Bit manipulation 1

this session
mostly recap

- Bases
- operators & properties
- Left & Right Shift
- check bit
- count bit
- toggle bit
- set/unset ith bit
- set x Continous bits

Bit Manipulation 1

Decimal Number system $\{0,1,2,3,4,5,6,7,8,9\}$ Base 10

$$\overset{2}{3}\,\overset{1}{4}\,\overset{0}{2} = 3\times10^2+4\times10^1+2\times10^0$$

$$\longleftarrow$$

$$(\overset{3}{2}\,\overset{2}{5}\,\overset{1}{6}\,\overset{0}{3})_{10} = 2\times10^3+5\times10^2+6\times10^1+3\times10^0$$

Binary Number System $\{0,1\}$ Base 2

$$\overset{2}{1}\,\overset{1}{1}\,\overset{0}{0}\,_2 = 1\times2^2+1\times2^1+\underline{0\times2^0} = 6$$

$$\overset{3}{1}\,\overset{2}{\underline{0}}\,\overset{1}{1}\,\overset{0}{1}\,_2 = 1\times2^3+1\times2^1+1\times2^0 = 11$$

base 8 $(127)_8 = 1\times8^2+2\times8^1+7\times\overset{1}{8^0}$

$(12\underline{8})_8 =$

base 16 $\{0,1,2,3,4,5,6,7,8,9,\overset{10}{A},\overset{11}{B},\overset{12}{C},\overset{13}{D},\overset{14}{E},\overset{15}{F}\}$

$$(\overset{3}{A}\,\overset{2}{F}\,\overset{1}{3}\,\overset{0}{2})_{16}$$

$$10\times16^3+15\times16^2+3\times16^1+2\underbrace{\times16^0}_{1}$$

base 64 Full stack

Front devs

Cookies

token

base2 : binary   one sigle digit in base 2 : bit

Bitwise operations

→ {AND, OR, NOT, XOR, Left shift, right shift}

   & | !   ^   <<   >>

or

~

A	B	A&B	A|B	!A	A^B
0	0	0	0	1	0
0	1	0	1	1	1
1	0	0	1	0	1
1	1	1	1	0	0

A^B
adding
without carry
^
toggling

Bitwise operation on decimal numbers:

5 → 101
6 → 110 &
100 =4

5 & 6 =4

	5	4	3	2	1	0
A : 20	0	1	0	1	0	0
B : 45	1	0	1	1	0	1

|

A|B   1 1 1 1 0 1 → 61
5 4 3 2 1 0

	7	6	5	4	3	2	1	0
A : 92	0	1	0	1	1	1	0	0
B : 154	1	0	0	1	1	0	1	0

0 0 0 1 1 0 0 0

A&B = 24

	7	6	5	4	3	2	1	0
!92	0	1	0	1	1	1	0	0

1 0 1 0 0 0 1 1 → 163

	7	6	5	4	3	2	1	0
A : 92	0	1	0	1	1	1	0	0
B : 154	1	0	0	1	1	0	1	0

A^B   1 1 0 0 0 1 1 0 → 198

Properties:

1) $A \& 1 = ?$

$A=10$ 1

$$\begin{array}{r} 1010 \\ \& \; 000\boxed{1} \\ \hline 000\boxed{0} \end{array} \qquad \begin{array}{r} 9 \\ \& \; 1 \end{array} \quad \begin{array}{r} 1001 \\ 000\boxed{1} \\ \hline 000\boxed{1} \end{array}$$

$$A \& 1 \begin{cases} 0 & \text{even} \\ 1 & \text{odd} \end{cases}$$

2) $A \& 0 = 0$

$$\begin{array}{r} 101 \\ 000 \; \& \\ \hline 000 \end{array}$$

3) $A \& A = A$

$$\begin{array}{r} 101 \\ 101 \; \& \\ \hline 101 \end{array}$$

4) $A | 0 = A$

$$\begin{array}{r} 101 \\ 000 \\ \hline 101 \end{array} \quad | \text{ OR}$$

5) $A | A = A$

$$\begin{array}{r} 101 \\ 101 \\ \hline 101 \end{array} \quad \text{OR}$$

6) $A \wedge 0 = A$

$$\begin{array}{r} 101 \\ 000 \\ \hline 101 \end{array}$$

7) $A \wedge A = 0$

$$\begin{array}{r} 101 \\ 101 \; \wedge \\ \hline 000 \end{array}$$

$a+b = b+a$

8) Commutative property

a&b = b&a

a|b = b|a

a^b = b^a

9) Associative Property $(a+b)+c = a+(b+c)$

(a&b)&c = a&(b&c) = a&b&c

(a|b)|c = a|(b|c) = a|b|c

(a^b)^c = a^(b^c) = a^b^c

ex $a\,\hat{}\,b\,\hat{}\,c\,\hat{}\,a\,\hat{}\,b =$

$\xrightarrow{8} a\,\hat{}\,a\,\hat{}\,b\,\hat{}\,b\,\hat{}\,c \overset{9}{\underset{=}{\rightarrow}} \underbrace{\overbrace{(a\,\hat{}\,a)}^{0}}_{7}\,\hat{}\,\underbrace{\overbrace{(b\,\hat{}\,b)}^{0}}_{7}\,\hat{}\,c$

