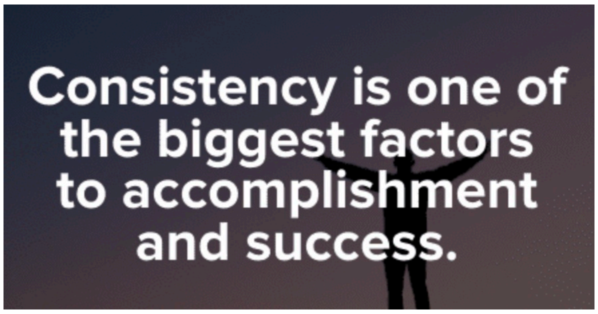
Today's Quote



Consistency is one of the biggest factors to accomplishment and success.

Content → Basics of Bit Manipulation.

Number System

- **Decimal number system** → {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} Base - 10.

$743 \rightarrow (7 \times 100) + (4 \times 10) + 3$

$\rightarrow (7 \times 10^2) + (4 \times 10^1) + (3 \times 10^0)$

$2564 \rightarrow (2 \times 10^3) + (5 \times 10^2) + (6 \times 10^1) + (4 \times 10^0)$

- **Binary Number System** → {0, 1} Base-2

$110 \Rightarrow (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) = 6$

$\underset{3}{1}\,\underset{2}{0}\,\underset{1}{1}\,\underset{0}{1} \Rightarrow (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) = 11$

0	10	20	30		90	100
1	11	|	|		91	
2	12	|	|		92	
3	1	|	|	——	|	
4	|	|	|		|	
5	|	|	|		|	
|	|	|	|		|	
9	19	29	39		99	

---

0 → 0

1 → 1

10 → 2

11 → 3

100 → 4

101 → 5

110 → 6

111 → 7

1 000 → 8

⋮

Binary To Decimal

Eg : 1 0 1 1 0

$4 \; 3 \; 2 \; 1 \; 0$

$0 * 2^0 = 0$

$1 \times 2^1 = 2$

$1 \times 2^2 = 4$

$0 \times 2^3 = 0$

$1 \times 2^4 = 16$

22

$\therefore (10110)_2 = (22)_{10}$

n = 10110

$\downarrow 10$

1011 , r → 0

$\downarrow 10$

101 , r → 1

$\downarrow 10$

10 , r → 1

$\downarrow 10$

1 , r → 0

$\downarrow 10$

0 , r → 1

Eg : 1 0 1 1 0 1 0

$6 \; 5 \; 4 \; 3 \; 2 \; 1 \; 0$

0×2^0

1×2^1

0×2^2

1×2^3

1×2^4

0×2^5

1×2^6

90

$\therefore (1011010)_2 = (90)_{10}$

pseudo-code.

```
int btod ( n ) {
    ans = 0
    power = 1    // 1 = 2^0
    while( n > 0) {
        r = n % 10
        n = n /10
        ans += (r * power)
        power = pow*2 ;
    }
    return ans
}
```

T.C → O(no. of digits) → $O(\log_{10} N)$

S.C → O(1)

Decimal to Binary

$(20)_{10} = (10100)_2$

2	20		
2	10	→	0
2	5	→	0
2	2	→	1
2	1	→	0
	0	→	1

$\underset{4}{1}\ \underset{3}{0}\ \underset{2}{1}\ \underset{1}{0}\ \underset{0}{0}$

$\downarrow \quad \downarrow$

$2^4 + 2^2 = 16+4 = \underline{\underline{20}}.$

$\therefore \ (45)_{10} = (\underset{5}{1}\underset{4}{0}\underset{3}{1}\underset{2}{1}\underset{1}{0}\underset{0}{1})_2$

$\Downarrow$

$2^0 + 2^2 + 2^3 + 2^5$

$= \underline{\underline{45}}$

2	45		
2	22	→	1
2	11	→	0
2	5	→	1
2	2	→	1
2	1	→	0
	0	→	1

[# todo → decimal to binary.]

