Bit Manipulation 1

- by Harsh Gupta

Goal

- Understand bit manipulation and basic operators.
- Learn tips for bit manipulation.
- Common patterns in bit manipulation problems.
- Practice some problems

Where is Bit Manipulation used?

Problems may or may not directly involve bit manipulation.

Some problems use bitwise operators in the problem statement itself, whereas some problems might use bit manipulation indirectly.

For example, if the problem involves powers of 2, it might be related to binary. Similarly, powers of other numbers might be in their own base.

How to approach Bit Manipulation problems?

“When solving a bitwise problem, $\boxed{\text{think in bits}}$”

When solving a problem that uses addition, multiplication, etc. we use decimal, which is the appropriate base for us.

When solving a problem that uses a bitwise operator, we need to use binary, which is the appropriate base.

1010

MSB to LSB

1010

2^1

2^0

$\rightarrow (0\times2^0) + (1\times2^1) + (0\times2^2) + (1\times2^3)$

20+2

22

22

$2\times10^0 + 2\times10^1$

$(22)_{10}$

1010

set

(not set)

unset

01

2

1 → -ve

Bitwise Operators - OR

The OR operator ("|") takes every corresponding bit of the two numbers and checks whether ***at least one*** of them is set.

A	B	Result
1	1	1
1	0	1
0	1	1
0	0	0

set

unset

A → 1 1 0 0

B → 1 0 1 0

result 1 1 1 0

1 → good   0 → not good

0 1 0 0 0 $0|0|0|0|0 = \boxed{0}$

$0|1|0|0|1|0 = 1$

$1|1|0|0 \rightarrow 1$

Bitwise Operators - AND

The AND operator ("&") takes every corresponding bit of the two numbers and checks whether ***both*** of them is set.

