7:35AM

Topics:- $\ll$ applications in bit Manipulation
- check bit
- unset bit
- count of set bit
- negative binary

Bit Manipulation 2

Usage of $\ll$

shifting the "1"

8 bit

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

$1 \ll 2$:

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

$1 \ll 4$:

0	0	0	1	0	0	0	0

$\ll$ | 1 ① ^② &③

- checking and/or setting specific bits

$$\begin{array}{llll}
 & \overset{543210}{101101} & & \overset{543210}{101101} \\
N=45 \rightarrow & 000100 \text{ (OR)} & & 010000 \; (1\ll 4) \text{ (OR)} \\
(1\ll 2): & \overline{101101} \rightarrow 45 & & \overline{111101} \rightarrow 61 \\
\text{OR} & & & \;\, 543210
\end{array}$$

① $N | (1\ll i)$

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

② $N \wedge (1\ll i)$

$$\begin{array}{llll}
N=45 & \overset{543210}{101101} & & \overset{543210}{101101} \\
(1\ll 2) & 000100 \text{ XOR} & (1\ll 4) & 010000 \text{ (XOR)} \\
 & \overline{101001} \neq 45 & & \overline{111101} \neq 45
\end{array}$$

$N \wedge (1\ll i) \rightarrow$ Flips ith bit

③ $N \& (1\ll i)$

$$\begin{array}{llll}
N=45 & \overset{543210}{101101} \text{ AND} & & \overset{543210}{101101} \text{ AND} \\
(1\ll 3) & 001000 & & 000010 \; (1\ll 1) \\
 & \overline{001000} \neq 0 & & \overline{000000} == 0
\end{array}$$

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

P1 Unset i th bit of a number if it is set otherwise no change

ex

```
        5 4 3 2 1 0
N = 45  1 0 1 1 0 1
i = 2   1 0 1 0 0 1
i = 4   1 0 1 1 0 1
```

1.1 check ith Bit

```
if(checkBit(N,i)) N = N^(1<<i)
~~else N=N~~ → not needed
ret N

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

ret N^(1<<i)<N
```

→ why does this work?

- if ith bit is set result will be smaller
- if ith bit is unset result will be larger

(MSB) most significant bit

least significant bit (LSB)

7 8bit

xxxx1

xxxx0

$\boxed{\text{return } n\&1==1}$

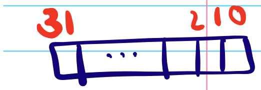
true

```
      isOdd
bool func1(N){
    return N&1 → not boolean
}
```