Addition.

$$\begin{bmatrix} 0+0 \rightarrow 0 \\ 0+1 \rightarrow 1 \\ 1+0 \rightarrow 1 \\ 1+1 \rightarrow 10 \end{bmatrix}$$

$$\begin{bmatrix} 1+1+1 \rightarrow 11 \end{bmatrix}$$

①

$$\begin{array}{rcccccl} & 1 & 1 & 1 & & & \\ & & 1 & 0 & 1 & \rightarrow & (5) \\ + & & 0 & 1 & 1 & \rightarrow & (3) \\ \hline & 1 & 0 & 0 & 0 & \rightarrow & (8) \end{array}$$

②

$$\begin{array}{rcccccccl} & 1 & 1 & 1 & 1 & & & & \\ & & 1 & 0 & 1 & 1 & 0 & \rightarrow & 22 \\ + & & 0 & 1 & 1 & 1 & 1 & \rightarrow & 15 \\ \hline & 1 & 0 & 0 & 1 & 0 & 1 & - & \underline{\underline{37}} \end{array}$$

Bitwise Operators

int → 4 Bytes → [32 bits].

$(42)_{10} = (101010)_2$

$$42 \rightarrow \left[\begin{array}{cccccccccccc} 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 1 & 0 & 1 & 0 \\ 31 & 30 & 29 & 28 & 27 & & 5 & 4 & 3 & 2 & 1 & 0 \end{array}\right]$$

$$(0\times2^{31}) + (0\times2^{30}) + (0\times2^{29}) + (0\times2^{28}) + \cdots (1\times2^{5}) + (0\times2^{4}) + (1\times2^{3}) + (0\times2^{2}) + (1\times2^{1}) + (0\times2^{0})$$

$$= 42.$$

operators → { AND, OR, NOT, XOR, [left shift, right shift] }

AND,	OR,	NOT,	XOR,	left shift,	right shift
&	|	! | ~	^	<<	>>

A	B	A & B	A | B	A ^ B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

1 → set

0 → unset

↳ { same same puppy shame }

Bitwise operations on numbers

① $\boxed{5 \,\&\, 6 = 4}$

5 → 1 0 1

6 → 1 1 0 &

4 → 1 0 0

② A = 20

B = 45

A | B = ?

A → 0 1 0 1 0 0

B → 1 0 1 1 0 1 |

1 1 1 1 0 1

5 4 3 2 1 0

⇓

$2^5+2^4+2^3+2^2+2^0 = \underline{\underline{61}}.$

③ A = 20

B = 45

A ^ B

A → 0 1 0 1 0 0 ^

B → 1 0 1 1 0 1

A^B → 1 1 1 0 0 1

5 4 3 2 1 0

$2^5+2^4+2^3+2^0 = \underline{\underline{57}}.$

Properties

① A & 1

$$(A=10) \Rightarrow \begin{array}{rcccc} & 1 & 0 & 1 & 0 \\ \& & 0 & 0 & 0 & 1 \\ \hline & 0 & 0 & 0 & 0 \end{array} \qquad (A=9) \Rightarrow \begin{array}{rcccc} & 1 & 0 & 0 & 1 \\ \& & 0 & 0 & 0 & 1 \\ \hline & 0 & 0 & 0 & 1 \end{array}$$

Conclusion

$$A \,\&\, 1 - \begin{cases} = 0 & [\text{last bit of } A \text{ is } 0] \Rightarrow A \text{ is an even no.} \\ = 1 & [\text{last bit of } A \text{ is } 1] \Rightarrow A \text{ is an odd no.} \end{cases}$$

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

$$\Rightarrow \underbrace{\underbrace{(1\times2^5)}_{\text{Even}} + \underbrace{(0\times2^4)}_{\text{Even}} + \underbrace{(1\times2^3)}_{\text{Even}} + \underbrace{(1\times2^2)}_{\text{Even}} + \underbrace{(0\times2^1)}_{\text{Even}}}_{\text{Even}} + \underbrace{(1\times2^0)}_{\text{odd}}$$

$$\text{Even} + \text{odd} = \underline{\text{odd}}.$$

② $A \& 0 = 0$

$$\begin{array}{rcccc} A \rightarrow & 1 & 0 & 1 & 1 \\ 0 \rightarrow & 0 & 0 & 0 & 0 \\ \hline 0 \rightarrow & 0 & 0 & 0 & 0 \end{array} \;\&$$

③ $A \& A = A$

$$\begin{array}{rcccc} A \rightarrow & 1 & 0 & 1 & 1 \\ A \rightarrow & 1 & 0 & 1 & 1 \\ \hline A \rightarrow & 1 & 0 & 1 & 1 \end{array} \;\&$$

④ $A | 0 = A$

$$\begin{array}{rl} A \rightarrow & 1011 \\ 0 \rightarrow & 0000 \quad | \\ \hline & 1011 \\ \hline \end{array}$$