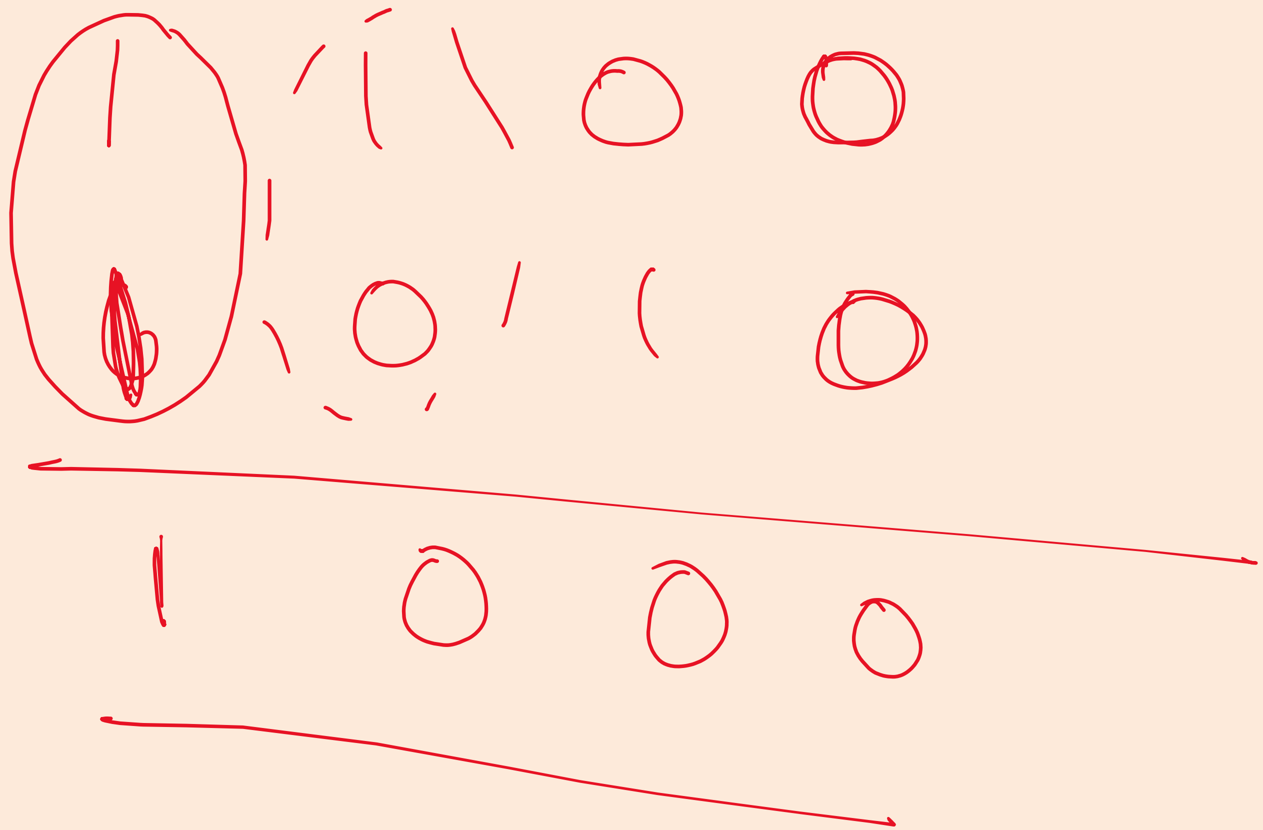
A	B	Result
1	1	1
1	0	0
0	1	0
0	0	0

1 0 0 0
1 1 0 0

&

1 0 1 0
1 1 1 1

1 0 0 0



Bitwise Operators - XOR

The XOR operator (“^”) takes every corresponding bit of the two numbers and checks whether ***exactly one*** of them is set.

A	B	Result
1	1	0
1	0	1
0	1	1
0	0	0

set (1)

A	1	1	0	0
B	1	0	1	0
	0	1	1	0

XOR

n numbers

1 → odd % 2 = 1

0 → (even) % 2 = 0

The result bit is set(1) iff odd number of bits are 1

Else it will be unset(0).

---

sum of bits % 2

odd

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

1

Some Properties of Bitwise Operators

- OR/AND/XOR are associative and commutative.
- A ^ 0 = A
- A ^ A = 0
- If A ^ B = C, then A ^ C = B
- A ^ B ^ B = A
- A & B ≤ min(A,B)
- A | B ≥ max(A,B)
- (A | B) + (A & B) = A + B
- (A & 1) is 1 if A is odd, else 0
- A & (A − 1) is 0 if A is a power of 2 (except when A=0)

1 0 1 0
0 0 0 0

A ^ B ^ B = B ^ A ^ B

A & B & C & D ≤ min(A, B, C, D)

A | B | C | D ≥ max(A, B, C, D)

$\oplus$ $22 + 42 + 44 \rightarrow \boxed{42 + 44 + 22}$

order doesn't matter (bitwise)

	1	0	1	0	toggle
^	1	1	1	1	
	0	1	0	1	

	1	0
0	0	1
0	1	1

$\boxed{1 \;\not< \;\wedge}$

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

0 can't change the bit in XOR observation

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

$A \wedge A = 0$

$A \to 100$
$B \to 101$

$A \wedge B = 001 - C$

$A \wedge C = B$

$A \wedge C \to$
100
001
101

$A \wedge B = C \qquad B \wedge A = C$

$A \wedge C = B \qquad C \wedge A = B$

$B \wedge C = A \qquad C \wedge B = A$

$A \wedge B \wedge C \wedge D = E$

$A \wedge B \wedge E \wedge D = C$

n numbers

1
0
0

10

1
0
1

0

even

1
0
0
1

1

3

$A \& B \le \min(A, B)$

10^1 10^0

friend

α

A → 1 0 1 0

B → 0 1 1 1

0 0 1 0

9 1

8 9

9 1 1 1 1

8 9 9 9 9

1 1 0 0

& 1 1 0 0

1 1 0 0

A → 1 1 1 1

B → 0 1 1 1

0 1 1 1

9 9 8 9 9

9 9 9 1 1

A 1 1 0 1 1

B 1 1 1 1 0 0

A

15

| | | |

B | | |

$\boxed{\min(A, B)}$

A → 1 0 1 1 1 1

B → 1 0 1 1 0 1

1 0 1 1 0 1

$$(A \mid B) + (A \& B) = A + B$$

$$0 \quad + \quad 0$$

$$\boxed{0} \quad \boxed{0}$$

```
1 0 1
1 1 0
```

Alice A

Bob B

$A \mid B$

$A \& B$

```
1 1 0 0
1 0 1 0
```

$1 + 1$

$1 + 1$

$1 + 0 = 1$

$A \mid B$

$A \& B$

$A + B$

A , B

1 0 1

1 0 0

0 0 0

A ∤ B 1 0 0

A | B 1 0 1

A + B

(1+1) (0+0) (1+0)

(1+1) (0+0) (0+1)

A ∤ B) + (A | B) = A + B

$7 + 2 = 9$

$$
\begin{array}{rrrr}
 & 1 & 1 & \\
0 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 \\
\hline
1 & 0 & 0 & 1
\end{array}
$$

$\boxed{1001} \rightarrow \boxed{9}$

$\boxed{2} \rightarrow \boxed{10}$

$$
\begin{array}{r|rr}
2 & 9 & 1 \\
2 & 4 & 0 \\
2 & 2 & 0 \\
2 & 1 & 1 \\
 & 0 &
\end{array}
$$

