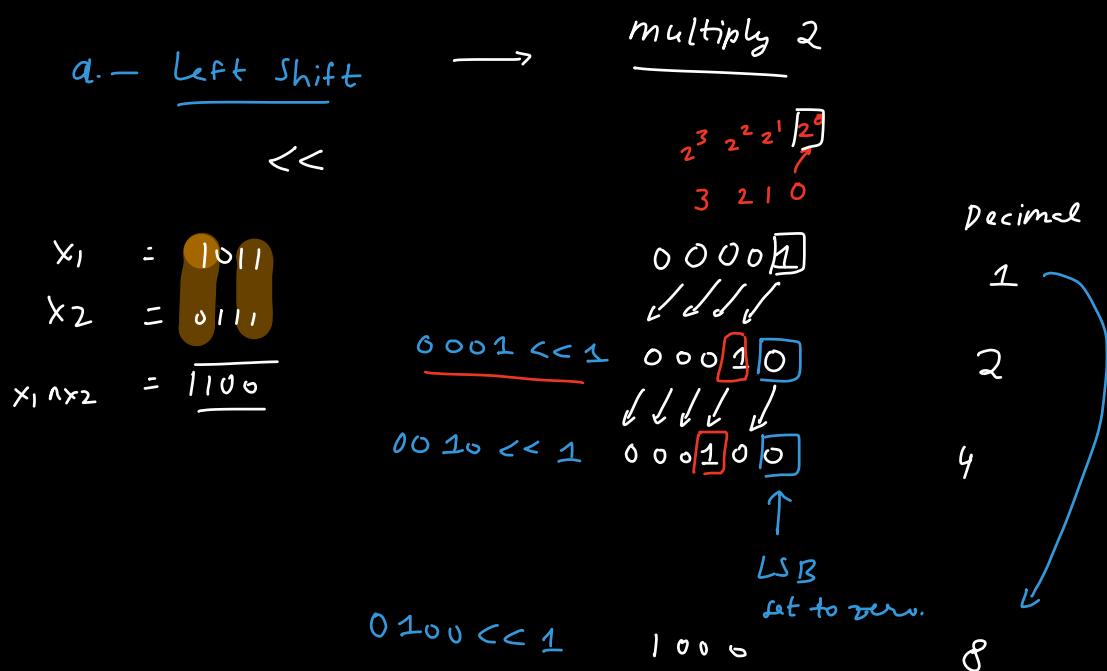


## Bit Manipulation - 2

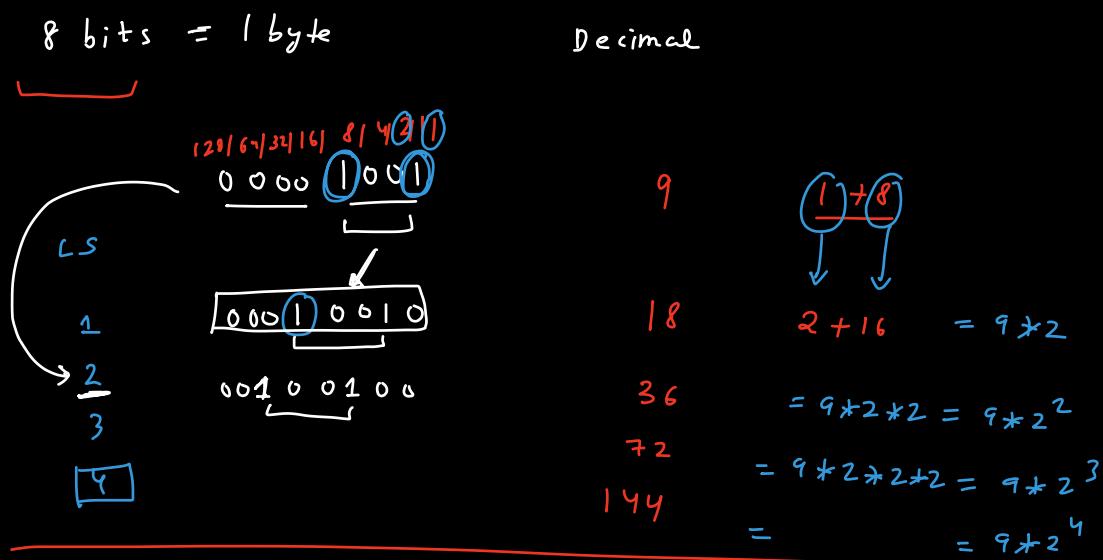
### A agenda

- 1. Bit Shifting Operators       $<<$ ,  $>>$
- 2. Bit Masking Concept + Problems
- 3. Problems

# ① Bit Shifting Operations



$$0001 \ll 3 = 1000$$



$$\text{decimal} \quad \quad \quad \text{decimal value}$$

(a)  $\ll b = a * 2^b \quad \quad \quad TC : O(1)$

$$\underline{\text{int}} \quad 32 \text{ bits} \quad 2^{32} - 1 \quad \approx 2 \times 10^9$$

Processor → instructions part of hardware.

$Q_{uiz-1}$  | << 4

$$= 1 + 2^4$$

= 16

$\alpha_{n_1, 2} - 2$        $30 < z$

$$= 30 \times 2^2$$

$$= 12_8$$

$a_{ui_2-3}$   $\underline{\&}$   $\underline{(\sim 0 \quad \ll 2)}$   $\Rightarrow \underline{a \& o = 0}$

$\sim 0$  |  $00\ 0000$        $0000$   $\rightarrow 0$

$(\sim 0)\ll 2$  |  $00\ 1111$        $1111$

$\& \underline{1011}$  |  $11\ 1111$        $\leftrightarrow$

$00\ 0000$        $1011$        $\leftrightarrow$

$=> 8$

b. Right Shift >>



