Lecture ÷ Bit Manipulation - 1

Agenda

- Basics of logical operators
- Power of left shift operators
- Check ith bit is set ?
- Total number of set bits in n.
- Unset ith bit of a number.
- Set bits in a range.

Truth table for bit-wise operators.

0 dominates   1 dominates   Same same puppy shame.

a	b	a & b	a | b	a ^ b	~ a
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

&	→	and
|	→	or
^	→	xor
~	→	not

AND properties

1.> Even | odd number.

In binary representation —

Even number ⇒ LSB = 0

odd number ⇒ LSB = 1

35 ⇒ 100011

63 ⇒ 111111

7 ⇒ 111

9 ⇒ 1001

8 ⇒ 1000

18 ⇒ 10010

24 ⇒ 11000

26 ⇒ 11010

A & 1 = 1 (odd)

else → even number

2.> A & 0 = 0

3.> A & A = A

OR properties

1.> $A \mid 0 = A$

2.> $A \mid A = A$

XOR properties

1.> $A \wedge 0 = A$

2.> $A \wedge A = 0$

Commutative property

$A \,\&\, B \Rightarrow B \,\&\, A$

$A \mid B \Rightarrow B \mid A$

$A \wedge B \Rightarrow B \wedge A$

Associative property

$(A \,\&\, B) \,\&\, C \Rightarrow A \,\&\, (B \,\&\, C)$

$(A \mid B) \mid C \Rightarrow A \mid (B \mid C)$

$(A \wedge B) \wedge C \Rightarrow A \wedge (B \wedge C)$

Quiz $a \hat{} b \hat{} a \hat{} d \hat{} b \Rightarrow \underbrace{a \hat{} a}_{0} \hat{} \underbrace{b \hat{} b}_{0} \hat{} d$

$0 \hat{} 0 \hat{} d = \boxed{d}$ Ans.

Quiz $1 \hat{} 3 \hat{} 5 \hat{} 3 \hat{} 2 \hat{} 1 \hat{} 5 \Rightarrow \underbrace{1 \hat{} 1}_{0} \hat{} 2 \hat{} \underbrace{3 \hat{} 3}_{0} \hat{} \underbrace{5 \hat{} 5}_{0}$

$\boxed{2}$ Ans.

Left shift operator

$a << n \Rightarrow a * 2^{n}.$

$1 << n \Rightarrow 1 * 2^{n} = 2^{n}.$

$2 << 3 \Rightarrow 2 * 2^{3} \Rightarrow 2 * 8 = 16.$

Right shift operator

$a >> n \Rightarrow a \,|\, 2^{n}.$

$1 >> n \Rightarrow \frac{1}{2^{n}}.$

$16 >> 2 \Rightarrow \frac{16}{2^{2}} = \frac{16}{4} = 4.$

Quiz $1 << 3 \Rightarrow 1 * 2^{3} = \boxed{8}$ Ans.

$a << n \Rightarrow a * 2^{n}.$

Power of left shift operator | Amazon |

Property 1

n = 45 ⇒ 101101 [and]

case1: n = 45 ⟶

$$\begin{array}{cccccc} 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{array}$$

1 << 2 ⟶ & 0 0 0 1 0 0

4

0 0 0 1 0 0

case2: n = 45 ⟶ 1 0 1 1 0 1

1 << 3 ⟶ & 0 0 1 0 0 0

8

0 0 1 0 0 0

case3: n = 45 ⟶ 1 0 1 1 0 1

1 << 4 ⟶ & 0 1 0 0 0 0

16

0 0 0 0 0 0

⋮

n & (1 << k)

1<<k [kth bit is 1]

0 [kth bit is 0]

Property 2 OR

$$n = 45 \longrightarrow \begin{array}{cccccc} 1 & 0 & 1 & 1 & 0 & 1 \end{array}$$

$$\underset{2}{1 << 1} \longrightarrow \Big| \begin{array}{cccccc} 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 1 & 0 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

$$n = 45 \longrightarrow \begin{array}{cccccc} 1 & 0 & 1 & 1 & 0 & 1 \end{array}$$

$$\underset{4}{1 << 2} \longrightarrow \Big| \begin{array}{cccccc} 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 1 & 0 & 1 \\ \hline \end{array}$$

$$n = 45 \longrightarrow \begin{array}{cccccc} 1 & 0 & 1 & 1 & 0 & 1 \end{array}$$

$$\underset{8}{1 << 3} \longrightarrow \Big| \begin{array}{cccccc} 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 1 & 1 & 0 & 1 \\ \hline \end{array}$$

$$n = 45 \longrightarrow \begin{array}{cccccc} 1 & 0 & 1 & 1 & 0 & 1 \end{array}$$

$$\underset{16}{1 << 4} \longrightarrow \Big| \begin{array}{cccccc} 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 1 & 1 & 0 & 1 \\ \hline \end{array}$$

⋮

$n \mid (1 << k)$

- Kth bit = 0 — Kth bit of n is set. [Kth bit becomes 1]
- kth bit = 1 — No change.

