最大异或和解题报告

浙江省镇海中学 杜瑜皓

1 试题来源

2015年集训队互测by 金策

2 试题大意

维护一个长度为n的数列$a_1, a_2, \cdots, a_n$，每个a_i为m位二进制数，有q个操作，类型有三种：

- 将一段数全部异或上一个数w
- 将一段数全部改成w
- 询问$a_1, a_2, \cdots, a_n$最大子集的异或和。

3 算法介绍

首先做个转化令$a'_1 = a_1, a'_i = a_i \oplus a_{i-1} (i > 1)$。那么原数列和现在的数列的最大子集异或和是一样的，因为a'_i在a_i生成的线性空间中，同理a_i在a'_i生成的线性空间中。所以它们生成了同一个线性空间。

那么第一个操作就相当于修改其中两个数，第二个操作就相当于将某些数变成0，并同时修改两个数。变成0这一部分可以均摊处理，所以就相当于$O(n+q)$次修改一个数的操作，同时要询问最大子集异或和。

询问一个数列最大子集异或和的一个方法是使用高斯消元，维护这个数列呈上三角形式的一组基，然后逐位判断。但是这个做法只能添加一个数，修改一个数并不方便。

考虑一个数据结构的经典做法离线，对时间建立线段树，维护每个数字出现的时间区间，然后对应的线段树上一些区间。

查询的时候，就遍历这个线段树，每次把线段树上这个节点中所有数字加入这组基，回溯的时候只要撤销掉就行了，撤销操作很简单，只要把这个基改成0就行了。

这样时间复杂度是$O(n^3 \log n)$的，事实上能拿到80 ~ 100分。

接下来考虑加速高斯消元的过程，考虑高斯消元的流程，加入一个数，从上到下每次看一下能否和这个基对应的最高位消掉。这样要做$O(n)$次，然后对这组基的影响是$O(1)$的。

我们可以分块，将w个基组成一块，每次预处理出来这w位对应基的情况，也就是有2^w中情况，那么消元能做到$O(n/w)$，因为连续w个能直接做。然后加入这组基，要将对应块中的预处理结果全部更新，就是$O(2^w)$的复杂度。取$w = O(\log n)$，那么加入一个基能做到$O(n^2/\log n)$。

于是这样做的时间复杂度还是$O(n^3)$的，并且这个过程还是可以使用bitset优化。

4 总结

本篇题解给出了一个与作者解法完全不同的做法，虽然更加复杂，常数更加大，但我觉得这还是一个比较有趣的做法。

虽然这是我想的一个做法，但是可惜$O(\frac{n^3}{\log n})$的在$\mathbb{GF}(2)$上的高斯消元早就有人发过paper了，事实上这个可以当成Method of the Four Russians 的一个简单推广。这个方法还能使用bitset优化，所以效率比渐进更好的算法快得多，有些库的实现就是这个方法的。