

Today's Content

- ① Negative numbers Representation
- ② Left Shift & Right Shift
- ③ 5 Interesting / Simple Bitwise problems
- ④ 1 Advanced Bitwise problem

8bit

10 : 0 0 0 0 1 0 1 0

-10 : | 0 0 0 1 0 1 0 X $\Rightarrow -128 + 8 = -118$

2's complement = $-a = \sim a + 1$ } Proof in Today's Doubts

~ 10 : 1 1 1 1 0 1 0 1 } $\sim a$ is Nothing but
is Complement

+1 : 0 0 0 0 0 0 0 1

$\frac{1.1.1.1.0.1.1.0}{2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0}$ Decimal Representation

$$\Rightarrow -128 + 64 + 32 + 16 + 4 + 2 = -128 + 118 = -10$$

$$23 : \underline{0} \quad \underline{0} \quad \underline{0} \quad \underline{1} \quad \underline{0} \quad \underline{1} \quad \underline{1} \quad \underline{1}$$

$$v_{23} : \quad 1 \quad | \quad | \quad 0 \quad | \quad 0 \quad 0 \quad 0 \quad]$$

$$+1 \quad 0 \quad |$$

1 1 1 0 1 0 0 1

$$-128 + 64 + 32 + 8 + 1 \Rightarrow -128 + 105 = -23$$

30 : 0 0 0 | 1 1 1 1 0

$$\sim 30 : \quad | \quad | \quad | \quad 0 \quad 0 \quad 0 \quad 0 \quad |$$

$$11 : \quad \underline{0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1}$$

1 1 1 0 0 0 1 0

$$-128 \quad 64 \quad 32 \quad 2 \quad + \quad -128 + 18 = -80$$

Left Shift

8 bit Number

$$a = 10$$

$-2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$

$\underline{0} \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0$

$$: 10$$

$$a \ll 1$$

$\underline{0} \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0$

$$: 2^4 + 2^2 = 20 : 10 \times 2$$

$$a \ll 2$$

$\underline{0} \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0$

$$: 2^5 + 2^3 = 40 : 10 \times 2^2$$

$$a \ll 3$$

$\underline{0} \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0$

$$: 2^6 + 2^4 = 80 : 10 \times 2^3$$

$$a \ll 4$$

$\underline{1} \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0$

$$: -2^7 + 2^5 = -128 + 32$$

$$= -96$$

$$a \ll 4 \rightarrow 10 \times 2^4 \rightarrow 160$$

$$a = 10$$

overflow & bit cannot
store 160

$$\rightarrow \text{In 8 bit Number: } [-128, 127]$$

$$a = 127 = \underline{0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1}$$

$$(a+1) + 1 = \underline{0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1}$$

$$127 + 1 = 128$$

$$a \ll n = a \times 2^n$$

$$a = 15, 15 \ll 2$$

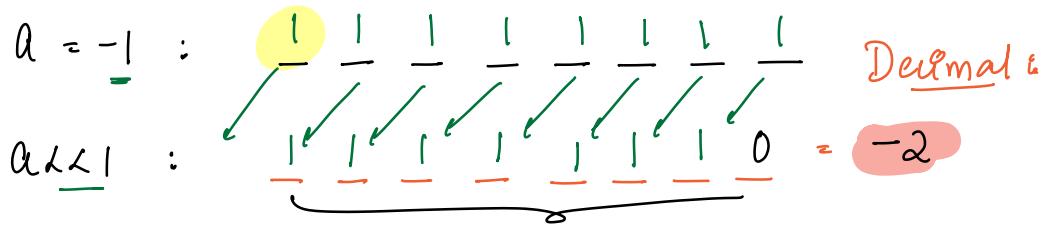
$\underline{0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1}$

$\underline{0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0}$

$$32 \ 16 \ 8 \ 4$$

$$\Rightarrow 60$$

$$\Rightarrow 15 \times 2^2$$



$$\underline{-1} \ll 1 = \underline{-1} * 2^1 = \underline{-2} \rightarrow \text{8 bit Number} \rightarrow \text{No Overflow}$$

Right Shift

8 bit Number

$$a = \underline{40} : \begin{array}{cccccccc} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{array} \xrightarrow{\text{LSB discarded}} \text{Dec} \rightarrow 40 :$$

$$a \gg 1 : \begin{array}{cccccccc} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{array} = 2^4 + 2^2 \Rightarrow 20 : 40/2^1$$

$$a \gg 2 : \begin{array}{cccccccc} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{array} = 2^3 + 2^1 \Rightarrow 10 : 40/2^2$$

$$a \gg 3 : \begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{array} : 40/2^3 \Rightarrow 5$$

$$a \gg 4 : \begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} : 40/16 \Rightarrow 2$$

