

# CodeChef JAN 2015 Xor Queries 解题报告

李昊

2015 年 11 月 1 日

## 1 试题大意

给定一个初始时为空的整数序列以及一些询问：

类型0：在数组最后加入数字 $x$ 。

类型1：在区间 $L..R$ 中找到数字 $y$ ，最大化 $(x \text{ xor } y)$ 。

类型2：删除数组最后 $k$ 个元素。

类型3：在区间 $L..R$ 中，统计小于等于 $x$ 的元素个数。

类型4：在区间 $L..R$ 中，找到第 $k$ 小的数。

询问次数 $\leq 500000$

## 2 解题思路

### 2.1 $L=1, R=N$ 的情况

先来考虑弱化的版本： $L=1, R=N$ 的情况。

建一棵trie，按照二进制从高位到低位建树，可以在 $O(\text{数字二进制位数})$ 的时间内支持插入，同样的也支持删除。在每个节点记录有多少数字在其子树中，即可支持三种查询。

#### 2.1.1 类型1

从高位到低位依次操作，判断当前节点代表与该位数字相反的子树是否为空，若非空则进入。最后所在点所表示的数即为答案。

### 2.1.2 类型3

$x$ 加1，从高位到低位操作，若当前位数字为1，答案加上当前节点代表0的子树的大小，然后进入当前节点代表1的子树；否则直接进入当前节点代表0的子树。

### 2.1.3 类型4

从高位到低位操作，若当前节点代表0的子树大小大于等于 $k$ ，则进入；否则减去当前节点代表0的子树大小，然后进入当前节点代表1的子树，最后所在点所表示的数即为答案。

## 2.2 $L=1$ 的情况

由于 $R$ 不一定等于 $N$ ，所以需要维护 $R=1 \dots N$ 所有情况时候的trie。观察发现插入操作和删除操作最多修改数字位数个节点，所以可以使用可持久化trie。每次要修改一个节点时，不在原来的节点上修改，而是复制一份，在新的节点上修改，每次最多新增数字位数个节点。

对于查询，在相应的代表 $R$ 的trie上执行上述操作即可。

## 2.3 $L、R$ 无限制

第 $L$ 到 $R$ 个数字形成的trie等价与第 $R$ 个可持久化trie-第 $L-1$ 个可持久化trie。所以每次操作时只需在第 $R$ 个可持久化trie和第 $L-1$ 个可持久化trie上执行上述操作即可。

## 2.4 时空复杂度

时空复杂度皆为 $O(m * \text{数字二进制位数})$