# CS100 算法入门

## 位运算（选学）

# 位运算：布尔运算的推广

→ 0和1就是true和false，并且有一些基本逻辑运算

| A,B | A=0,B=0 | A=0,B=1 | A=1,B=0 | A=1,B=1 |
|---|---|---|---|---|
| A && B（and） | 0 | 0 | 0 | 1 |
| A ll B（or） | 0 | 1 | 1 | 1 |
| A ^ B（xor） | 0 | 1 | 1 | 0 |

→ 把整数写成二进制形式，对每一位做如上的运算
→ 得到一个新的二进制数，再转回十进制
→ 这种操作就称为(按)位运算

| A,B | A=5,B=6<br>101 110 | A=7,B=1<br>111 001 |
|---|---|---|
| A & B（and） | 5&6=4<br>101&110=100 | 7&1=1<br>111&001=001 |
| A l B（or） | 5l6=7<br>101l110=111 | 7l1=7<br>111l001=111 |
| A ^ B（xor） | 5^6=3<br>101^110=011 | 7^1=6<br>111^001=110 |

George Boole (1815-1864)

# 位运算规律

**交换律：**

A&B=B&A (A和B都是1)

A|B=B|A (A和B中有一个是1)

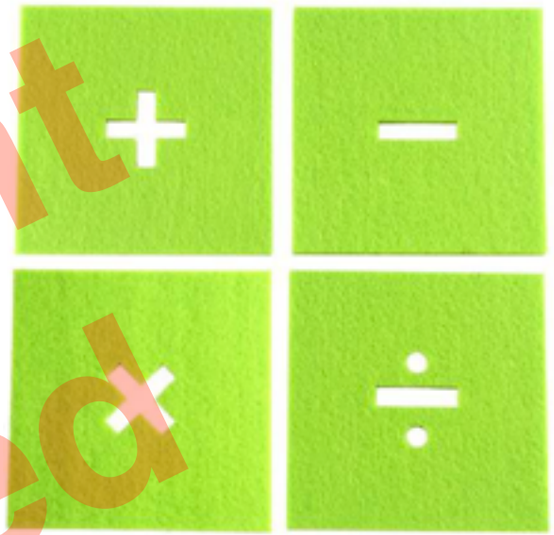
A^B=B^A (A和B不相等)

**结合律：**

(A&B)&C=A&(B&C) (ABC都是1)

(A|B)|C=A|(B|C) (ABC中有一个是1)

(A^B)^C=A^(B^C) (ABC中有**一个或三**个是1)



**自反律：**

A&A=A

A|A=A

A^A=0

| A,B | A=0,B=0 | A=0,B=1 | A=1,B=0 | A=1,B=1 |
|---|---|---|---|---|
| A && B（and） | 0 | 0 | 0 | 1 |
| A ll B（or） | 0 | 1 | 1 | 1 |
| A ^ B（xor） | 0 | 1 | 1 | 0 |

