的权值求出上面这条轮廓线下方格子的权值。

我们考虑在这个子网格上画  $O(B)$  条轮廓线，每条轮廓线从子网格的左上角出发向右走到一个关键点，然后向下走到网格的第 0 行和第 1 行之间，然后向右走到网格的右边界。如下图所示（省略了基础边）：



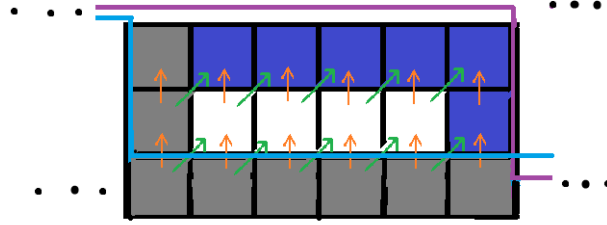
我们从左往右考虑每一条轮廓线，每次需要根据正好在前一条轮廓线左下方的格子的权值求出正好在后一条轮廓线左下方的格子的权值。



例如，当我们从蓝色的轮廓线转移到紫色的轮廓线的时候，我们就是知道灰色格子的权值，要求出深蓝色格子的权值。

我们首先暴力求出蓝色的轮廓线右边一列的格子，这一步的复杂度是  $\Theta(B)$  的。然后我们就只需要考虑这两条轮廓线之间的子网格。

此时每一行要么全都有操作边，要么全都没有。如果第  $i$  行的格子都没有操作边，则第  $i$  行每个格子的权值和第  $i+1$  行同一列格子的权值一致，那么我们可以删掉第  $i$  行，在计算完成后第  $i+1$  行最右边格子的权值就是第  $i$  行最右边格子的权值。这样就转化为了每个格子都既有基础边又有操作边的情况（也就是性质 D，不过性质 D 只要考虑上下边界，这里还要考虑左右边界）。



先对最左边的列做差分，撤销掉灰色格子之间的基础边造成的影响，然后我们就可以对每个蓝色格子算灰色格子对它的贡献的异或和。

这里我们将灰色格子分为在左边界上的和在下边界上的，把蓝色格子分为在右边界上的和在上边界上的（四个角上的格子可以任选一类放入）。于是我们只需对四种类型的贡献分开计算。设最下面一行为第 0 行，最上面一行为第  $h$  行，最左边一列为第 0 列，最右边一列为第  $w$  列，下面我们列出四种情况的系数（均对 2 取模）：

- 下边界  $(0, s) \rightarrow$  右边界  $(t, w)$ :  $\binom{t}{w-s}$
- 下边界  $(0, s) \rightarrow$  上边界  $(h, t)$ :  $\binom{h}{t-s}$
- 左边界  $(s, 0) \rightarrow$  上边界  $(h, t)$ :  $\binom{h-s}{t}$
- 左边界  $(s, 0) \rightarrow$  右边界  $(t, w)$ :  $\binom{t-s}{w}$

考虑对四种情况分别计算。第一种相当于每个数贡献到自己的子集然后 reverse；第三种相当于每个数可以贡献到子集的超集然后 reverse；第二种和第四种就是算法五。均可以使用数位 dp 解决。于是处理左右相邻两条轮廓线的复杂度为  $\Theta((h+w) \log \min(h, w))$ 。

$\min(h, w) \leq h \leq B$ ,  $\sum_w = n$ , 于是处理上下相邻两条轮廓线的复杂度为  $\Theta((\sum_w + \sum_h) \log B) = \Theta((n + B^2) \log B)$ 。

那么总复杂度就是  $\Theta(\frac{q}{B}(n + B^2) \log B) = \Theta((\frac{nq}{B} + qB) \log B)$ 。

取  $B = \Theta(\sqrt{n})$ , 时间复杂度为  $\Theta(q\sqrt{n} \log n)$ , 可以通过子任务 5,7, 结合算法二、四、六期望得分 71。

### 3.8 算法八

算法七已经解决了大部分问题，接下来只需支持查询某个格子的权值即可。这有两种处理方法，一种是直接使用算法六种处理查询某个格子的权值的方法暴力在下方轮廓线的区间上计算贡献，另一种方法是把查询格子所在列也设为关键列直接查出这一整列的权值。两种方法均不增加时间复杂度，其中后一种方法的适用性更广。后一种方法能查一整列，且如果把操作改成区间前缀和而不是区间差分也可以做。std 实现的是后一种方法。

时间复杂度为  $\Theta(q\sqrt{n} \log n)$ , 这个  $\log n$  实际上只有一半，且数位 dp 常数较小（和前缀和差不多），可以通过所有子任务，期望得分 100。



## 4 命题思路

本题是一道原创题。

本题中在四个边界之间互相贡献的形式，和 [SD/SXOI2022] 子串统计一题中的贡献形式是一致的。笔者当时在写这题时，起初之间考虑格子的贡献，写起来十分混乱，后来想到使用轮廓线的转移，维护轮廓线上的边的权值，思路就十分清晰。本题也使用了这种维护轮廓线和上面的边的转移方式。

笔者在命制此题时，一开始是从算法六的单点查询方法考虑，想通过三次根号重构，以  $n^{\frac{2}{3}}$  为块大小进行序列分块来解决本题，后来发现仍然不能避免数位 dp。笔者认为传统的数据结构方法在本题上没有优势。本题虽然采用的是数据结构题的形式，但做法与数据结构题关系不大，而是使用了一些组合方法。

本题的子任务层层递进，有较高的区分度。前三个子任务只需要比较暴力地维护即可通过，从第四个子任务开始需要选手将问题转化为网格图格路计数等模型。从第四个子任务开始，每个子任务的做法都是下一个子任务做法的基础，这些子任务对选手的解题有启发作用。

本题难度适中，代码量适中，涉及知识面广，对选手的水平有较为综合的考察。

## 5 参考资料

1. CCF NOI 科学委员会，《全国青少年信息学奥林匹克系列竞赛大纲》：<https://www.noi.cn/generic/web/viewer.html?file=/upload/resources/file/2021/04/06/152179.pdf>。

李昕然同学、魏佳泽同学参与了本题的验题工作。