Reattempt 2 (Live till Sunday)

Expert Mock Interview.

floater (7 + 5) fixed

20 (8 + 12)

8 ↓ interactive

12 → Mock Interviews

Job ⇒ Skillsets ⇒ (Expert Mock Interview + Content)

(30 days for DSA & MI) ⇓ Revise or Cover Up Backlog

⇐ DSA —
Language —
LLD + SQL —
HLD —
Project —

DSA ⇒ PSP ≥ 80%.

60 ≤ PSP < 80

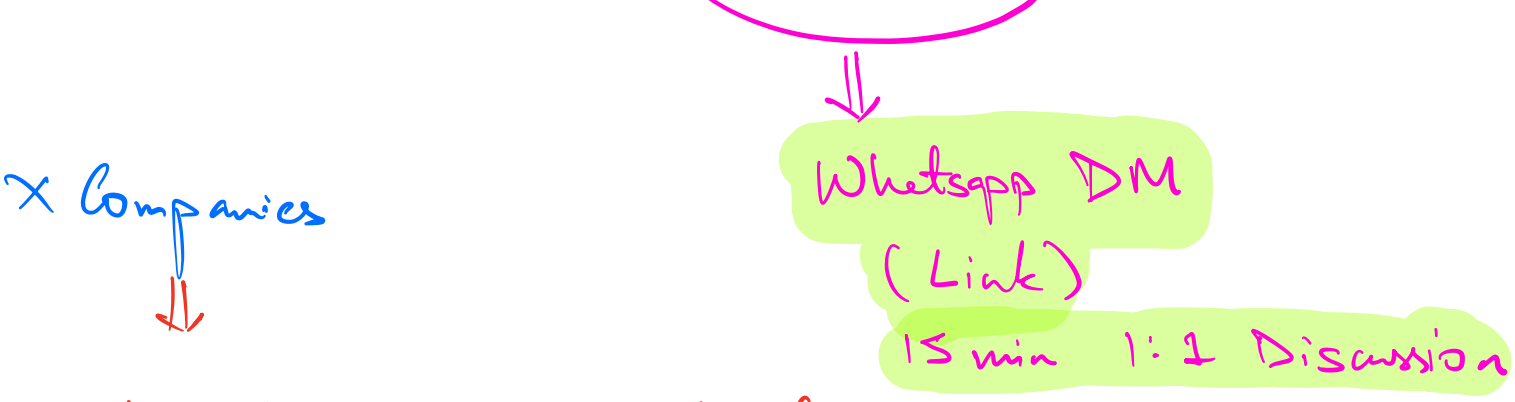
40 ≤ PSP < 60 (Border)

40 < PSP

2 Weeks

3.5L

Fees



Truth Table

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

Basic AND Properties

1) Even / Odd Property. (N Bit no.)

$$\frac{b^{N-1}}{2^{N-1}} \quad \frac{b^{N-2}}{2^{N-2}} \quad \cdots \quad \frac{b^3}{2^3} \quad \frac{b^2}{2^2} \quad \frac{b^1}{2^1} \quad \frac{b^0}{2^0}$$

$$\text{Value} = 2^{N-1} \times b_{N-1} + 2^{N-2} \times b_{N-2} \ldots 2^2 b_2 + 2^1 b_1 + 2^0 b_0$$

1

$$2^{N-1} \times b_{N-1} + 2^{N-2} \times b_{N-2} \ldots 2^2 b_2 + 2^1 b_1 + b_0$$

$$2 \times \left(2^{N-2} b_{N-1} + 2^{N-3} b_{N-2} \ldots 2^1 b_2 + 2^0 b_1 \right) + b_0$$

$\Downarrow$

$$\text{Even} + b_0$$

If
- b_0 (LSB) $= 0 \Rightarrow$ Even
- b_0 (LSB) $= 1 \Rightarrow$ Odd.