getLeastSignificant

P2 Count the number of set bits in N → is positive

N = 1 0 1 1 0 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
    for(int i=0; i<BITs(32); i++){
        if( checkBit(n,i) ) ans += 1
    }
    ret ans;
}
```

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

Quiz
TC: $O(\log_2 n)$
SC: O(1)

ans
1
2

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

N>>1

$\log_2 n$

Negative binary numbers

intro:

Most significant bit

$$\begin{array}{lr} & {\scriptstyle 1\,1\,1} \\ ⑦ & 0111 \\ + & 1 \\ \hline 8 & 1000 \\ & {\scriptstyle 3\,2\,1\,0} \end{array}$$

$$\begin{array}{lr} & {\scriptstyle 7\,6\,5\,4\,3\,2\,1\,0} \\ ⓐ & 01111111 \\ + & 1 \\ \hline & 10000000 \\ & {\scriptstyle 7\,6\,5\,4\,3\,2\,1\,0} \end{array}$$

$01111111 = 2^7 - 1 = 128 - 1 = 127$

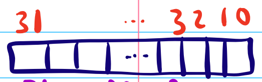
$10000000 \rightarrow 2^7$

$2^0 + 2^1 + 2^2 + 2^3 + \dots + 2^6$

$$\frac{a(r^i - 1)}{r - 1}$$

$n + 1 = 2^i$

$n = 2^i - 1$



$2^{31} > 2^{30} + 2^{29} + \dots + 2^2 + 2^1 + 2^0$

$(-45)_{10} = (?)_2$

1 - Convert absolute number to binary ✓
2 - invert all bit 1's Complement ✓
3 - add 1 to the inverted number 2's Complement

why like this?
why this works?

$45 + B = 0$ ✓
B will be -45 if

$$\begin{array}{lr} & {\scriptstyle 1\,1\,1\,1\,1\,1\,1} \\ & {\scriptstyle 7\,6\,5\,4\,3\,2\,1\,0} \\ \text{8 bit} & 00101101 \ (+45) \\ \text{1's Comp.} & 11010010 \\ + & 1 \\ \hline B \leftarrow \text{2's Comp.} & 11010011 \\ (-45) & {\scriptstyle 7\,6\,5\,4\,3\,2\,1\,0} \end{array}$$

$= 2^7 + 2^6 + 2^4 + 2^1 + 2^0 = 211$

$= \underline{-1 \times 2^7} + 2^6 + 2^4 + 2^1 + 2^0 = -128 + \underbrace{(64 + 16 + 2 + 1)}_{83}$

$= -45$

MSB | 1 → negative
MSB | 0 → positive

Quiz $(-3)_{10} = (?)_2$

$(+3)_{10} \rightarrow (\qquad)_2$

$+3 = (00000011)_2$

$$\begin{array}{lr} & 11111100 \\ + & 1 \\ \hline & 11111101 \\ & {\scriptstyle 7\,6\,5\,4\,3\,2\,1\,0} \end{array}$$

$-1 \times 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 1$

$-128 + \underbrace{(64 + 32 + 16 + 8 + 4 + 1)}_{125} = -3$

Quiz $((45)_{10})_2 + ((-45)_{10})_2$

$$\begin{array}{r}
\cancel{1}\;1\,1\,1\,1\,1\,1\,1 \\
0\,0\,1\,0\,1\,1\,0\,1 \\
1\,1\,0\,1\,0\,0\,1\,1 \\
\hline
0\,0\,0\,0\,0\,0\,0\,0
\end{array} \rightarrow 0$$

N+B=0

$(45-12)_2 = (45)_2 + (-12)_2 = (33)_2$

$$\begin{array}{r}
0000\;1100 \\
\hline
1111\;0011 \\
1 \\
\hline
1111\;0100
\end{array}$$

$$\begin{array}{r}
\cancel{1}\;1\,1\,1\,1\,1 \\
0\,0\,1\,0\,1\,1\,0\,1 \\
1\,1\,1\,1\,0\,1\,0\,0 \\
\hline
0\,0\,1\,0\,0\,0\,0\,1 \\
5\,4\,3\,2\,1\,0
\end{array}$$

$2^5+2^0=33$

Range of int

MSB 31 is reserved for sign

Max $0\overbrace{1111\,11\cdots1}^{31} = 2^{31}-1 \simeq +2\times10^9$

Min $1\overbrace{000\cdots0}^{31} = 2^{31}\times-1 = -2^{31} \simeq -2\times10^9$

Range of long

Max $0\overbrace{1111\cdots1}^{63} = 2^{63}-1 \simeq +9\times10^{18}$

Min $1\underbrace{0000\cdots0}_{63} = -2^{63} \simeq -9\times10^{18}$

P3 Calculate sum of all elements in array a[ ]

```
        int n = a.len
wrong → int sum = 0
            for (i=0; i<n; i++){
                sum += a[i]
            }
            ret sum
                     → wrong
```

Constraints

$1 <= N <= 10^5$

$1 <= a[i] <= 10^6$

$$a\{\underbrace{10^6, 10^6, \ldots, 10^6}_{10^5}\}$$

$sum = 10^{11}$

P4 For given two integers a and b, return a*b.

```
int f (int a, int b){
        int ans = 0
        ans = a*b
        ret ans   X
}
```

$a <= 2*10^9$

$b <= 2*10^9$

$= 2 \times 10^9 \times 2 \times 10^9$

$= 4 \times 10^{18}$

```
long f (int a, int b){
        long ans = 0
        ans = a*b  X      C, C++, Java
        ret ans
}
```

←
101000

```
long f (int a, int b){
|        long ans = 0
|        ans =(long)( a * b ) ✗
|        ret ans
}
```

```
long f (int a, int b){
|        long ans = 0
|        ans =((long)(a))* b ✓
|        ret ans
}
```