

## 1. 计算机存储单位

bit, Byte, KB, MB, GB, TB, PB ...

“比特” 保存 0 或 1    Byte: 字节    “字节跳动的 ByteDance”    1 Byte = 8 bit

1 KB = 1024 Byte ( $2^{10}$  Byte) (Kilo)

1 MB = 1024 KB ( $2^{20}$  Byte) (Million)

1 GB = 1024 MB ( $2^{30}$  Byte) (Giga...)

## 2. “int” 整型

4 Byte = 32 bit    32个0和1    [000...0] ~ [111...1]     $2^{32}$  个数字

负数怎么办?    最高位 0 表示正    1 表示负

+127 : 0111 1111    1111 1111 : -127

## 3. 原码, 反码, 补码

二进制表示, 除高位外    int: 32位    0111...11    +2147483647     $2^{31}-1$

000...000  $\Rightarrow$  ?    +0

10000000  $\Rightarrow$  ?    -0

32个0    31个0

原码  $\Rightarrow$  反码

+0: 000...00 (32个0)

-0: 111...11 (32个1)

↑  
负

+0: 000...00 (32个0)

-0: (111...11 + 1) = 0...00 (32个0)

↑  
+0 和 -0 是一样的, 都是 0.

反码: 正数不变, 负数取反

补码: 正数不变, 负数为反码 + 1 (取模)

[100...000] 补:

位运算 &: “按位与”    两个二进制数

6 & 11

6  $\rightarrow$  0110

11  $\rightarrow$  1011

0110  
& 1011

0010  $\Rightarrow$  2

有一个 0 就是 0

两个 1 才是 1

6 & 11 = 2

假设 X=6

0000 0110

补码: 0000 0110

按位取反 (负数)

-X = -6

1000 0110

反码: 1111 1001

[-6] 补码: 反码 + 1 : 1111 1010

↑  
负数

↑  
负数

X & (-X)

0000 0110

& 1111 1010

0000 0010  $\Rightarrow$  2

假设 X=12

原: 0000 1100

[12] 补: 0000 1100

-X = -12

原: 1000 1100

[-12] 反: 1111 0011

[-12] 补: 1111 0011 + 1 = 1111 0100

↑  
符号位

↑  
按位取反

12 & (-12) : 0000 1100

12  $\rightarrow$  4

& 1111 0100

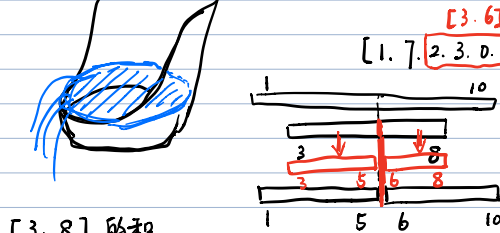
0000 0100 = 4

getSum(x) 求 [1...x]    若要求 [l...r]    getSum(r) - getSum(l-1)    前缀和

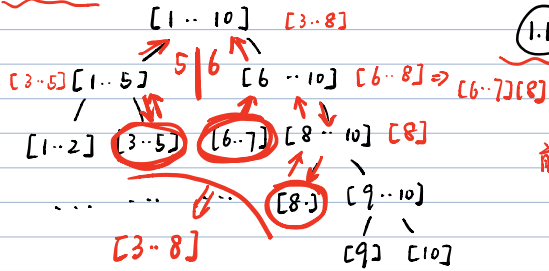


前缀

$$[1 \dots r] - [1 \dots L-1] = [L \dots R]$$

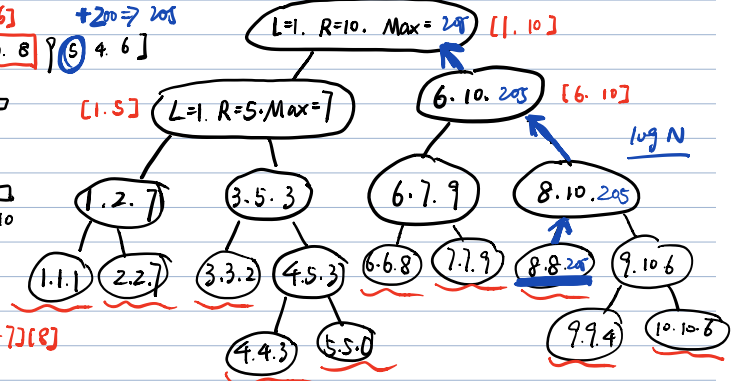


$[3 \dots 8]$  的和



拆分:  $[3 \dots 8] = [3 \dots 5] + [6 \dots 8]$

$[3 \dots 5]$  在左边  $[6 \dots 8]$  在右边



前两个数字相等:  $L=R$ , 含义: 区间长度为1, 最低层的叶子节点.