Bit Masking

Eg

	7	6	5	4	3	2	1	0
X =	1	0	0	1	0	1	0	1
M = (&)	0	0	0	0	0	0	0	1
Result	0	0	0	0	0	0	0	1

1 (If $b_0 == 1$) Odd
00000001 = 1

0 (If $b_0 == 0$) Even
00000000 = 0

(X & 1)
- 1 (Odd)
- 0 (Even)

$$X = \begin{array}{ccccc} 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{array}$$

$$M = 00100 = (4)$$

2) $A \& 0 = 0$

$$\begin{array}{r} 10101 \\ \& \;\; 00000 \\ \hline 00000 \end{array}$$

3) $A \& A = A$

$$A = \begin{array}{ccccc} 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{array}$$

$$A = \begin{array}{r} 10101 \\ \& \;\; 10101 \\ \hline 10101 \end{array}$$

$1 \& 1 = 1$

$0 \& 0 = 0$

Basic OR Properties

1) $A \mid 0 = A$

2) $A \mid A = A$

Basic XOR Properties

1) $A \wedge 0 = A$

$1 \wedge 0 = 1$

$0 \wedge 0 = 0$

```
A   1 0 1 0 1
    0 0 0 0 0
  ---------
A   1 0 1 0 1
```

2) $A \wedge A = 0$

$1 \wedge 1 = 0$

$0 \wedge 0 = 0$

```
  1 0 1 0 1
  1 0 1 0 1
---------
  0 0 0 0 0
```

Commutative Property

```
  10 ⇒ 1 0 1 0
& 14 ⇒ 1 1 1 0
-------------
       1 0 1 0
```

```
  14 ⇒ 1 1 1 0
& 10 ⇒ 1 0 1 0
-------------
       1 0 1 0
```

$A \& B = B \& A$

$A \mid B = B \mid A$

$A \wedge B = B \wedge A$

The order of operands do not matters.

Associative Property

Grouping of operators do not affect the result of operation

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

Same for OR & XOR

Q Evaluate the expression

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

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

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

$$= 0 \wedge d$$

$$= d$$ Ans

Q Evaluate the Expression

$$1 \wedge 3 \wedge 5 \wedge 3 \wedge 2 \wedge 1 \wedge 5$$

$= (1 \wedge 1) \wedge (3 \wedge 3) \wedge (5 \wedge 5) \wedge 2$

$= 0 \quad 0 \quad 0$

$= 2$

Single Number

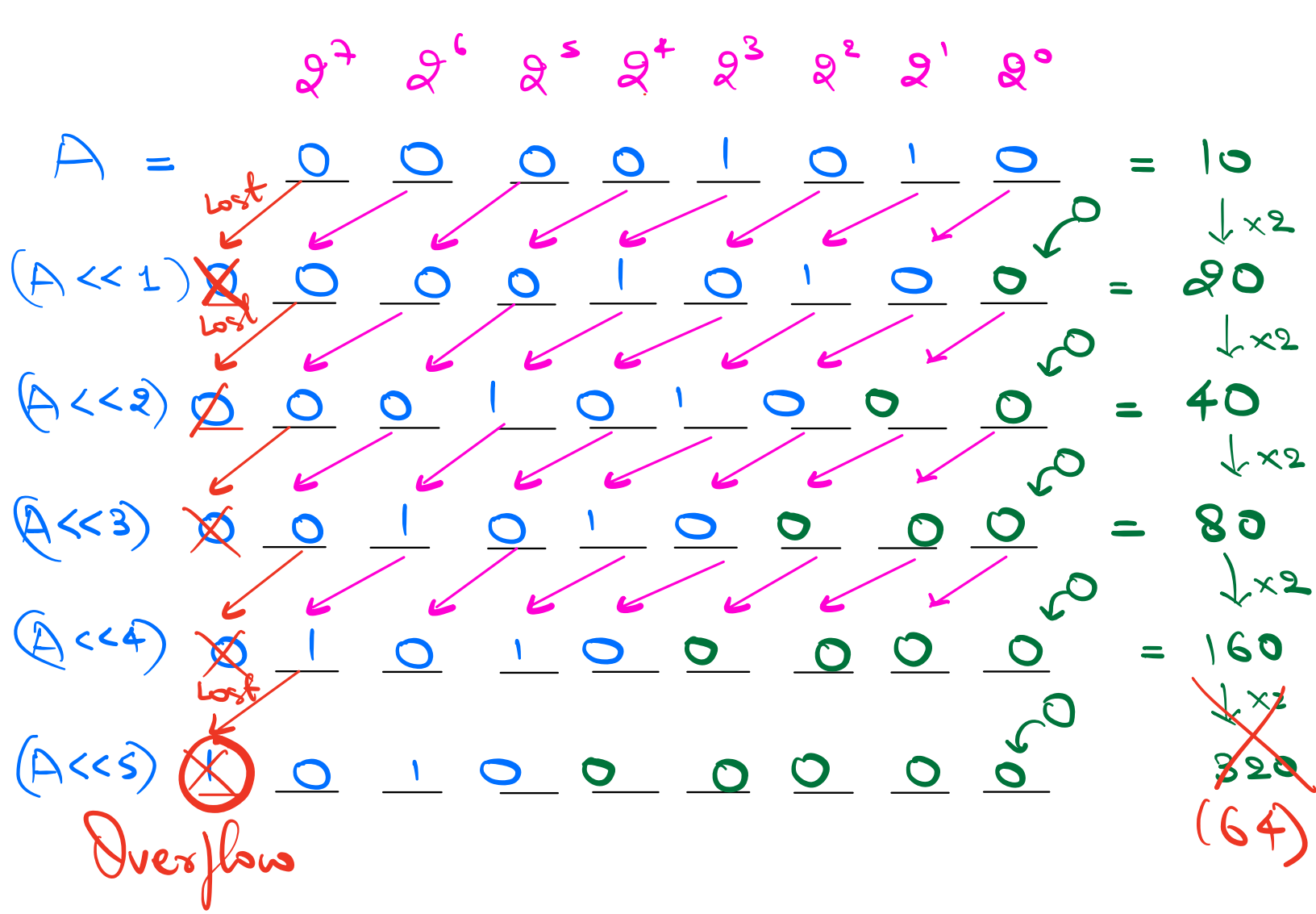
A ⇒ Every no. occurs twice but one find that unique no.