Property 3   xor operator.

$$\begin{array}{lll} n = 45 & \longrightarrow & 1\ 0\ 1\ 1\ 0\ 1 \\ \underset{2}{1 \ll 1} & \longrightarrow & {}^\wedge\ 0\ 0\ 0\ 0\ 1\ 0 \\ \hline & & 1\ 0\ 1\ 1\ 1\ 1 \\ \hline \end{array}$$

$$\begin{array}{lll} n = 45 & \longrightarrow & 1\ 0\ 1\ 1\ 0\ 1 \\ \underset{8}{1 \ll 3} & \longrightarrow & {}^\wedge\ 0\ 0\ 1\ 0\ 0\ 0 \\ \hline & & 1\ 0\ 0\ 1\ 0\ 1 \\ \hline \end{array}$$

$$\begin{array}{lll} n = 45 & \longrightarrow & 1\ 0\ 1\ 1\ 0\ 1 \\ \underset{16}{1 \ll 4} & \longrightarrow & {}^\wedge\ 0\ 1\ 0\ 0\ 0\ 0 \\ \hline & & 1\ 1\ 1\ 1\ 0\ 1 \\ \hline \end{array}$$

$\vdots$

$n \wedge (1 \ll k)$

- Kth bit = 0   Kth bit will change to 1
- Kth bit = 1   Kth bit will change to 0.

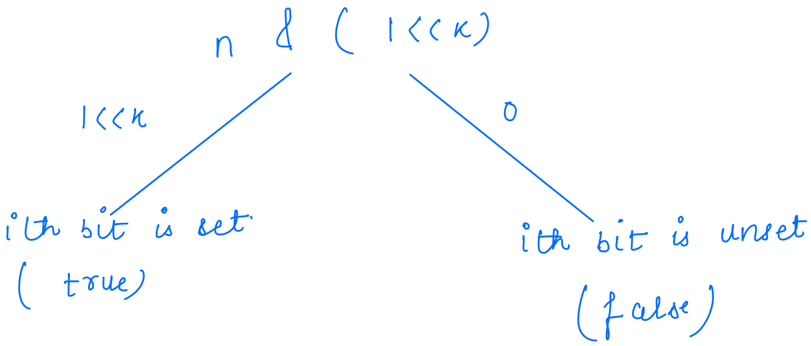
Break: 7:59 – 8:10 AM

Qu. check whether ith bit is set or not?

input $n = 45$ $\begin{array}{cccccc} 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{array}$

$i = 2 \longrightarrow$ yes

Approach



code

```
boolean checkSetBit (int n, int k) {
        if(n & (1<<k) == 0) {
                return false;
        }
      return true;
}
```

TC: O(1)

SC: O(1)

Qu. Count total numbers of set bit in n.

input   n = 12

1 1 0 0 ⇒ ans = 2

(45) 1 0 1 1 0 1 ⇒ ans = 4.

Approach 1

[& <<]

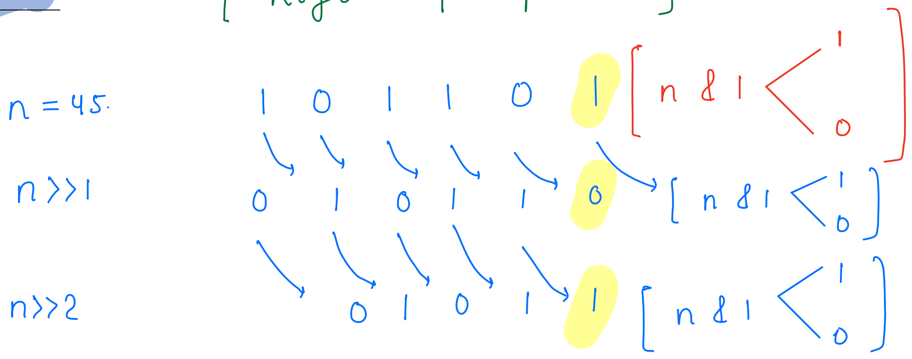
```
                    int countTotalsetBits (int n) {

                          int cnt = 0;

O(1) ≃ O(32) ——————  for(i=0; i<32; i++) {
                    O(1) —— if(checksetBit(n, i)) {
                                  cnt++;
                              }
                          }

                          return cnt;
                    }
```

TC: O(1)

SC: O(1)