01001

odd → LSB → $\boxed{\text{always be 1}}$

even + even + even + even = $\boxed{\text{even}}$

$\boxed{a \quad b \quad c \quad d \quad e}$ → 1

$(2^0 \times e) + \boxed{(2^1 \times d)} + (2^2 \times c) + (2^3 \times b) + (2^4 \times a)$

1

even even even even

even

even → $\boxed{\text{LSB} \to 0}$

A 0 1 1 0 1 0 1 0 &1

& 0 0 0 0 0 0 1

0 0 0 0 0 0 1

faster

$A \% 2 == 1$

if (A&1) {

A → odd

}

$\boxed{A \,\&\, A-1}$

$A \to$ power of two

10000 → 2^4

01000 → 2^3

00100 → 2^2

10 → 2

A-1

100000

01111

$\boxed{10^2}$ 10^3 1000

100

A → 10000

A-1 → 01111

&

00000

0-1 = -1

$(A \,\&\, (A-1)) = 0$

$\boxed{A = 0}$ iff A → power of 2

A == 0

1 1
10 0 0
2 2

1 0
2

10

1

-

0 1 1 1

2's complement

$2^0 = 1$

0

power of 2?

Bitwise Operators - Misc

- Bitwise Left Shift ("<<"):

  $\boxed{A \ll B}$ shifts the A to the left by B bits, adding B zeros at the end. This value is the same as $\boxed{A \times 2^B}$.

$$18 \ll 3 = 144$$

$\boxed{A / 2^B}$

- Bitwise Right Shift (">>"):

  $A \gg B$ shifts the A to the right by B bits, deleting B zeros from the end. This value is the same as floor of $A \div 2^B$.

$$18 \gg 3 = 2$$

$5/2=2$

$(A \gg B)$

$\boxed{A / 2^B}$

floor division

Left Shift

$$\boxed{(A << B) \rightarrow A \times 2^B}$$

$$(A << B) \rightarrow (A \times 2^3)$$

10100

$A \rightarrow \boxed{101}$

$$(5 << 2) \rightarrow 5 \times 2^2 = \boxed{20}$$

10100

10100

floor $5/2 \rightarrow 2$

$\text{ceil}(5/2) = \boxed{3}$

2.5

$101 \rightarrow 1010 \rightarrow 10$

5

$\times 2$

$10100 \rightarrow 20$

1 0 1 1 1 0

5 → 1 0 1 | 2 → $\boxed{5/2^1}$

5/4

$\boxed{5 >> 1}$ → $5/2^1 = 5/2 = 2$

$\boxed{5 >> 2}$

5 >> 1   1 0 1

0 0 0 0 0 1 0 1

1 0   1 0 1 0

1 0 1

1

$5/2^2 = 5/4 = \boxed{1}$

Manipulating bits

- Set the k-th bit (OR) ← $A \mid (1 \ll k)$

(0th index)

- Check if k-th bit is set or not (AND) $A \,\&\, (1 \ll k)$
- Toggle the Kth bit (XOR)

$0 \to 1$
$1 \to 0$

$A \wedge (1 \ll k)$

- Unset the Kth bit (XOR)

$1 \to 0$
$0 \to 0$

if bit is set, toggle it

A

5 4 3 2 1 0

1 0 1 1 0 0

1<<4

0 0 0 0 1

0 0 0 0 1 0 0 0 0

Set the 4th bit for me → 1 1 1 1 0 0

1<<K

1 0 1 1 0 0

1 0 1 1 0 0

0 1 0 0 0 0

OR

1 1 1 1 0 0

1 1 1 1 0 0

$\underline{A}$

kth bit

$\& \; (1 << k)$

5	4	kth 3	2	1	0
1	0	1	1	0	0
0	0	1	0	0	0
0	0	1	0	0	0
0	0	0	0	0	0

→ 0

3rd bit is set or not?

$\& \, 1$

$\boxed{1 << 0}$

1×2^0

1

$(A \,\&\, (1<<k)) \,!= 0$

kth bit is set

$\underline{A \,\&\, (1<<k)} = \underline{0}$ → kth bit is $\underline{\text{unset}}$

