

# Bit Manipulation 1

## Bitwise Operators

A	B	$A \& B$	$A   B$	$A \wedge B$	$\sim A$
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

## Properties

1.  $A | 0 = A$

$$A = \begin{array}{r} 1010 \\ 0000 \\ \hline 1010 \end{array}$$

2.  $A | A = A$

$$1 | 1 = 1 \quad 0 | 0 = 0$$

3.  $A \wedge 0 = A$

$$1 \wedge 0 = 1 \quad 0 \wedge 0 = 0$$

4.  $A \wedge A = 0$

$$1 \wedge 1 = 0 \quad 0 \wedge 0 = 0$$

5.  $A \& 0 = 0$

6.  $A \& A = A$

$0 \& 0 = 0$

$1 \& 1 = 1$

## 7. Odd / Even

4  $\rightarrow$  1 0 0  
6  $\rightarrow$  1 1 0  
8  $\rightarrow$  1 0 0 0  
10  $\rightarrow$  1 0 1 0

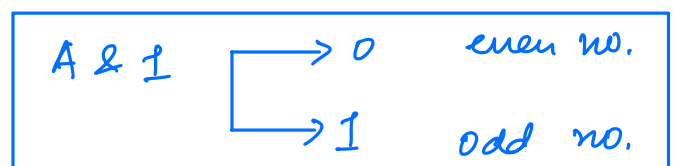
Even  $\rightarrow$  ..... 0

5  $\rightarrow$  1 0 1  
7  $\rightarrow$  1 1 1  
9  $\rightarrow$  1 0 0 1  
11  $\rightarrow$  1 0 1 1

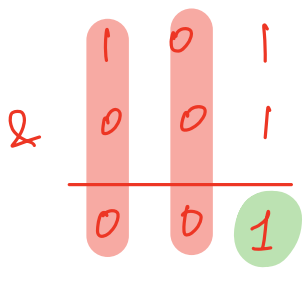
Odd  $\rightarrow$  ..... 1

last bit determines even/odd.

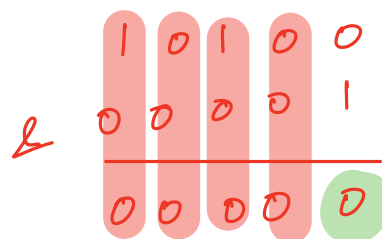
how to check last bit ?



$A = 5$



$A = 20$



## 8. Commutative Property

$$A \& B = B \& A$$

$$A | B = B | A$$

$$A \wedge B = B \wedge A$$

## 9. Associative Property

$$(A \& B) \& C = A \& (B \& C)$$

$$(A | B) | C = A | (B | C)$$

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

## Quiz

$$a \wedge b \wedge a \wedge d \wedge b$$

$$= (a \wedge a) \wedge (b \wedge b) \wedge d$$

$$= (0 \wedge 0) \wedge d$$

$$= 0 \wedge d = d$$

## Quiz

$$1^{\wedge} 3^{\wedge} 5^{\wedge} 3^{\wedge} 2^{\wedge} 1^{\wedge} 5$$

$$\Rightarrow (1^{\wedge} 1)^{\wedge} (3^{\wedge} 3)^{\wedge} (5^{\wedge} 5)^{\wedge} 2$$

$$= 0^{\wedge} 0^{\wedge} 0^{\wedge} 2 = 2$$

## Question

Given an integer array where every no. occurs twice except one number. find that unique no.?

