

数据结构与算法

Data Structure and Algorithm

XXVIII 位操作

授课人: Kevin Feng

翻 译: 孙 兴

课前回顾

- 💡 数学回顾
- 📦 数组
- ⭐ 递归 vs 迭代
- 💡 动态数组 vs 链表
- 📦 堆栈和队列
- ☁️ 哈希表
- 💡 树
- 📱 堆
- 💡 排序
- 📱 图论
- 📱 分治法
- ☁️ 动态规划



位操作 | Bit Operation

- 按位操作是在单个字节的层面上对一个或多个位模式或二进制数字符号进行的操作。处理器支持这种快速简单的操作，而且可以用来比较和计算数值。
- 按位操作比除法快得多，是乘法的几倍速度，有时候也比加法快得多。

~	complement	Bit n of ~x is the opposite of bit n of x			
&	And	Bit n of x&y is 1 if bit n of x and bit n of y is 1.			
	Or	Bit n of x y is 1 if bit n of x or bit n of y is 1.			
^	Exclusive Or	Bit n of x^y is 1 if bit n of x or bit n of y is 1 but not if both are 1.			
>>	Right Shift (divide by 2)	Bit n of x>>s is bit n-s of x.			
<<	Left Shift (multiply by 2)	Bit n of x<<s is bit n+s of x.			
And (&):	0 & 0 = 0	1 & 0 = 0	0 & 1 = 0	1 & 1 = 1	
Or ():	0 0 = 0	1 0 = 1	0 1 = 1	1 1 = 1	
Xor (^):	0 ^ 0 = 0	1 ^ 0 = 1	0 ^ 1 = 1	1 ^ 1 = 0	

基本技巧

⊙ XOR

- $\text{myBits} \wedge 0$: 不变
- $\text{myBits} \wedge 1$: 翻转

⊙ 左移位相当于乘以2的指数

- $x \ll n = x * (1 \ll n)$
- $8 \ll 2 = 8 * (1 \ll 2) = 32$

⊙ 按位右移相当于一个除以2的指数

- $00011001 \gg 2 = 00000110$
- $00011001 \gg 4 = 00000001$

位操作

- ⦿ Set Bit: 此处n是位数, 0是最低有效位。
 - `setBit(a, n)`
- ⦿ Clear Bit
- ⦿ Toggle Bit
- ⦿ Test Bit
 - `testBit(a, n)`
- ⦿ 将整数转换为Bits(二进制)
 - `toBinary(n)`
- ⦿ 将Bits(二进制)转换为整数
 - `convertBits2Integer(binary)`

位操作 II

- ⊙ 用Bits展示小数。
 - 给定一个数（十进制小数，如3.72），用字符串将这个数传入，打印它的二进制表示。如果这个数不能表示为二进制，打印‘ERROR’。
- ⊙ 将HEX转换为整数
 - `hex2Integer(s)`
- ⊙ 将整数转换为HEX
 - `integer2Hex(d)`
- ⊙ 计算一个整数（二进制表示）包含的1的个数
- ⊙ $(n \& (n - 1) == 0)$ 可以检测的是？
- ⊙ 下一个2的幂数
 - 给定一个整数n，找到下一个数（这个数比n大，且是2的幂数）
- ⊙ 检测两个整数的是否有相反符号
 - 给定两个整数，检查他们是否有相同的符号

位操作 III

- ⦿ 乘以7
 - 给定一个整数 x ，不使用乘法 $*$ ，计算 $7*x$ 。
- ⦿ 计算一个整数的符号
- ⦿ 不使用分支计算一个整数的绝对值
- ⦿ 就地（in-place）整数交换
- ⦿ 将整数A转换为整数B
 - 计算将整数A转换为整数B所需要的位数
- ⦿ Amazing Mask
 - 给定两个32-bit的数，N和M，以及两个比特位置 i 和 j 。写一个方法，这个方法可以将N的*i*到*j*之间的值设置为M（即，M变成N的*i*到*j*之间的一个子串）

位操作 IV

- ⦿ 不使用算术运算符进行添加
 - 写一个两数相加的函数。不可以使用+或者任何算术运算符。
- ⦿ 找到丢失的数字（一个或多个）
- ⦿ 找到下一个最小/最大整数（二进制表示，与原数值有相同多个1）

数学

⦿ Reservoir Sampling

- 在一个无穷的数据流中，挑选一个概率相等的数字。

⦿ 阶乘结果的尾部0

- 写一个算法，计算 n 阶乘结果中尾部0的个数。

⦿ 最大公约数

⦿ 数学运算

- 写一个可以实现 $*$ ， $-$ ， $/$ 运算的方法，但是你只能用 $+$ 运算符

⦿ 2的个数

- 写一个计算0到 n 之间的2的个数



数据结构与算法

Data Structure and Algorithm

XXVIII 位操作

授课人: Kevin Feng