↳ Integer Division

$$a = \underline{50} : a \gg 2 : \underline{50}/2^2 \Rightarrow \underline{50}/4 \Rightarrow 12 \quad \left. \begin{array}{l} \text{+ve positive Numbers} \\ a \gg N : a/2^N \end{array} \right\}$$

$$a = \underline{10} : a \gg 4 : \underline{10}/2^4 \Rightarrow \underline{10}/16 \Rightarrow \underline{0}$$

TODO: Right Shift for Negative Numbers

$a \gg N : a/2^N$

↳ No overflow

Problems

→ Check if i^{th} bit of N is Set or Unset

$\{ \ 1 \leq N \leq 10^9, \ 0 \leq i \leq 30 \} \rightarrow i \text{ is bit position}$

bool CheckBit (int N, i) {

$N = 21, i = 2 \rightarrow \text{Set}$

$\begin{array}{r} 4 \ 3 \ 2 \ 1 \ 0 \\ | \quad | \quad | \quad | \quad | \\ 1 \ 0 \ 1 \ 0 \ 1 \end{array}$

$N = 25, i = 2 \rightarrow \text{unset}$

$\begin{array}{r} 4 \ 3 \ 2 \ 1 \ 0 \\ | \quad | \quad | \quad | \quad | \\ 1 \ 1 \ 0 \ 0 \ 1 \end{array}$

Example

$N = 106 \therefore 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0$

$i = 0 \therefore (N \& 1) == 1, \text{ set else unset}$

$N = 106 \therefore 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0$

$i = 1 \therefore (N \gg 1) \& 1 == 1$

$N = 24, i = 2 \rightarrow \text{unset} \rightarrow \text{False}$

$\begin{array}{r} 4 \ 3 \ 2 \ 1 \ 0 \\ | \quad | \quad | \quad | \quad | \\ 1 \ 1 \ 0 \ 0 \ 0 \end{array}$

$N = 14, i = 3$

$\begin{array}{r} 4 \ 3 \ 2 \ 1 \ 0 \\ | \quad | \quad | \quad | \quad | \\ 1 \ 1 \ 1 \ 0 \end{array}$

$\rightarrow \text{Set} \rightarrow \text{True}$

$N = 106$

$\begin{array}{r} 7 \ 6 \ 5 \ 4 \ 3 \ 2 \ 1 \ 0 \\ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \end{array}$

$i = 0 \therefore (N \& 1) \neq 0$

$\begin{array}{r} 7 \ 6 \ 5 \ 4 \ 3 \ 2 \ 1 \ 0 \\ \rightarrow 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \\ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \end{array}$

$N \& (\underline{1} \& \underline{1}) == 0$

2nd bit is unset

$N = 106 \Rightarrow 0\ 1\ 1\ 0\ \underline{1}\ 0\ 1\ 0$
 i = 3: $(N \gg 3) \& 1 = 1$

```

bool checkBit(N, i){
    if ((N >> i) & 1 == 1) {
        return True
    }
    else {
        return False
    }
    return (N >> i) & 1 == 1
}
    
```

```

bool checkBit(N, i){
    if (N & (1 << i) != 0) {
        return True
    }
    else {
        return False
    }
}
    
```

LO: ZEROFILLS

Note: All bitwise operations only on Integer Data types

$N = 106 \Rightarrow 0\ 1\ 1\ 0\ \underline{1}\ 0\ 1\ 0$
 $i = 3$
 $(1 << 3) = 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0$

$N \& (1 << 3) != 0$

3rd bit is Set

Double

$a = 10$

$a + 10 \rightarrow 20$

$\text{print}(a) \rightarrow 10$

$a = 10$

$a \underline{\underline{\gg}} 1 \rightarrow 5$

$\text{print}(a) \rightarrow 10$

Given integer N calculate No: of Set bits are present

$$\text{Count Set Bits}(N) : \Theta \leq N \leq \log^2 N \rightarrow \text{Int} : 32 \text{ bits}$$

$$N = 25 : \begin{array}{cccccc} 1 & 1 & 0 & 0 & 1 \end{array} : 8 \text{ bits} \quad \xrightarrow{\text{int}} \quad \begin{array}{c} [3] \\ 0 \end{array}, \underbrace{30 \dots 0}_{0}$$