Sol<sup>n</sup> :  $ans = \oplus i, ^{\wedge} a[i]$

$$ans = a[0]$$

$$\text{for } (i=1 \text{ to } n-1) \{$$

$$ans^{\wedge} = a[i]$$

}

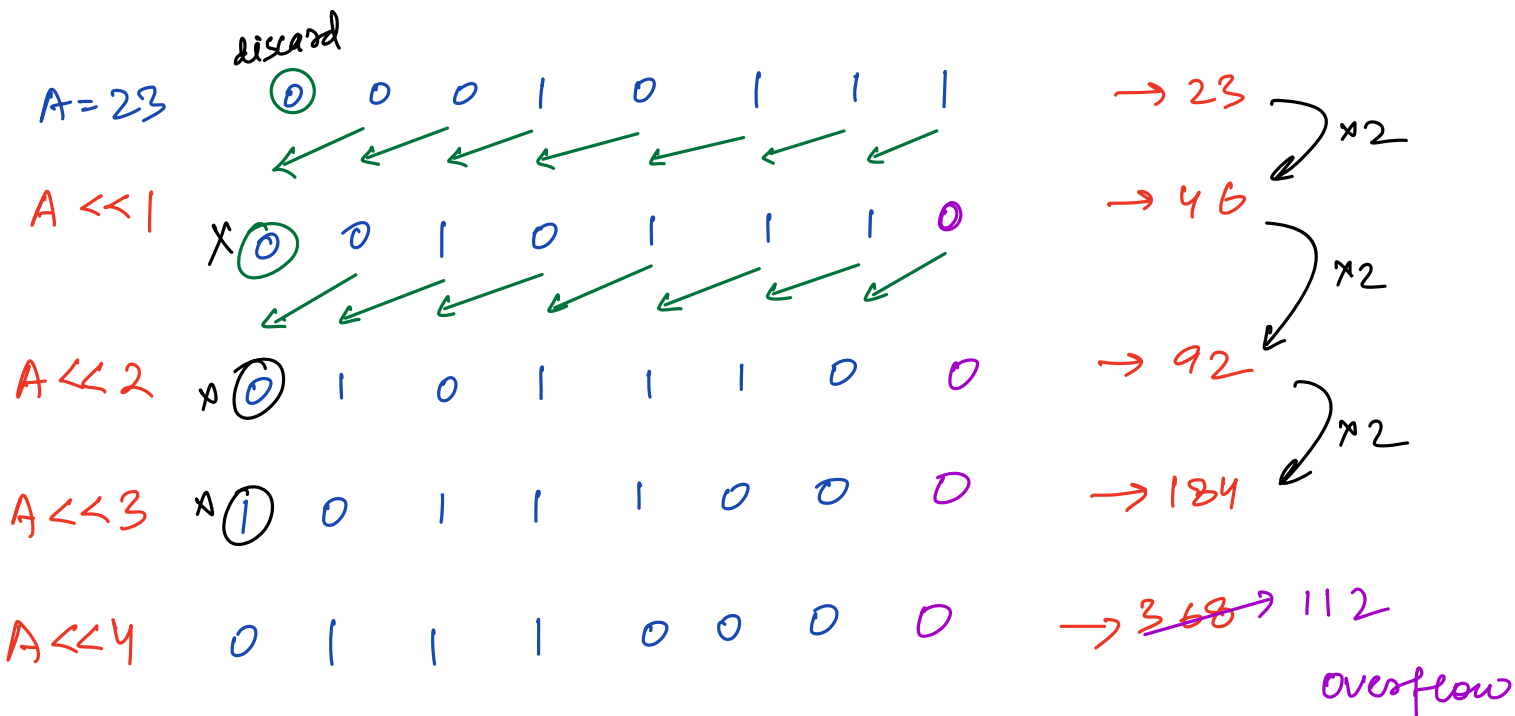
return ans

$$TC = O(N)$$

$$SC = O(1)$$

# Left Shift operator (<<)

for explaining  $\rightarrow$  8 bit numbers  
 $\hookrightarrow$  0 - 255



$$N \ll 1 = N \times 2$$

$$N \ll K = N \times 2^K$$

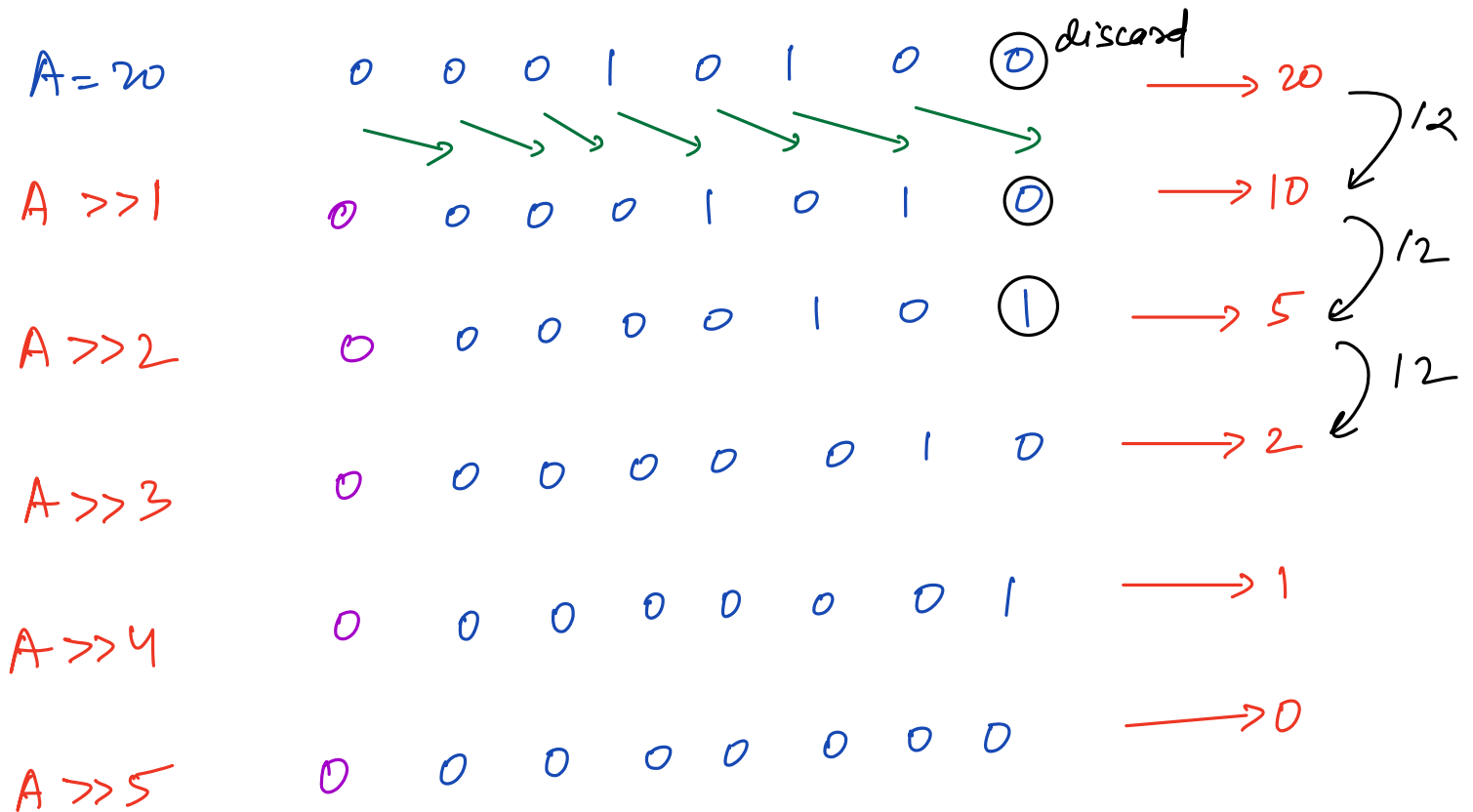
$$1 \ll n = 2^n$$

take care of overflow

$$x \times 2 > \text{INT\_MAX} \quad \text{X}$$

$$x > \frac{\text{INT\_MAX}}{2} \quad \checkmark \quad \text{overflow condition}$$

## Right Shift ( $\gg$ )



$$\begin{aligned} N \gg 1 &= N/2 \\ N \gg K &= N/2^K \end{aligned}$$

no overflow

## Quiz

$$1 \ll 3$$

$$1 \ll n = 2^n$$

$$\Rightarrow 2^3 = 8$$

## Power of left shift operator

$$(1 \ll K) = 2^K \quad 000\dots0 \mid 0000\dots0 \quad (\text{only } K^{\text{th}} \text{ bit is set})$$

$\uparrow$   
 $K^{\text{th}} \text{ bit}$

### 1. OR

$N \mid (1 \ll K) \Rightarrow \text{set } K^{\text{th}} \text{ bit of } N$

$N = 45$

$K = 3$

1	0	1	1	0	1
OR					
0	0	1	0	0	0
1	0	1	1	0	1

$= 45$  if  $K^{\text{th}}$  bit is already 1  $\Rightarrow$  no change

$N = 45$

$K = 4$

1	0	1	1	0	1
OR					
0	1	0	0	0	0
1	1	1	1	0	1

$= 61$

$\text{else} \rightarrow N + 2^K$

### 2. XOR

$N \wedge (1 \ll K) \rightarrow \text{toggle } K^{\text{th}} \text{ bit}$

$$N = 45$$

$$K = 3$$

XOR

1	0	1	1	0	1
0	0	1	0	0	0
1	0	0	1	0	1

$$= 37 \quad \left( \begin{array}{l} \text{if bit is} \\ 1 \end{array} = N - 2^K \right)$$

