

位运算+数据结构+前缀树

421. 数组中两个数的最大异或值

题目描述：

给你一个整数数组 `nums`，返回 `nums[i] XOR nums[j]` 的最大运算结果，其中 $0 \leq i \leq j < n$ 。

进阶：你可以在 $O(n)$ 的时间解决这个问题吗？

题目分析： $a \oplus b = x \rightarrow a = x \oplus b; b = a \oplus x$;

```
class Solution {
private:
    // 最高位的二进制位编号为 30
    static constexpr int HIGH_BIT = 30;

public:
    int findMaximumXOR(vector<int>& nums) {
        int x = 0;
        for (int k = HIGH_BIT; k >= 0; --k) {
            unordered_set<int> seen;
            // 将所有的  $pre \oplus k(a_j)$  放入哈希表中
            for (int num: nums) {
                // 如果只想保留从最高位开始到第 k 个二进制位为止的部分
                // 只需将其右移 k 位
                seen.insert(num >> k);
            }

            // 目前 x 包含从最高位开始到第 k+1 个二进制位为止的部分
            // 我们将 x 的第 k 个二进制位置为 1，即为  $x = x * 2 + 1$ 
            int x_next = x * 2 + 1;
            bool found = false;

            // 枚举 i
            for (int num: nums) {
                if (seen.count(x_next ^ (num >> k))) {
                    found = true;
                }
            }
            if (found) x = x_next;
        }
        return x;
    }
};
```

```

        break;
    }
}

if (found) {
    x = x_next;
}
else {
    // 如果没有找到满足等式的 a_i 和 a_j, 那么 x 的第 k 个二进制位只
    能为 0

    // 即为 x = x*2
    x = x_next - 1;
}
}
return x;
}
};

```

作者: LeetCode-Solution

链接: <https://leetcode-cn.com/problems/maximum-xor-of-two-numbers-in-an-array/solution/shu-zu-zhong-liang-ge-shu-de-zui-da-yi-h-n9m9/>

来源: 力扣 (LeetCode)

著作权归作者所有。商业转载请联系作者获得授权, 非商业转载请注明出处。

```

struct Trie {
    // 左子树指向表示 0 的子节点
    Trie* left = nullptr;
    // 右子树指向表示 1 的子节点
    Trie* right = nullptr;

    Trie() {}
};

class Solution {
private:
    // 字典树的根节点
    Trie* root = new Trie();
    // 最高位的二进制位编号为 30
    static constexpr int HIGH_BIT = 30;

public:
    void add(int num) {
        Trie* cur = root;
        for (int k = HIGH_BIT; k >= 0; --k) {
            int bit = (num >> k) & 1;
            if (bit == 0) {
                if (!cur->left) {
                    cur->left = new Trie();
                }
            }
        }
    }
};

```

```

        cur = cur->left;
    }
    else {
        if (!cur->right) {
            cur->right = new Trie();
        }
        cur = cur->right;
    }
}

int check(int num) {
    Trie* cur = root;
    int x = 0;
    for (int k = HIGH_BIT; k >= 0; --k) {
        int bit = (num >> k) & 1;
        if (bit == 0) {
            // a_i 的第 k 个二进制位为 0, 应当往表示 1 的子节点 right 走
            if (cur->right) {
                cur = cur->right;
                x = x * 2 + 1;
            }
            else {
                cur = cur->left;
                x = x * 2;
            }
        }
        else {
            // a_i 的第 k 个二进制位为 1, 应当往表示 0 的子节点 left 走
            if (cur->left) {
                cur = cur->left;
                x = x * 2 + 1;
            }
            else {
                cur = cur->right;
                x = x * 2;
            }
        }
    }
    return x;
}

int findMaximumXOR(vector<int>& nums) {
    int n = nums.size();
    int x = 0;
    for (int i = 1; i < n; ++i) {
        // 将 nums[i-1] 放入字典树, 此时 nums[0 .. i-1] 都在字典树中
        add(nums[i - 1]);
        // 将 nums[i] 看作 ai, 找出最大的 x 更新答案
    }
}

```

```
        x = max(x, check(nums[i]));  
    }  
    return x;  
}  
};
```

作者: LeetCode-Solution

链接: <https://leetcode-cn.com/problems/maximum-xor-of-two-numbers-in-an-array/solution/shu-zu-zhong-liang-ge-shu-de-zui-da-yi-h-n9m9/>

来源: 力扣 (LeetCode)

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。