

F题

题解

根据题意，要求得 $a_i \text{ xor } x_j$ 的最大值，也就是选取一个 a_i 使得任意一个二进制位

尽量与 x_j 不同，可以考虑将 a_i 根据二进制位存储为一颗字典树，在查询时便可以很方便可以得到答案。

不得不感叹字典树的代码是如此的经典，能在寥寥几行完成需要完成的工作

建树

```
1 void insert(int x)
2 {
3     int cur = 0, cur_bit; // cur为当前的节点编号, cur_bit为当前的二进制
4     for (int i = 30; i >= 0; --i)
5     {
6         cur_bit = (x >> i) & 1; // 自高位到低位取二进制位
7         if(dict_tree[cur][cur_bit] == 0) // 当前节点是叶子节点, 则向下延伸
8             dict_tree[cur][cur_bit] = cur_node++;
9         cur = dict_tree[cur][cur_bit]; // 获取下一个节点编号
10    }
11    value[cur] = x; // 在相应的cur中存入x的值
12 }
```

查询

```
1 int query(int x)
2 {
3     int cur = 0, cur_bit;
4     for (int i = 30; i >= 0; --i)
5     {
6         cur_bit = !((x >> i) & 1); // 因为是异或, 尽量满足x与a的二进制位不同
7         if(dict_tree[cur][cur_bit] == 0)
8             cur_bit = !cur_bit; // 如果无法满足不同则变为相同
9         cur = dict_tree[cur][cur_bit];
10    }
11    return value[cur];
12 }
```

以上