A = [1, 3, 5, 3, 2, 1, 5]

```
ans = 0;
for (i=0; i<N; i++) {
    ans = ans ^ A[i];
}
return ans;
```

Left Shift Operator (<<)

8 bit no.



$10 \xrightarrow[(1)]{<<} 2\times10 \xrightarrow[(2)]{<<} 2^2\times10 \xrightarrow[(3)]{<<} 2^3\times10 \xrightarrow[(4)]{<<} 2^4\times10$

$(A << i) = (A \times 2^i)$ (As long as there is no Overflow)

Right Shift Operator

Assume 8 bit no.

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

A = 0 0 0 0 1 0 1 0 = 10

Lost

$\downarrow \div 2$

(A>>1) 0 0 0 0 0 1 0 1 = 5

$\downarrow \div 2$

(A>>2) 0 0 0 0 0 0 1 0 = 2

(No Overflow)

(0101) $5 \div 2 = 2$

(0100) $4 \div 2 = 2$

$1 << 3 = ??$

$(1 << 3) = 1 \times 2^3 = 8$

(1×2^N)

$(1 << i) = 2^i$

$2^4\ 2^3\ 2^2\ 2^1\ 2^0$

4 3 2 1 0

X = 1 0 1 0 1

M = 0 0 1 0 0 = 2^2 = (1<<2)

i^{th} bit = 2^i

Create a Mask for ith bit

..... (i+1) , i 3 2 1 0

M = 0....0 1 .....0 0 0 0

$M = 2^i = (1<<i)$

Power of Left Shift

1) Given a no. N. Set the ith bit of the number.

7 6 5 4 3 2 1 0

N = 10 ⇒ 0 0 0 0 1 0 1 0

i = 4 OR M = 0 0 0 1 0 0 0 0 (1<<4)

0 0 0 1 1 0 1 0

7 6 5 4 3 2 1 0

N = 10 ⇒ 0 0 0 0 1 0 1 0

i = 3 OR M = 0 0 0 0 1 0 0 0 (1<<3)

0 0 0 0 1 0 1 0

N = N | (1<<i)

2) Given a no. N. Toggle the ith bit

(1 → 0, 0 → 1)

```
                7  6  5  4  3  2  1  0
N = 10 ⇒        0  0  0  0  1  0  1  0
i = 3  XOR M=   0  0  0  0  1  0  0  0      (1<<i)
```

```
                7  6  5  4  3  2  1  0
N = 10 ⇒        0  0  0  0  1  0  1  0
i = 4  XOR M=   0  0  0  1  0  0  0  0      (1<<i)
```

