

# Assignment 5 Report

## 数组中两个数异或求最大值

曹旭 大数据学院 16307110230

### 一、题目描述与分析

有一个非空整数类型数组， $a[0], a[1], a[2], \dots, a[n-1]$ ，其中  $0 \leq a[i] < 2^{31}$ ，通过设计算法找到  $a[i] \text{ XOR } a[j]$  的最大结果，其中  $0 \leq i, j < n$ ，并在时间复杂度  $O(n)$  实现这一操作。对于这一问题，假如使用暴力法（两两比较），时间复杂度为  $O(n^2)$ ，在  $n$  较大时花费较多时间。考虑到徐老师最近刚讲完树，因此尝试构建 Tire 树（字典树）存储输入的数组，再每个数进行比较来解决这一问题，由于已经给了限定条件  $0 \leq a[i] < 2^{31}$ ，将  $n$  个整数放入 Tire 树的时间复杂度  $T \leq 32n$ ，全部数进行比较的时间复杂度  $T = 32n$ 。因此总时间复杂度为  $O(n)$ ，符合题目的要求。但是，假如没有限定条件  $0 \leq a[i] < 2^{31}$ ，此时时间复杂度为  $O(n \log(n))$ ，这是因为二叉树的深度是用  $\log(n)$  来衡量的。

### 二、使用 Tire 树的解决方法

首先设计 transfer 函数，该函数可以将 array 数组中的十进制数转换为对应的二进制数，该函数输出一个存储二进制的数组，用于下一步二叉字典树的构建。

```
int *transfer(int num, int nodenum) {
    int i, tempnodenum;
    int *a = (int *)malloc(sizeof(int) * nodenum);
    tempnodenum = findnodenum(num);
    for (i = nodenum - 1; i >= 0; i--) {
        if (num != 0) {
            a[i] = num % 2;
            num /= 2;
        }
        else
            a[i] = 0;
    }
    return a;
}
```

构建二叉字典树，以 0—7 为例，如下图所示表所示：

十进制	0	1	2	3	4	5	6	7
二进制	000	001	010	011	100	101	110	111

表 1. 0—7 十进制和二进制转换

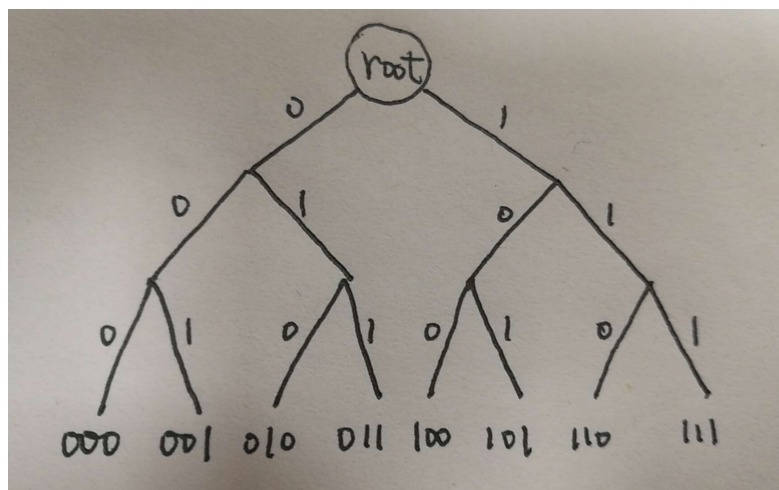


图 1. 0—7 二进制二叉字典树的构建图

在构建树的时候，需要同时考虑空间复杂度，因此采用链表存储的二叉树存储结构，而非顺序存储。然后，将 array 数组内的所有数都放入这样的一棵二叉树 T 中。完成这一

步骤后，开始逐个计算每个数在数组中可能的最大异或值。以题目给的例程为例：

Sample Input

6

3 10 5 25 2 8

在二叉树中，3 为:root->0->0->0->1->1, 10 为:root->0->1->0->1->0，以此类推。令  $T1=T$ ,  $T2=T$ ,  $x=0$ ，从 3 (00011) 开始遍历二叉树  $T1$ ,  $root \rightarrow 0$ ，此时另一棵树  $T2$  寻找  $root$  是否存在  $\rightarrow 1$  的情况，假如有，则走到  $root \rightarrow 1$ ，令  $x = x + 2^4$ ，假如没有，则走到  $root \rightarrow 0$ ， $x$  不变，继续走到下一层并判断。

利用这一策略，可以得到数组中每个数的最大异或值的值，再从中选出最大的值为输出。这一方法的时间复杂度为  $O(n)$ 。

题目存在一些与题意不符的问题：

助教老师给的 Framework 中可以将 `int` 改成 `long long`，要不然较大的数  $2^{31}$  没法输入，会溢出。