4th bit

A 1 1/0 0 0 1 0

1<<K

1<<4

0 1 0 0 0 0

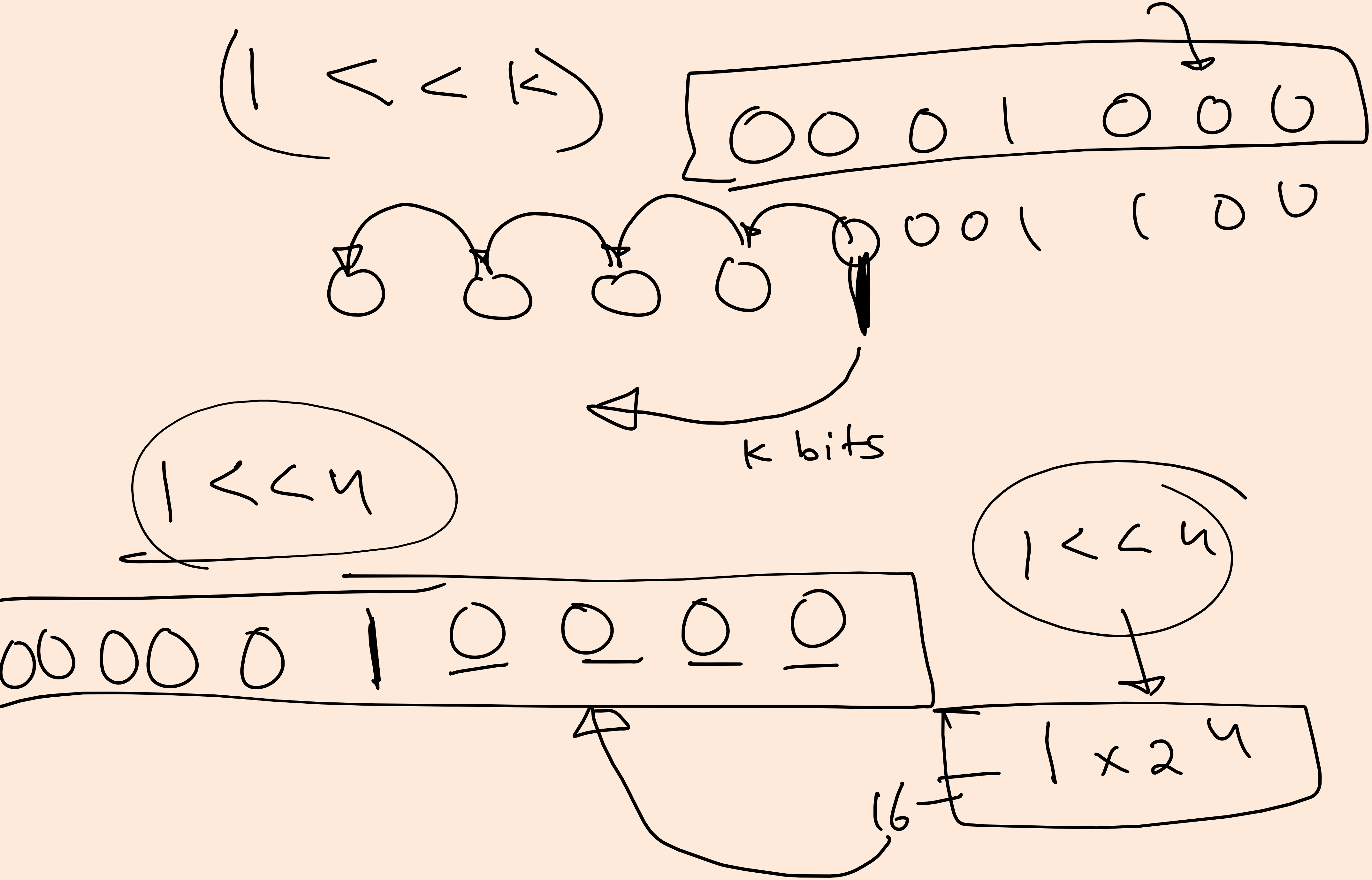
1 0/1 0 0 1 0

$\boxed{A \wedge (1 << K)}$ → Kth bit will get toggled.

$\boxed{1 \to 0} \to$

$0 \to 0$

```
if (A & (1<<k)){
    A = A ^ (1<<k)
}
```



(1 << k)
0001000
k bits
1<<4
00000 1 0000
1<<4
1x2^4
16

Bitwise Equation

Resources

Number base:

https://brilliant.org/wiki/number-base/

Bit manipulation:

- https://codeforces.com/blog/entry/73490 (highly recommended)
- https://www.hackerearth.com/practice/basic-programming/bit-manipulation/basics-of-bit-manipulation/tutorial/

Resources

More properties:

https://stackoverflow.com/questions/12764670/are-there-any-bitwise-operator-laws

Useful tricks and formulas:

- https://www.geeksforgeeks.org/bitwise-hacks-for-competitive-programming/
- https://www.geeksforgeeks.org/bit-tricks-competitive-programming/
- https://www.geeksforgeeks.org/bits-manipulation-important-tactics/
- https://www.geeksforgeeks.org/builtin-functions-gcc-compiler/

Resources - Extra

On bitwise operators:

https://medium.com/biffures/bits-101-120f75aeb75a

https://medium.com/biffures/part-2-the-beauty-of-bitwise-and-or-cdf1d8d87891

https://medium.com/biffures/part-3-or-and-20ccc9938f05

https://medium.com/biffures/part-4-bitwise-patterns-7b17dae3eee0