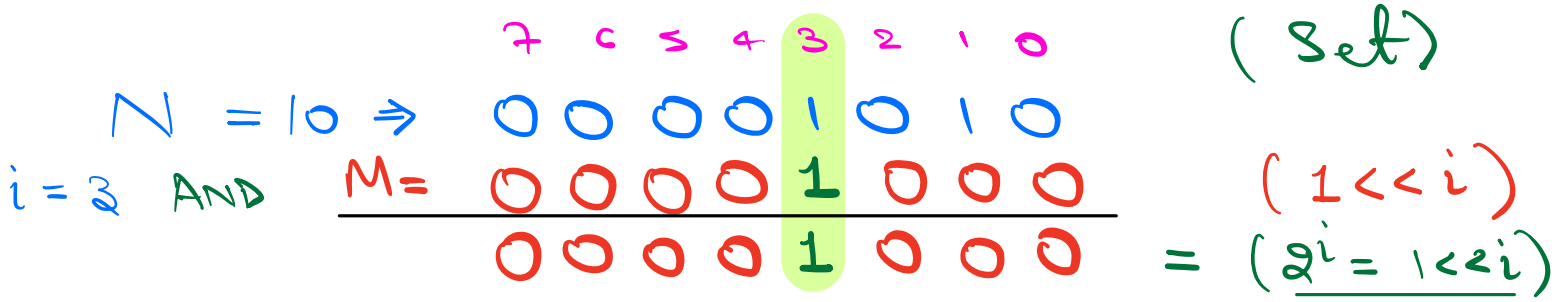
ith bit

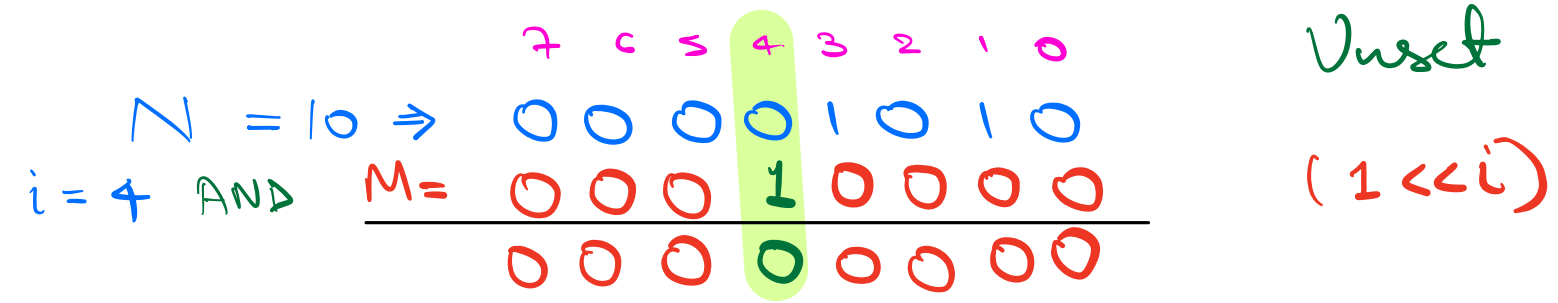
$0 \wedge 1 = 1$

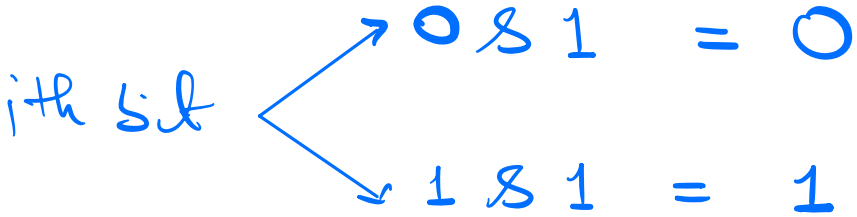
$1 \wedge 1 = 0$

$N = N \wedge (1 << i)$

3) Given a no. N. Check if ith bit set or not








```
bool checkBit (N, i) {
    if ((N & (1<<i)) == 0) {   // the ith bit is unset
        return false;
    }
    else {  ith bit is set
        return True.
    }
}
```

```
N = 1 ;
(N << 1)
print (N);  ⇒ 1
N = (N<<1)
print (N)  ⇒ 2
```

Q

Given an integer N.
Count the no. of set bits in N.

NOTE: Integer has 32 bits.

```
count = 0;
for (i = 0; i < 32; i++) {
        if ( checkBit (N, i)) {
                count++;
        }
}
return count;
```

Approach 2

```
count = 0;
while (N > 0) {
        if ((N & 1) == 1) {
                count++;
        }
        N = (N >> 1);    (Dividing by 2)
}
return count;
```

Integer N ⇒ Count Set bit

Sol^n

Divide N repeatedly by 2
so it becomes 0

⇓

O (log N)

Max Value of Integer (N) = $2^{31} - 1$

Max value of $\log_2 N$ ≅ 31
of an Integer N

Integer N
& set bits from (i to j)

Iterate from [i, j] & set each bit.

Doubt Session

⇓

Poll in Whatsapp.

LLD ⇒ Low Level Design

HLD ⇒ High Level Design.