$N = 40 : 101000 : 2 \text{ bits}$

$N=35$: 100011 : 3 bits

countSet Bits(N) {

→

 cnt = 0

 q = 0; q <= 30; q++) {

 if (checkBit(N, q)) {

 cnt += 1;

 }

 }

 return cnt;

N = 27: 11011 —
N = 27, ^mi code: 31 iterations

$N = 109$

$N >= 1 : \quad | \quad | \quad 0 \quad | \quad | \quad 0 \quad | \quad |$

$N >= 1 : \quad | \quad | \quad 0 \quad | \quad | \quad 0 \quad | \quad |$

while ($N >= 1$) {

$(N // 1) == 1$

// cut++

$N = N // 2$

r chum cut; $\Rightarrow (\log_2 N)$

$N > 2^7, 2^{\underline{\text{nd}}} \text{ cond: } 5 \text{ iterations}$

→ Prerequisites

$$\begin{array}{r}
 & 4 & 3 & 2 & 1 & 0 \\
 \overline{\underline{11}} & : & 0 & 1 & 0 & 11 \\
 \overline{\underline{25}} & : & & 1 & 0 & 0 & 1 \\
 \hline
 \overline{\underline{1125}} & : & \textcolor{red}{1} & 0 & 0 & 1 & 0
 \end{array}$$

At 4th bit both are diff

At 1st bit both are diff

$$\overline{\underline{a^1b}} = 0 \quad a \underline{==} b$$

Given two integers N , find least significant set bit position.

$$N = 44 : \begin{array}{cccccc} 5 & 4 & 3 & \textcolor{green}{2} & 1 & 0 \\ 1 & 0 & 1 & \textcolor{blue}{1} & 0 & 0 \end{array}$$

$$q = 0; \alpha = 30; q++ \{$$

if (checkBit(N, q)) {

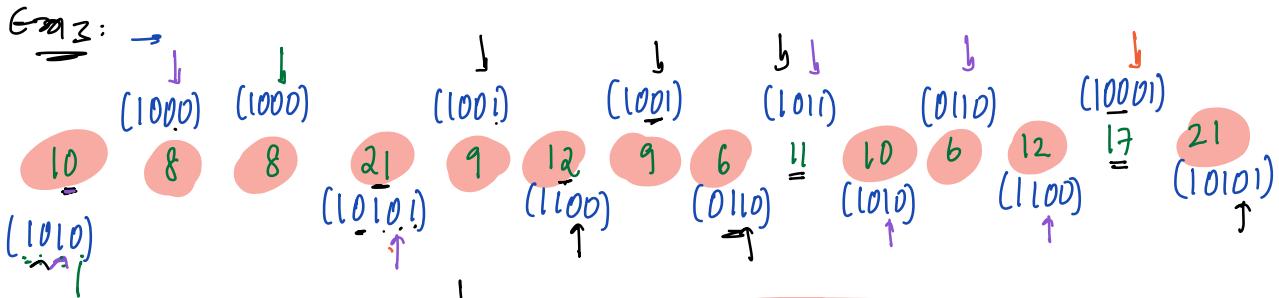
 ans = q; break

} }

Given N array elements, every element repeats twice except 2 unique elements, find the 2 unique elements

$$\underline{\underline{\text{Ex1:}}} \quad 3 \quad 4 \quad 5 \quad 4 \quad 7 \quad 5 \quad : \quad \text{3 7} \quad \text{output.}$$

$$\underline{\underline{\text{Ex2:}}} \quad 4 \quad 9 \quad 9 \quad 8 \quad 5 \quad 4 \quad 8$$