Approach 2

>> [Right shift operator]



```
int countTotalsetBits (int n) {
        int cnt=0;
        while( n > 0)  {
               if(n & 1)  {
                      cnt + =1;
               }
               n = n >>1;
        }
        return ans;
}
```

TC: O(1) ≃ O(Logn) [upperbound = 32]

SC: O(1)

Qu:3 Unset the ith bit of a number if ith bit is set. [toggle]

input: n = 6 $\overset{3}{0}\ \overset{2}{1}\ \overset{1}{1}\ \overset{0}{0}$

i = 2 0 0 1 0 [ans = 2]

Approach

```
int unset ithBit (int n, int i) {
        if( checksetBit( n, i)) {
            n = n ^ (1<<i);
        }
        return n;
}
```

TC: O(1)

SC: O(1)

$$\boxed{n = n \ \& \ \sim(1<<i)}$$

Qu.4 A group of computer scientists is working on a project that involves encoding binary numbers. They need to create a binary number with a specific pattern for their project. The pattern requires A 0's followed by B 1's followed by C 0's. To simplify the process, they need a function that takes A, B and C as inputs and return the decimal value of resulting binary number. Can you help them by writing a function that can solve this problem efficiently. ?

Constraints

$0 <= A, B, C <= 20$

input

A = 4

B = 3

C = 2.

pattern

requires A 0's followed by B 1's followed by C 0's.

$$\underbrace{0000\overset{4}{1}\overset{3}{1}\overset{2}{1}\overset{1}{0}\overset{0}{0}} = \underline{28 \text{ Ans}}$$

Hint:

8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0
				1	1	1		

→ n

0 0 0 0 1 1 1 0 0

$n = n \mid (1<<2)$

$n = n \mid (1<<3)$

$n = n \mid (1<<4)$

$[\,C, \quad B+C-1\,]$

$2 \quad 2+3-1$

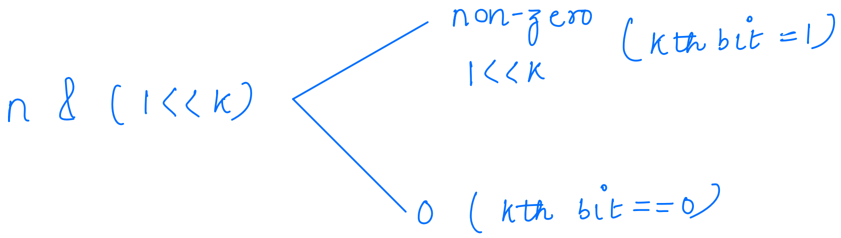
4

2, 3, 4 ✓

```
long solve( int A, int B, int C) {
    long ans = 0;
    for(i=c; i< B+c; i++){
        ans = ans | (1<<i);
    }
    return ans;
}
```

$$TC: O(\overset{1 \,\|\, B}{n})$$

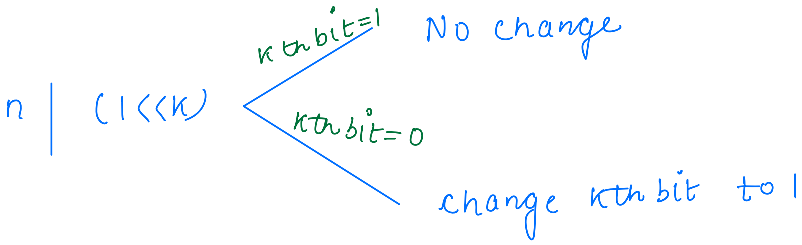
$$SC: O(1)$$



n & (1<<k)

non-zero (kth bit =1)
1<<k

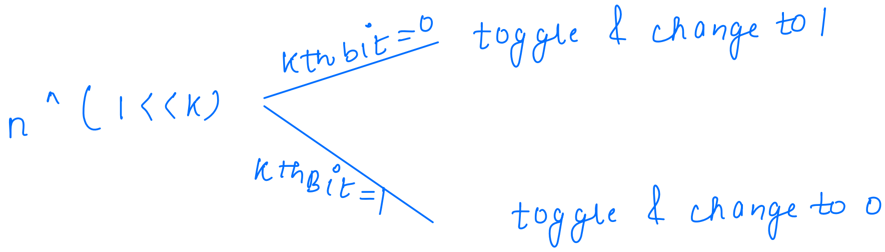
0 (kth bit==0)



n | (1<<k)

kth bit=1 No change

kth bit=0 change kth bit to 1



n ^ (1<<k)

kth bit=0 toggle & change to 1

kth Bit=1 toggle & change to 0

Thankyou ☺

Doubts

$a = 2$

$b = 6$

$c = 3$

10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	1	1	1	1	0	0	0

[2 , 7]