$$0101 \gg 1 \quad \begin{array}{r} 010 \\ \downarrow \quad \downarrow \\ 0 \quad 2^1 \end{array} \quad \begin{array}{r} 0010 \\ \overline{\quad} \\ \downarrow \end{array} \quad \rightarrow \quad 2 \quad 5 // 2 = 2$$

$$0101 \gg 2 \quad \begin{array}{r} 0001 \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ 128 \quad 64 \quad 32 \quad 16 \quad 8 \quad 4 \quad 2 \quad 1 \end{array} \quad \rightarrow \quad 1 \quad 2 // 2 = 1$$

$$0010 \quad 0010 \gg 1 \quad \begin{array}{r} 0010 \quad 0010 \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \end{array}$$

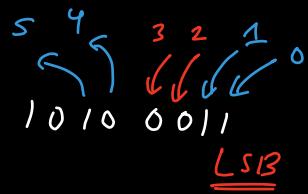
$$\begin{array}{r} \left(\frac{32}{2}\right) + \left(\frac{2}{2}\right) = \frac{34}{2} \\ \downarrow \quad \downarrow \\ \frac{16}{2} + \frac{1}{2} = \frac{17}{2} \quad \frac{37}{2} \\ \downarrow \quad \downarrow \\ 8 + 0 = 8. \quad \frac{34}{2} = \frac{34}{2^2} \end{array}$$

Quiz - 4

$$a \gg b = a // (2^b)$$

$$16 \gg 5 = \frac{16}{2^5} = \frac{16}{32} = \boxed{0}$$

## 2. Bit Masking



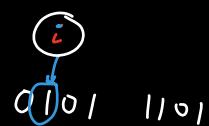
a. Given a number  $N$ , check if  $i$ th bit is set in the number or not.

$$\begin{cases} 0 \& 1 = 0 \\ 1 \& 1 = 1 \end{cases}$$

the number or not.