$$N = 45$$

$$K = 4$$

1	0	1	1	0	1
0	1	0	0	0	0
1	1	1	1	0	1

$$= 61$$

$$\text{else} = N + 2^K$$

### 3. AND

$$N \& (1 \ll K) \quad \begin{array}{l} \rightarrow 2^K \quad (K^{\text{th}} \text{ bit is set}) \\ \rightarrow 0 \quad (K^{\text{th}} \text{ bit is unset}) \end{array}$$

$$N = 45$$

$$K = 3$$

AND

1	0	1	1	0	1
0	0	1	0	0	0
0	0	1	0	0	0

$$= 2^3 (2^K)$$

$$\text{if } K^{\text{th}} \text{ bit is } 1 \Rightarrow 2^K$$

$$N = 45$$

$$K = 4$$

1	0	1	1	0	1
0	1	0	0	0	0
0	0	0	0	0	0

$$= 0$$

$$K^{\text{th}} \Rightarrow 0$$



## Unset $k^{\text{th}}$ bit

```
def unset(N, K) {
```

```
    X = N & (1 << K)
```

if  $X = 0$ ,  $k^{\text{th}}$  bit is already unset

```
    if (X > 0) {
```

```
        N = N ^ (1 << K)
```

```
    }
```

```
}
```

TC =  $O(1)$

SC =  $O(1)$

Check if  $k^{\text{th}}$  bit is set or unset?

```
def isSet(N, K) {
```

```
    X = N & (1 << K)
```

```
    if (X == 0) return false
```

```
    return true
```

```
}
```

## Question 1

Given an integer  $N$ , count total set bits in  $N$ .  
32 bit integer.

$N = 12$

1 1 0 0

ans = 2

ans = 0

for (i = 0 to 31) {

if (N & (1 << i) > 0) {

ans++

}

}

return ans

TC =  $O(1)$

SC =  $O(1)$

Another approach

ans = 0

while (N > 0) {

if (N & 1) {

ans++

}

TC =  $O(\log N)$

SC =  $O(1)$

$N = N >> 1$  //  $N = N/2$

}

return am

## Question

for each train, there is a special no.

Special no.: 28 bit no. where if  $i^{\text{th}}$  bit is set  
then train runs on that day

find train which runs the most?

$A = [ \overset{0}{20}, \overset{1}{7}, \overset{2}{10} ]$

20  $\rightarrow$  1 0 1 0 0

7  $\rightarrow$  1 1 1

11  $\rightarrow$  1 0 1 0

am = 1

code

```
def findTrain(a[], N) {
```

```
    curr_count = -1
```

```
    ans = -1
```

```
    for (i=0 to n-1) {
```

```
        c = countBits(a[i]) // return total set bits in a[i]
```

```
        if (c > curr_count) {
```

```
            curr_count = c
```

```
            ans = i
```

```
        }
```

```
    }
```

```
    return ans
```

```
}
```

$TC = O(N)$

$SL = O(1)$

Question

Create a binary number with specific pattern.

The pattern is : A 0's followed by B 1's followed by C 0's.

A, B, C is input

$0 \leq A, B, C \leq 20$

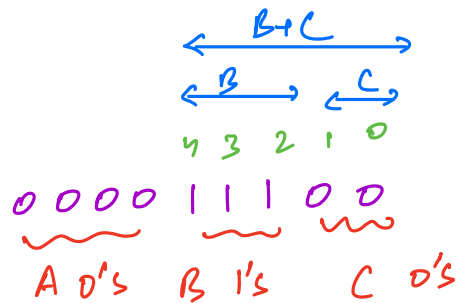
Return the integer.

eg

$$A = 4$$

$$B = 3$$

$$C = 2$$



⇒ 28

1. Ignore first A 0's.

Set bits from C to  $B+C-1$

long ans = 0

for (i = C to B+C-1) {

ans = ans | (1 << i)

}

return ans

TC =  $O(B)$

SC =  $O(1)$

max value of ans = 11111 0 00000  
20 1's 20 0's

⇒ ~40 bit no.

⇒ use long

$$2^3 - 1 = 7 = 111$$

$$2^4 - 1 = 15 = 1111$$

$$2^B - 1 = \underbrace{1111 \dots 1}_{B \text{ times}}$$

$$(2^B - 1) \ll C \Rightarrow \underbrace{1111}_{B \text{ times}} \underbrace{000}_{C \text{ times}}$$

$$((1 \ll B) - 1) \ll C$$

$$ans = ((\underline{1L} \ll B) - 1) \ll C$$

OR

$$\text{long } ans = 1$$

$$ans = ((ans \ll B) - 1) \ll C$$