⑤ $A | A = A$

$$\begin{array}{rl} A \rightarrow & 1011 \\ A \rightarrow & 1011 \quad | \\ \hline A \rightarrow & 1011 \\ \hline \end{array}$$

⑥ $A \hat{} 0 = A$

$$\begin{array}{rl} A \rightarrow & 1011 \\ 0 \rightarrow & 0000 \quad \hat{} \\ \hline & 1011 \\ \hline \end{array}$$

**

⑦ $A \hat{} A = 0$

[xor of two same values = 0]

$$\begin{array}{rl} A \rightarrow & 101101 \quad \hat{} \\ A \rightarrow & 101101 \\ \hline & 000000 \\ \hline \end{array}$$

Commutative Property

$a \& b = b \& a$

$a | b = b | a$

$a \hat{} b = b \hat{} a$

$a \& b = b \& a$

$\underbrace{(a \& b)}_{x} \& c = c \& \underbrace{(b \& a)}_{x}$

$x \& c = c \& x$.

Associative Property.

$$(a \,\&\, b) \,\&\, c = a \,\&\, (b \,\&\, c)$$

$$(a \mid b) \mid c = a \mid (b \mid c)$$

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

$$\begin{bmatrix} a \,\&\, b \,\&\, c \\ a \,\&\, c \,\&\, b \\ b \,\&\, a \,\&\, c \\ b \,\&\, c \,\&\, a \\ c \,\&\, a \,\&\, b \\ c \,\&\, b \,\&\, a \end{bmatrix}$$

All are equal.

Q:) $\underline{a} \wedge \underline{b} \wedge \underline{d} \wedge \underline{b} \wedge \underline{c} \wedge \underline{c} \wedge \underline{a}$

$$= (\underline{\underline{a \wedge a}}) \wedge (\underline{\underline{b \wedge b}}) \wedge (\underline{\underline{c \wedge c}}) \wedge d$$

$$= \underline{0 \wedge 0 \wedge 0} \wedge d \quad = 0 \wedge d \;=\; \underline{\underline{d}}.$$

Q.) Given an integer arr[N]. All the no's are present twice in the array except only one. Find the no. which is only present once.

arr →	[6	3	7	3	9	9	6]
	0	1	2	3	4	5	6

⇒ $\underline{6} \wedge \underline{3} \wedge 7 \wedge \underline{3} \wedge \underline{9} \wedge \underline{9} \wedge \underline{6}$

⇒ $\underline{6 \wedge 6} \wedge \underline{3 \wedge 3} \wedge \underline{9 \wedge 9} \wedge 7$ ⇒ $0 \wedge 7 = \underline{\underline{7}}$

Pseudo-code.

```
int fun( arr, N) {
    ans = 0
    for( i = 0 ; i < N ; i++) {
        ans = ans ^ arr[i]
    }
    return ans
}
```

$$\boxed{\begin{array}{l} T.C \rightarrow O(N) \\ S.C \rightarrow O(1) \end{array}}$$

arr →	[6,	7,	7,	9,	6]
	0	1	2	3	4

[ans = 0]

ans = 0^6 = 6

ans = 6^7 = 1

ans = 1^7 = 6

ans = 6^9 = 15

ans = 15^6 = 9

6 → 0110

7 → 0111

9 → 1001

$$\begin{array}{r} 1111 \\ 0110 \\ \hline 1001 \\ \hline \end{array}$$

Left - shift.

int → 4 bytes → 32 bits

// Assumption → int → 1 byte → 8 bits. [just for explanation]

$a=45$ 0 0 1 0 1 1 0 1   [45]

$a<<1$ discard   0 1 0 1 1 0 1 0 $[90] \Rightarrow [45 \times 2^1]$

$a<<2$ discard   1 0 1 1 0 1 0 0 $[180] \Rightarrow [45 \times 2^2]$

$a<<3$ discard   0 1 1 0 1 0 0 0 $[360] \Rightarrow [45 \times 2^3]$

7 6 5 4 3 2 1 0

$2^6 + 2^5 + 2^3 = 64+32+8 = \boxed{104}$ Why?