Idea

$N:$



$$i=6$$

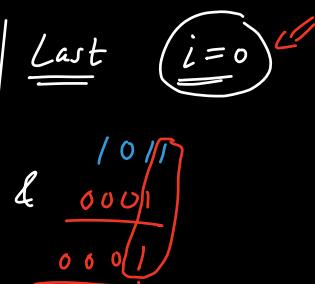
$$a \& 0 = 0$$

res:

$$\begin{array}{c} \& 0100 \quad 0000 \\ \hline & 0100 \quad 0000 \end{array} \Rightarrow \underline{\text{Bitmask}}$$

- filter relevant data
- get rid of junk.

$$0000 \quad 0001 \ll 6 = 0100 \quad 0000$$



$$1 \ll 0 = 1 \star$$

$$1 \ll 1 = 2$$

def test\_bit (number, index):

# Todo. Ret. True/False.

$$\text{bitmask} = (1 \ll \text{index})$$

$$\text{res} = \text{number} \& \underline{\text{bitmask}}$$

$$\text{return } \underline{\text{res}} > 0$$

b. Set Given Bit in number  $N$  (Update  $N$ )

$$\begin{array}{r} \text{7654} \quad 3210 \\ \downarrow \\ N = \begin{array}{c} 0101 \\ \underline{\quad\quad\quad} \\ \Rightarrow 0010 \quad \underline{0000} \end{array} \rightarrow (1 \ll 5) \\ \text{Set } i=5^{\text{th}} \text{ bit in } N. \end{array}$$

```
def set_bit(num, idx):  
    bitmask = (1 << idx)
```

# num should have  
idx bit set keeping  
all other bits same

num = num | mask

return num

$$\begin{array}{l} 1 \rightarrow 1 \\ 0 \rightarrow 1 \\ \hline 0|1 = 1 \\ 1|1 = 1 \end{array} \quad \left. \begin{array}{l} \text{In the above,} \\ N \& (1 \ll 5) = 0 \end{array} \right\} \text{Set} \\ \boxed{N = \begin{array}{c} 0111 \\ \uparrow \\ \text{---} \end{array}}$$

$1|0 = 1$  retains.  
 $0|0 = 0$

## Midway Recap

$$a \ll b = a * 2^b$$

### ① Shifting

$$\begin{array}{r} \text{a.} \\ \text{0010} \\ \hline \text{1000} \end{array} \quad \ll 2 \quad = \quad 8$$

$$\begin{aligned} 2 \ll 2 &= 2 * 2^2 \\ &= 2^3 = 8. \end{aligned}$$

$$\text{b.} \quad \begin{array}{r} \boxed{1011} \\ \hline \text{11} \end{array} \quad \gg 3$$

$$a \gg b = a // 2^b$$

$$0001 = 1$$

$$11 // 2^3 = 11 // 8 = 1$$

### ② Bit Masking

✓ Check •  $N \& (1 \ll i)$

✓ Set •  $N | (1 \ll i)$

$$N = 1101$$

$$i = 3$$

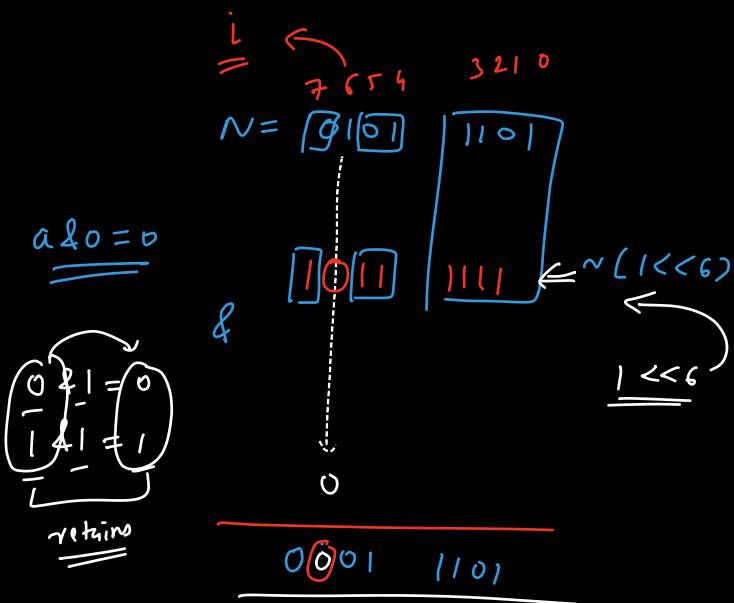
$$\begin{array}{ccc} 0001 & 0 & | \\ \swarrow & & \\ 0010 & 1 & | 0 \\ \swarrow & & \\ 0100 & 2 & | 00 \\ \swarrow & & \\ 1000 & 3 & | 000 \end{array}$$