(Step 1) nr of all elements

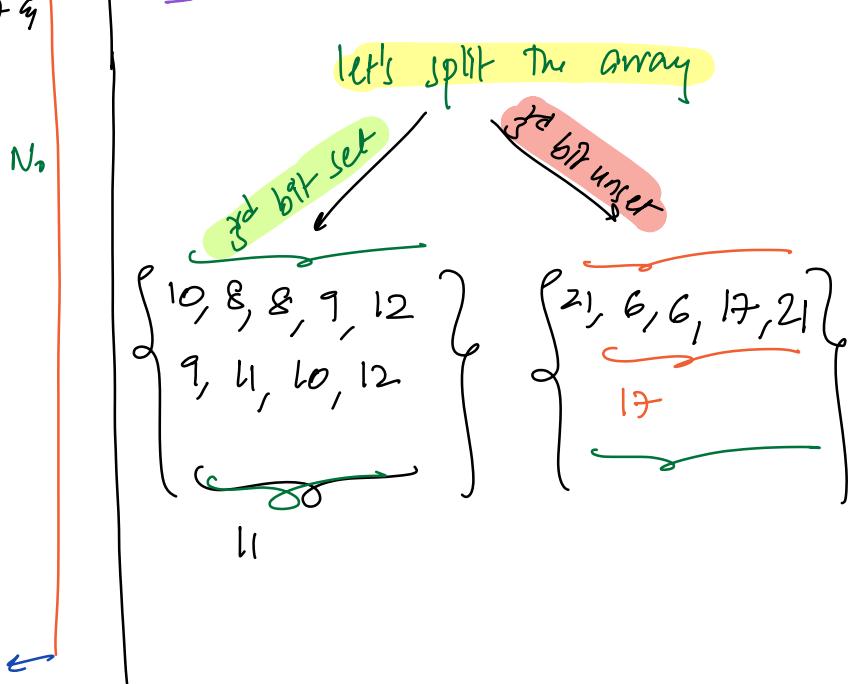
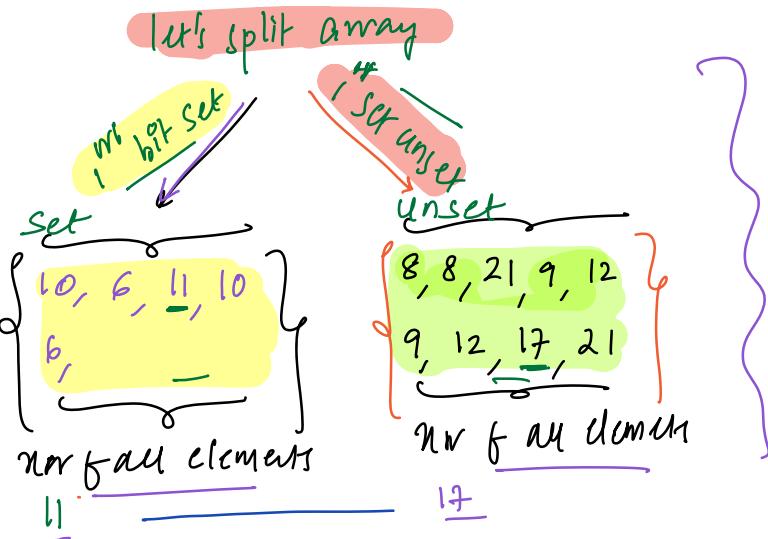
→ nr of unique elements

$$\begin{array}{r}
 4 \ 3 \ 2 \ 1 \ 0 \\
 \rightarrow 17 : 1 \ 0 \ 0 \ 1 \ 0 \ 1 \\
 11 : 0 \ 1 \ 0 \ 1 \ 1 \\
 = \\
 17 \ 11 : 1 \ 0 \ 0 \ 1 \ 0
 \end{array}$$

At bit pos=1, both 17 & 11 are different

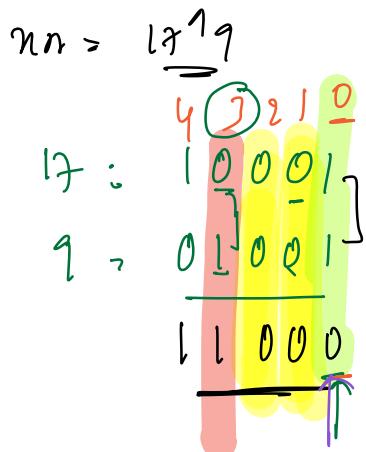
Can we split on 0th bit? No

1st bit: YES
2nd bit: NO
3rd bit: YES
4th bit: YES



Pseudo Code

- 1) val = nn for all elements
- 2) pos = Get any set pos in val
- 3) Split based whether a number is set a or not at pos^m