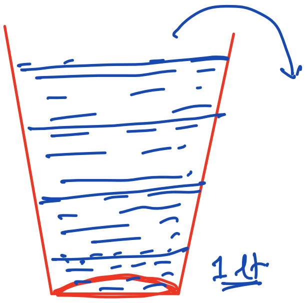
$$\boxed{\begin{aligned} a<<n &= a * 2^n \\ 1<<n &= 2^n \end{aligned}}$$

Max^m no. that can be stored in 8-bits =

1 1 1 1 1 1 1 1 → 255.

∴ 360 is too large to be stored in 8-bits.

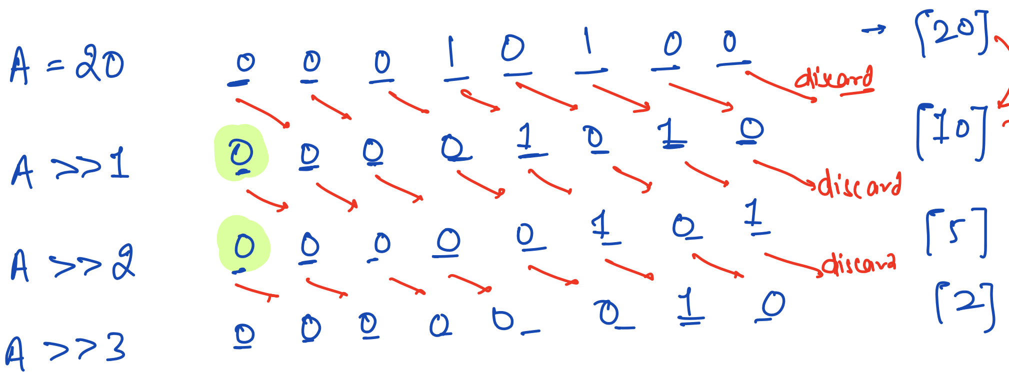
⇒ Overflow condition



Capacity = 10 lt.

$1\text{ lt} \xrightarrow{*2} 2\text{ lt.} \xrightarrow{*2} 4\text{ lt} \xrightarrow{*2} 8\text{ lt} \;]*2$

Right Shift Operator



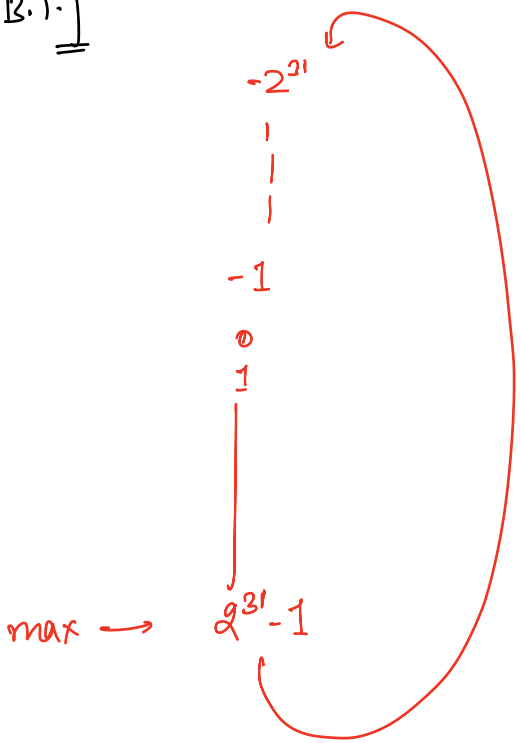
5/2 = 2.

$$\boxed{a >> n = \frac{a}{2^n}}$$

∴ No overflow condition.

Doubts:

[basics of B.T.]



$2^{31}-1 + 1 = -2^{31}$

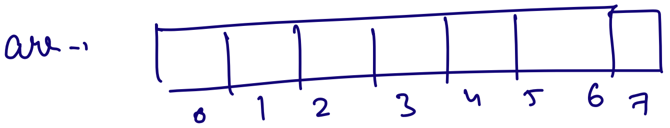
```
for(i=0; i< N; i++){
            // o(1)
            i++
            // o(1)
}
```

⇒

```
for(i=0; i<N; i+=2){
        //o(1)
        //o(1)
}
```

long ans.

$$(\underbrace{(\text{long})A.\text{get}(i)}_{\text{int}} * \underbrace{\overset{(\text{long})}{(N-i)}}_{\text{int}} * \underbrace{\overset{(\text{long})}{(i+1)}}_{\text{int}})$$



i -> 2, j - 5

```
for( K= i ; k <= j ; K++){
    print( arr[K]);

}
```

Psum | Carry Forward / S.W.