$$1 \ll i = \underline{\underline{1000}}$$

| \_\_\_\_\_  
i zeros.

$$\begin{array}{r} 0101 \\ \hline 1000 \\ \hline 1000 \end{array} \Rightarrow 8 > 0$$

Unset.  
 C. Clear i<sup>th</sup> Bit



$(\rightarrow 0)$   
 $\text{set} \rightarrow \text{unset}$   
 $\text{unset} \rightarrow \text{unset}$   
 $0 \rightarrow 0$

If 1 then  
 should become zero.

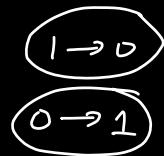
def clear\_bit(num, idx):  
 $\text{bitmask} = \sim(1 \ll \text{idx})$   
 return num & bitmask

d. Toggle Bit

$i=5$

$\begin{array}{r} 0101 \\ \text{bitmask} \\ \hline 0100 \end{array}$	$\begin{array}{r} 1101 \\ \text{num} \\ \hline 0000 \end{array}$	$1 \ll i$
--	--	-----------

---



mask bit is 1.

$$\begin{array}{r} 1 \wedge 1 = 1 \\ 0 \wedge 1 = 0 \end{array}$$

$$\begin{array}{r} 011 = 1 \\ 111 = 0 \end{array} \quad \text{XOR}$$

$i=6$

$\begin{array}{r} 0101 \\ \text{bitmask} \\ \hline 0000 \end{array}$	$\begin{array}{r} 1101 \\ \text{num} \\ \hline 0000 \end{array}$	$1 \ll 6$
--	--	-----------

---

so obtain.

$$\begin{array}{r} 1 \wedge 0 = 0 \\ 0 \wedge 0 = 0 \end{array}$$

```
def toggle_bit(num, idx):
```

```
    bitmask = 1 << idx
```

```
    return num ^ bitmask,
```

### 3. Problems

#### a. Single Set Bit

Check if the given number has 1 set bit!

exactly

Brute Force

$$N = 15$$

Count the set bits

2	15	num
2	7	1
2	3	1
2	1	1
0	0	1

Decimal  $\rightarrow$  Binary.

$$\begin{array}{r}
 1111 \\
 \uparrow \uparrow \uparrow \uparrow \\
 15 / .2 \\
 = 1 \\
 15 / 2 = 7 \\
 7 / .2 = 1 \\
 7 / 2 = 3 \\
 3 / .2 = 1 \\
 3 / 2 = 1 \\
 1 / .2 = 1
 \end{array}$$

$$\begin{aligned}
 2^0 + 2^1 + 2^2 + 2^3 &= 1 + 2 + 4 + 8 \\
 &= 15.
 \end{aligned}$$

def single\_set\_bit(N):

```

cnt = 0
while (N > 0):
    C = N // 2
    N = N / 2  $\Rightarrow$  N = N >> 1
    if C == 1:
        cnt += 1
    if cnt == 1:
        return True
    else:
        return False.
  