**怎么证明?**

**只要对某一位成立，就对所有整数成立**

# 快速判重技术

**唯一数问题：**

输入n个整数（n<=100000000），其中只有一个数只出现一次，其他数都出现两次。输出只出现一次的那个数

**样例输入：**

11

5 3 2 4 2 7 4 6 5 3 7

**样例输出：**

6



# 唯一数问题：思路

```
输入n
result=0
for (i=1;i<=n-1;i++) {
   输入a
   result=result^a
}
输出result
```

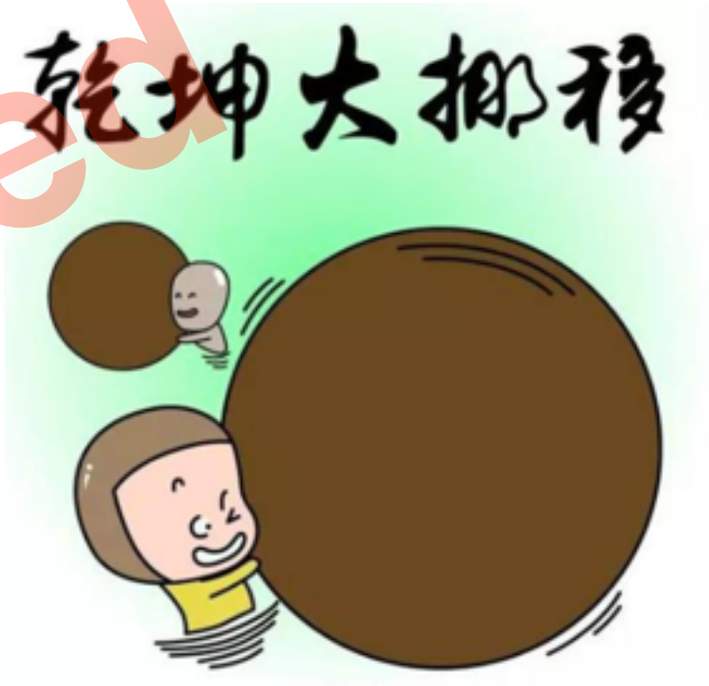
→ **根据n的规模，肯定需要线性算法，每个数回头找一下是不行的**

→ **如果是排过序的，那就很简单了**

→ **如何不排序也能达到排序后的效果?**

**没看懂？这就是利用了位运算的运算规律**

5^3^2^4^2^7^4^6^5^3^7
=(2^2)^(3^3)^(4^4)^(5^5)^6^(7^7) **(交换律+结合律)**
=0^0^0^0^6^0 **（自反律）**
=6



# 唯一数问题：解答

```
12  int findOneNumberInTwos(int n){
13      int result=0,a;
14      for (int i=0;i<n;i++){
15          cin>>a; // 输入第i个数
16          result ^= a;     // 累积异或
17      }
18      return result;
19  }
20
21  int findOneNumberInThrees(int n){...}
33
34  int main(int argc, const char * argv[]) {
35      int n;
36      cin>>n;
37      /*11
38       5 3 2 4 2 7 4 6 5 3 7*/
39      cout<<findOneNumberInTwos(n)<<endl;
40      /*10
41      5 4 4 7 7 4 6 5 5 7*/
42  //      cout<<findOneNumberInThrees(n)<<endl;
43      return 0;
44  }
45
```

```
11
5 3 2 4 2 7 4 6 5 3 7
6
```