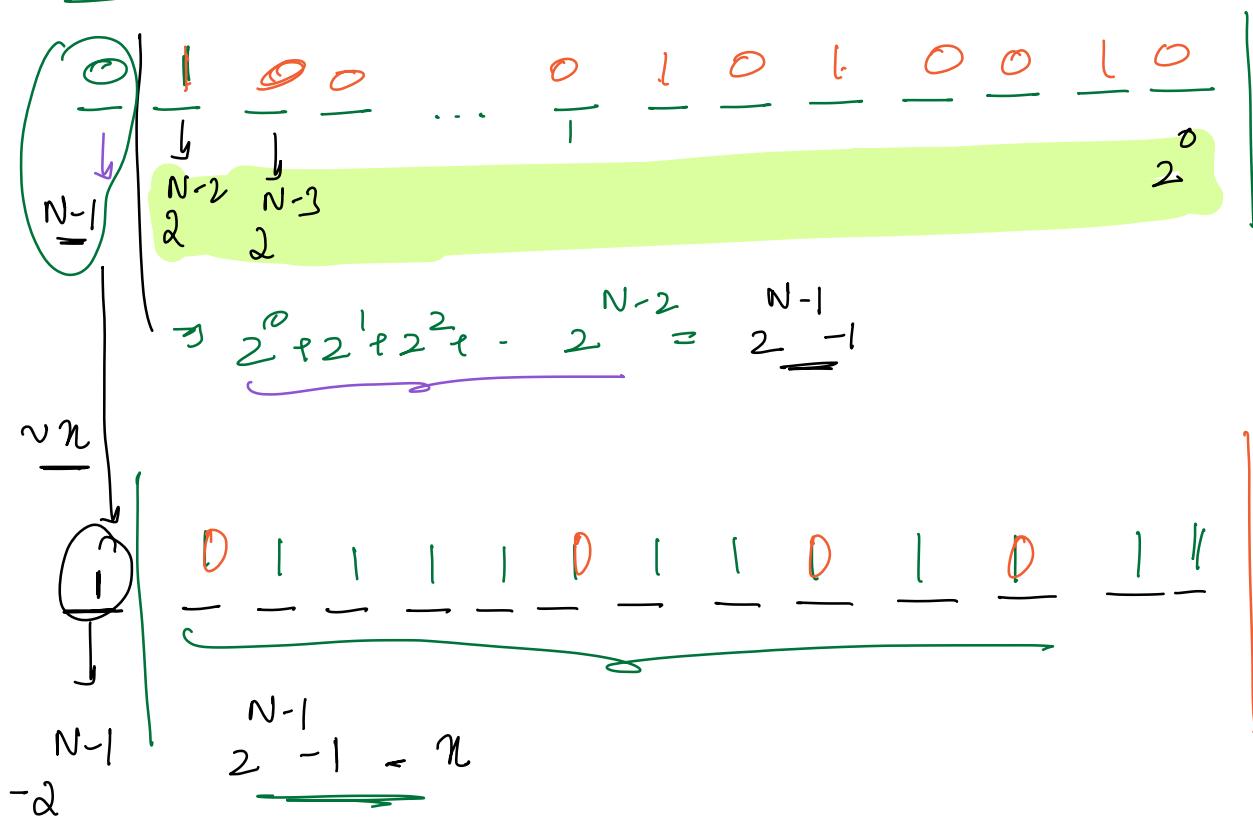
```

→ val = 0
p = 0; q < N; p++ {
    val = val ^ arr[i]
    pos = 0
    p = 0; q = 3; p++ {
        if (checkBit(val, q)) {
            pos = i; break;
        }
    }
    set = 0, unset = 0
    p = 0; q < N; p++ {
        if (checkBit(arr[q], pos)) {
            set = set ^ arr[q];
        }
        else {
            unset = unset ^ arr[q];
        }
    }
    print(set, unset)
}

```

2nd Complement

$n > 0$, N bit {Doubt})

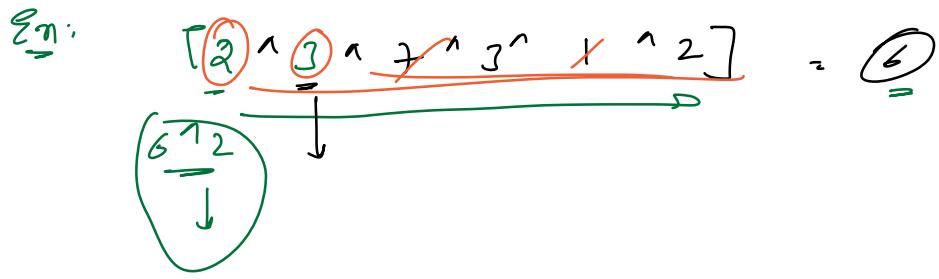


$$\Rightarrow -2^{N-1} + 2^{N-1} - 1 = n$$

$$\sim n = -1 - n$$

$$[-n = \sim n + 1]$$

$$\text{if } n = \underline{\sim n + 1}$$



110
010
100

$\rightarrow P = O; Q \times N; R_{11}; \{$

$g = i; j \times N; g_{11}; \{$

if $(\text{car}[i] \wedge \text{ar}[j] = = 0) \{$

}

}

$\begin{array}{r} 5 \\ + 4 \\ \hline 9 \end{array}$