```

2	<u><u>N = 16</u></u>
2	8
2	4
2	2
2	1
0	0

$$\begin{array}{r}
 \text{num} \\
 \uparrow \uparrow \uparrow \uparrow \\
 0 \\
 0 \\
 0 \\
 1 \\
 \boxed{1}
 \end{array}
 \quad \boxed{10000}$$

Cnt = 1

TC: O(\log\_2 N)

$$15 \rightarrow x \quad 1111$$

$$16 \rightarrow \sim 0000 = 2^4$$

$$4 \quad 000 = 2^2$$

$$5 \quad 101$$

$$6 \quad 110$$

$$7 \quad 111$$

~~$\sqrt{8} \neq 2^3$~~ 

$$8 \quad 000 = 2^3$$

$$9 \quad 1001$$

Powers of 2 have  
a single set bit

$$6 \quad 1 \quad 4 \quad 3 \quad 2 \quad 1 \quad 0$$

$$0 \quad 0 \quad 0 \quad | 000 = 2^3$$

$$0 \quad | 1 \quad 0 \quad 000 = 2^5$$

$$0 \quad 0 \quad 1 \quad 1000 = \cancel{\int 2^3 + 2^4}$$

$$1 << i$$

$$\sim(1 << i)$$

Bitmask

$$\begin{array}{r} a \\ - 1 \\ \hline a-1 = \end{array}$$

a      0100      0000      0000  
      | 2      2      2      2  
      0100      0000      0001  
                 borrow.

0000      0000      0001  
                 2      2      2  
                 0001      1111

least significant set bit

eliminate

$$\begin{array}{r} 0000 \\ - 0000 \\ \hline 0000 \end{array}$$

$$\begin{array}{r} 0000 \\ - 0001 \\ \hline 0001 \end{array}$$

$$\begin{array}{r} 0000 \\ - 0011 \\ \hline 0011 \end{array}$$

$$\underline{\underline{1 - 1 = 0}}$$

$$\begin{array}{r} 0101 \quad 0111 \\ - 0000 \quad 0001 \\ \hline 0101 \quad 0110 \end{array}$$

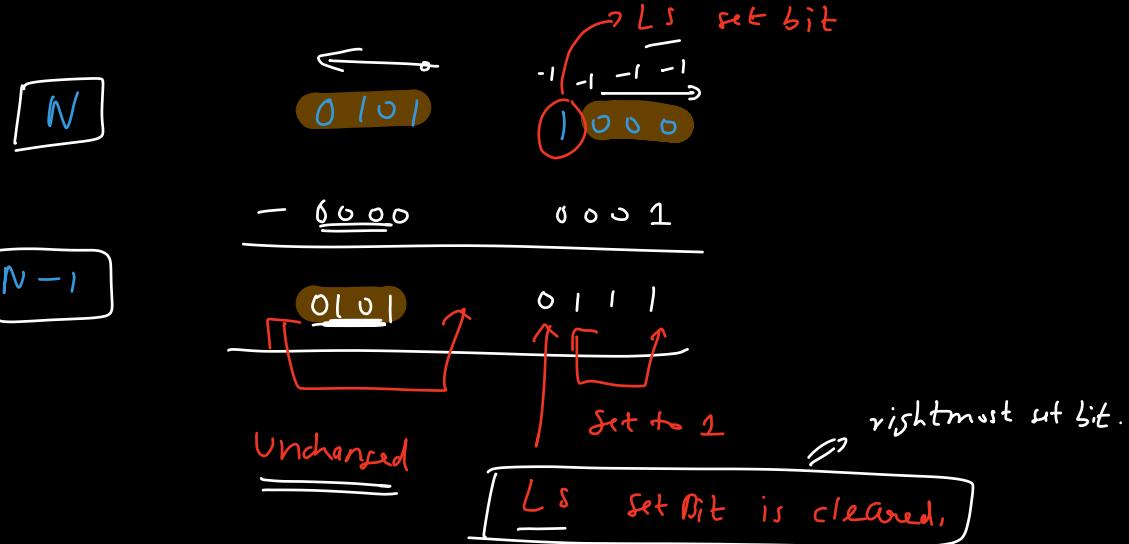
$$\begin{array}{r} 1110 \quad 1010 \\ - 0000 \quad 0001 \\ \hline 1110 \quad 1001 \end{array}$$

$$\begin{array}{r} 5 \quad 0101 \\ 1 \quad 0001 \\ \hline 4 \quad 0100 \end{array}$$

$$\begin{array}{r} 6 \quad 110 \\ 1 \quad 001 \\ \hline 5 \quad 101 \end{array}$$

$$\begin{array}{r} -1 -1 -1 \\ 1978 \\ - 999 \\ \hline 0979 \end{array}$$

↑  
maintained.