→ findOneNumberInThrees这是红包题，我不会轻易给你们看到的😋

# 作业

**4.计算机的存储是以字节(Byte，简写B)为单位的，1个B表示8位(bit)二进制数。所以位运算其实应该叫“按位运算”**

一定数量的字节数有特殊的称谓。例如：1KB=$2^{10}$B，大约能存一篇500字纯文本作文。写出你所知道的其他类似称谓，并举一个例子

（参照前面下划线的这句话）

**5.（选做）唯一数问题2（onefromthree.cpp）**

**输入n个整数（n<=10000000），其中只有一个数只出现一次，其他数都出现三次。输出只出现一次的那个数**

（提示：

1.这题挺难的，你们肯定做不出😏

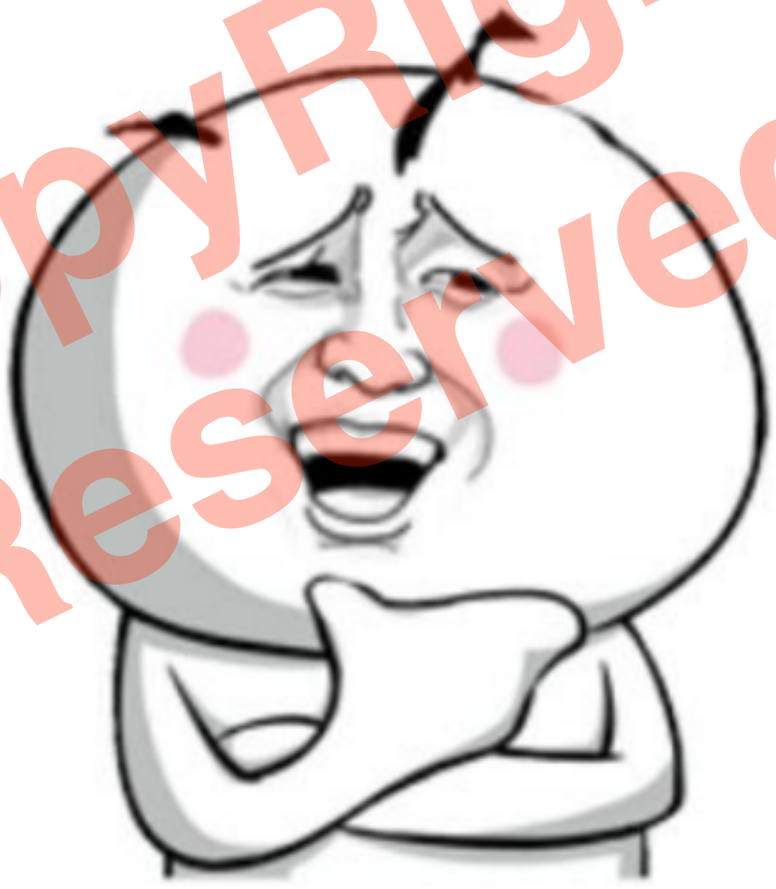
2.我把答案发给你们。如果做不出的可以看答案

3.看懂答案的，交算法的解释，**解释对的也有红包**

4.如果你想不清楚，先考虑**二进制的一位**

## 端午节附加作业（下周停课）

# 并没有~骗你们的



# 扩展阅读：其他位运算符

| A | A=0 | A=1 | A=101 | A=111 | | |
|---|---|---|---|---|---|---|
| **A<<1** | 00000000 | 00000010 | 00001010 | 00001110 | 左移(2的幂次) | $1<<x=2^x$ |
| **A>>1** | 00000000 | 00000000 | 0.00000010 | 00000011 | 右移(除2的幂次) | $A>>x=A/2^x$ |
| **~A** | 11111111 | 11111110 | 11111010 | 11110001 | 取反 | |

# 扩展阅读：压缩存储

| | 上午班(32) | 下午班(16) | 男生(8) | 女生(4) | 长得好(2) | 读小学(1) | 特征值 | 上午班的帅哥(42) | 下午班的小学生(17) |
|---|---|---|---|---|---|---|---|---|---|
| 王宇骥 | 0 | 1 | 1 | 0 | 1 | 1 | 27 | 27 & 42= 10 | 27 & 17= 17 |
| 瞿意 | 1 | 0 | 1 | 0 | 1 | 0 | 42 | 42 & 42=42 | 42 & 17=0 |
| 吴瑞麟 | 1 | 0 | 1 | 0 | 1 | 1 | 43 | 43 & 42=42 | 43 & 17=1 |
| 李宜澍 | 1 | 0 | 0 | 1 | 1 | 0 | 38 | 38 & 42=2 | 38 & 17=0 |
| 袁欣怡 | 0 | 1 | 0 | 1 | 1 | 0 | 22 | 22 & 42=2 | 22 & 17=16 |
| 章小洲 | 0 | 1 | 1 | 0 | 1 | 1 | 27 | 27 & 42=10 | 27 & 17=17 |
| 吴瑞麟和瞿意的共同特征 | 42 & 38=34 | 上午班<br>长得好 | | | | | | | |
| 吴瑞麟和章小洲的共同特征 | 43 & 27=11 | 帅哥<br>读小学 | | | | | | | |


→ 当处理多个布尔值的时候，使用位运算是非常方便的。例如判断一个对象的多种正负属性

→ 一次位运算与一次布尔运算的时间是一样的（但位运算一次就能做多个逻辑判断），因此位运算既省时又省空间

→ 位运算之所以快，因为计算机底层都是用二进制计算的，所以本质上所有运算都是位运算