$\underset{6}{\overset{=}{\rightarrow}} \underbrace{0\,\hat{}\,0}_{0}\,\hat{}\,c \underset{6}{\overset{=}{\rightarrow}} 0\,\hat{}\,c = c$

a<<3

int | 4Bytes
8Bytes
1Bytes

left shift : <<

7 6 5 4 3 2 1 0

a=45 | 0 0 1 0 1 1 0 1 = 45 x2

a<<1 | 0 | 0 1 0 1 1 0 1 0 = 90 x2

a<<2 | 0 | 1 0 1 1 0 1 0 0 = 180

a<<3 | 1 | 0 1 1 0 1 0 0 0 = 104

360?!

s

256+104=360

Right shift >>

a=20 | 0 0 0 1 0 1 0 0 = 20 /2

a>>1 | 0 0 0 0 1 0 1 0 | 0 = 10 /2

a>>2 | 0 0 0 0 0 1 0 1 | 0 = 5 /2

a>>3 | 0 0 0 0 0 0 1 0 = 2 /2

a>>4 | 0 0 0 0 0 0 0 1 = 1 /2

a>>5 | 0 0 0 0 0 0 0 0 = 0

$\log_2 n$

- check bit
- unset bit
- count of set bit

Usage of $\ll$

shifting the "1"

8 bit

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

$1 \ll i$

$1 \ll 2$:

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

$1 \ll 4$:

7	6	5	4	3	2	1	0
0	0	0	1	0	0	0	0

$\ll$ | 1 ①
 ^ ②
 & ③

- checking and/or setting specific bits

① $N|(1 \ll i)$

$N = 45 \rightarrow$

$$\begin{array}{lc} & 5\,4\,3\,2\,1\,0 \\ N=45 & 101101 \\ (1\ll 2): & 000100 \ \text{(OR)} \\ \hline \text{OR} & 101101 \end{array} \qquad \begin{array}{cl} 5\,4\,3\,2\,1\,0 & \\ 101101 & \\ 010000 & (1\ll 4)\ \text{(OR)} \\ \hline 111101 & \end{array}$$

$$N|(1\ll i)\begin{cases} N & \text{if bit } i \text{ is set} \\ N+(1\ll i) & \text{if } i \text{ is unset} \end{cases}$$

② $N \wedge (1 \ll i)$

$$\begin{array}{lc} & 5\,4\,3\,2\,1\,0 \\ N=45 & 101101 \\ (1\ll 2) & 000100 \ \text{XOR} \\ \hline & 101001 \end{array} \qquad \begin{array}{lc} & 5\,4\,3\,2\,1\,0 \\ & 101101 \\ (1\ll 4) & 010000 \ \text{(XOR)} \\ \hline & 111101 \end{array}$$

$N \wedge (1 \ll i) \rightarrow$ toggle bit i

③ $N \& (1 \ll i)$

$$\begin{array}{lc} & 5\,4\,3\,2\,1\,0 \\ N=45 & 101101 \ \text{AND} \\ (1\ll 3) & 001000 \\ \hline & 001000 \end{array} \qquad \begin{array}{cl} 5\,4\,3\,2\,1\,0 & \\ 101101 & \text{AND} \\ 000010 & (1\ll 1) \\ \hline 000000 & \end{array}$$

$$N\&(1\ll i)\begin{cases} 1\ll i & \text{(if ith bit is set)} \\ 0 & \end{cases}$$

P1 Given a positive integer n and bit # i

true/ false ← check if the i th bit is set:

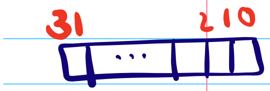
```
bool checkBit(N,i){
    ret N&(1<<i)!=0          ==(1<<i)
}
        ret N|(1<<i)==N
        ret N^(1<<i)<N

        ret N>>i&1  this?  Carrect
```

positive

P2 Count the number of set bits in N

N = $\underline{1}0\underline{1}\underline{1}0\underline{1}$ → ans = 4



int → 32 bit

double 64 bit
long 64 bit

long long 128 bit
long double 128 bit

TC:
O(BITs)

```
int countBit1(int n){
    ans=0              32 ↖
    for(i=0;i<BITs;i++){
# bits  if(checkBit(n,i))  ans++
    }
    ret ans
}
```

```
int CountBit2(int n){
    ans=0
    while(n>0){
        if(n&1==1) ans++;
        n =n>>1          (n >>=1)
    }
    ret ans
}
```

Quiz
TC: O(log n)
SC: O(1)

7	6	5	4	3	2	1	0
0	0	0	0	1	0	1	0

P3 Given a positive integer n and bit # i
toggle the i th bit in n :

```
7654 32 10
00101101                toggle bit 2
 int toggle(int n, int i){
     ret    n ^(1<<i)
```

ex chmod in Unix
rwx

P4 Write two functions to | I set
                          | II unset   ith bit of n

```
I   int set (int n, int i){
        ret   n | (1<<i)
    }

II  int unset (int n, int i){
        ret   n&!(1<<i)
    }

    if(checkBit(n,i))
        ret   n^(1<<i)
```

```
   !(00100)
=  (11011)
```

```
     43210
     10101
&  !(00100)  <- 1<<2
     11011
     10001
```

P5 Unset X Continious bits in N from right.

0000 1111

N= 1110 1101

7654 3210

1110 0000

X=4

1<<4

16

00010000

76543210

00000 1111

15

```
int unsetBits (n , x){
    ans = n
    for(i=0; i <= x ; i++){
        ans = unset(ans, i)
    }
    ret ans
}
```

n& !((1<<x)-1)