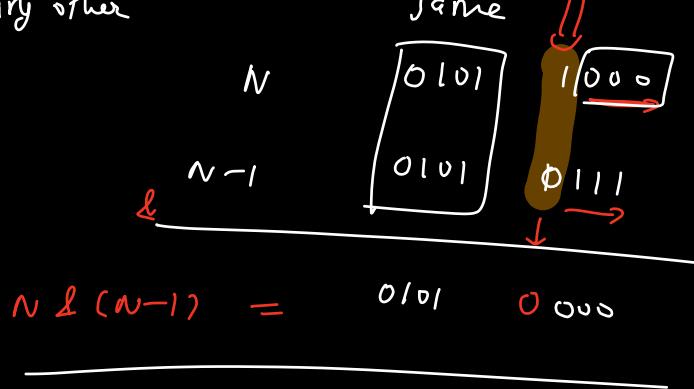


Clear Ls set Bit of  $N$

Without affecting other  
bits.

$$a \& a = a$$

$$a \& 0 = 0$$



We cleared the rightmost set bit.

Original prob      Single set bit

$$N = \begin{array}{r} - \underline{\text{01}} \text{ 00} \quad \underline{\text{0000}} \\ \underline{\text{00}} \end{array}$$

$$N-1 = \begin{array}{r} - \underline{\text{00}} \text{ 11} \quad \underline{\text{1111}} \\ \underline{\text{00}} \end{array}$$

$$N \& (N-1) = \begin{array}{r} - \underline{\text{00}} \text{ 00} \quad \underline{\text{0000}} \\ \underline{\text{00}} \end{array} \Rightarrow \boxed{0}$$

$N \& (N-1) \Rightarrow$  clear the rightmost set bit.  
(LS)

$$\begin{array}{l} N = 10100000 \\ N-1 = 10011111 \\ \hline N \& (N-1) = 10000000 \end{array}$$

If  $> 1$  set bits in  $N$   
 $\Rightarrow \underline{\underline{N \& (N-1) = 0}}$

Conclude : If 1 set bit  
 $\Rightarrow \underline{\underline{N \& (N-1) = 0}}$  ✓ except  $N=0$

Can we say if  $N \& (N-1) = 0 \Rightarrow N$  is power of 2

Edge Case       $N=0$        $0 \& (0-1)$   
                        ||  
                        X       $0 \& 1 = 0$

---

def single\_set\_bit(num):  
    if num == 0:  
        return False  
    return  $num \& (num-1) == 0$

$O(\log n)$   
to  
 $O(1)$

6) Can we also optimize count set bits using this logic?

$N = N \& (N-1) \Rightarrow$  clear the rightmost set bit

```
def count_set_bits(N):  
    cnt = 0  
    while(N > 0):  
        cnt += 1  
        N = N & (N-1)
```

TC:  $O(\text{num of set bits})$

$N = 1010\ 0000$   
 $N-1 = 1001\ 1111$   
 $N \& (N-1) = 1000\ 0000$

$N^1$

$N^1 - 1 = 0111\ 1111$   
 $\underline{0000\ 0000}$

$O[1]0[11]0$   
↓  
 $01101100$   
↓  
 $0110\underline{1}000$   
↓  
 $011\underline{0}00000$   
↓  
 $0\underline{1}000000$   
↓  
Stop.  $\leftarrow 0000\cdots$

1  
2  
3  
4  
5

$\Rightarrow$  Doubt

$2^3 \quad 19 \quad 15 \quad 11 \quad 7 \quad 321^0$   
1 0

$$= \underline{\underline{2^{23}}}$$

$O(\log n) \rightarrow 23 \text{ steps.}$

$O(\text{count set bits}) \rightarrow \underline{\underline{1 \text{ step.}}} \checkmark$