

Bit Manipulation Basics

Content

- Decimal Number System
- Binary Number System
- Binary to Decimal Conversion
- Decimal to Binary Conversion
- Addition of Decimal Number
- Addition of Binary Number
- Bitwise Operator
- -ve No. Binary Representation {2's complement}
- Range of Datatypes
- Importance of Constraints

~ 69.5 → 75 → 80
Average Psp

↑ more Psp ↑ confidence ↑ better contest performance

If you are stuck at a problem {max time 25 mins} ,

- Hint 1 → Hint 2 → Video solution
- TA { video call Help Request
- Post in WA group
- Reach out to me .

Decimal Number System

$$\begin{array}{r} 2 \ 1 \ 0 \\ 3 \ 4 \ 2 \end{array} \longrightarrow 300 + 40 + 2 \longrightarrow 3 \times 10^2 + 4 \times 10^1 + 2 \times 10^0$$

$$\begin{array}{r} 3 \ 2 \ 1 \ 0 \\ 2 \ 5 \ 6 \ 3 \end{array} \longrightarrow 2 \times 10^3 + 5 \times 10^2 + 6 \times 10^1 + 3 \times 10^0$$

Binary Number System

digits $\longrightarrow \{0, 1\}$

$$\begin{array}{r} 2 \ 1 \ 0 \\ 1 \ 1 \ 0 \end{array} \longrightarrow 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$
$$= 4 + 2 + 0 = \underline{\underline{6}}$$

Diagram annotations for digits 0 and 1:
0: false, off, unset
1: true, on, set

$$\begin{array}{r} 3 \ 2 \ 1 \ 0 \\ 1 \ 0 \ 1 \ 1 \end{array} \longrightarrow 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$
$$= 8 + 0 + 2 + 1 = 11$$

Binary to Decimal Conversion

convert 1101 to decimal

$$\begin{array}{r} 3 \ 2 \ 1 \ 0 \\ 1 \ 1 \ 0 \ 1 \end{array} \longrightarrow 2^3 + 2^2 + 2^0 = 8 + 4 + 1 = 13$$


Convert 10101 to decimal

$$\begin{array}{r} 4 \ 3 \ 2 \ 1 \ 0 \\ 1 \ 0 \ 1 \ 0 \ 1 \end{array} = 2^4 + 2^2 + 2^0 = 16 + 4 + 1 = \underline{\underline{21}}$$


$$\begin{array}{r} 6 \ 5 \ 4 \ 3 \ 2 \ 1 \ 0 \\ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \end{array} = 2^6 + 2^4 + 2^3 + 2^1$$
$$= 64 + 16 + 8 + 2 = \underline{\underline{90}}$$


Decimal to Binary Conversion

Convert 20 to binary.

	num	remainder	
2	20	0	 Bottom to top $2^4 + 2^2 = 16 + 4 = 20$
2	10	0	
2	5	1	
2	2	0	
2	1	1	
	0		

4 3 2 1 0
1 0 1 0 0

2	90	0		1 0 1 1 0 1 0
2	45	1		
2	22	0		
2	11	1		
2	5	1		
2	2	0		
2	1	1		
	0			

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

Addition of Decimal Numbers

$$\begin{array}{r} 11 \\ 368 \\ + 453 \\ \hline 821 \end{array}$$

Addition of Binary Numbers

$$1 + 1 = 2 \longrightarrow \begin{array}{l} 10 \\ \downarrow \\ \text{decimal} \end{array} \quad \begin{array}{l} 10 \\ \downarrow \\ \text{binary} \end{array}$$

$$\begin{array}{r} 1111 \\ 10101 = 21 \\ + 1101 = 13 \\ \hline 100010 = 34 \\ \hline 2^5 + 2^1 = 32 + 2 = \underline{\underline{34}} \end{array}$$

$$\begin{array}{r} 1110101 = 53 \\ + 100110 = 38 \\ \hline 1011011 = 91 \\ \hline 2^6 + 2^4 + 2^3 + 2^1 + 2^0 \\ 64 + 16 + 8 + 2 + 1 \\ \underline{\underline{91}} \end{array}$$

$$\begin{array}{r} 11 \\ 111 \\ + 11 \\ \hline 110 \end{array}$$

$$1 + 1 + 1 = 3 = 11$$

$$\begin{array}{r} 11 \\ 10110 \\ + 00111 \\ \hline 11101 \end{array}$$

Bitwise Operators

Addition without carry

Binary Operator

		AND	OR	XOR
A	B	$A \& B$	$A B$	$A \wedge B$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

If all 1s \rightarrow 1
else 0

If any 1 \rightarrow 1
else \rightarrow 0

same same
zero game

Unary Operator

	NOT
A	$\sim A$
0	1
1	0

Example

5 & 6

5 1 0 1
6 & 1 1 0

1 0 0 \rightarrow 4

5 & 6 = 4

20 & 45

$$\begin{array}{r} 20 \\ 45 \end{array} \begin{array}{cccccc} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{array}$$

$$000100 \rightarrow 4$$

20 | 45

$$\begin{array}{r} 20 \\ 45 \end{array} \begin{array}{cccccc} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{array}$$

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

$$\begin{aligned} & 2^5 + 2^4 + 2^3 + 2^2 + 2^0 \\ & 32 + 16 + 8 + 4 + 1 \\ & = \underline{\underline{61}} \end{aligned}$$

92 ^ 154

$$\begin{array}{r} 92 \\ 154 \end{array} \begin{array}{cccccc} 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{array}$$

$$\underline{\underline{11000110}}$$

~92

$$\begin{array}{r} 92 \\ \sim 92 \end{array} \begin{array}{cccccc} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{array}$$

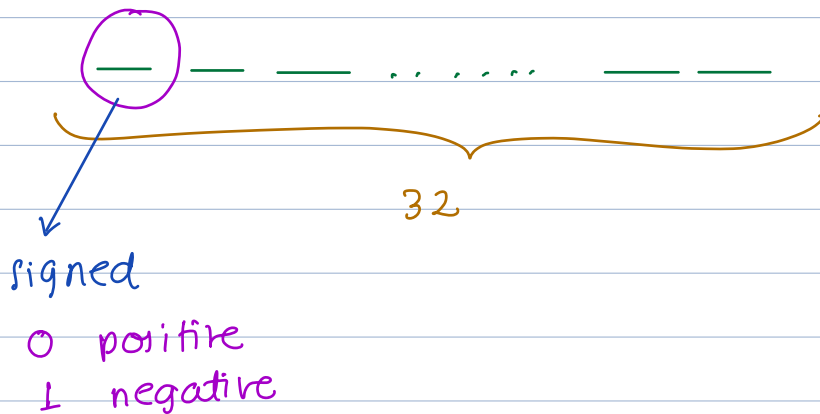
$$\begin{array}{r} 20 \\ \wedge 45 \end{array} \begin{array}{cccccc} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{array}$$

$$\underline{\underline{111001}}$$

Binary Representation of -ve

Very first bit acts as the signed bit

int \longrightarrow 4 bytes \longrightarrow 32 bits



For simplicity we will assume only 8 bits.

-5 in 8 bit representation.

Steps for 2s complement.

\longrightarrow Ignore the sign and write binary repre..

0 0 0 0 0 1 0 1

$\longrightarrow \sim N$

1 1 1 1 1 0 1 0

{ 1's complement }

$\longrightarrow +1$

7	6	5	4	3	2	1	0
1	1	1	1	1	0	1	1

$$\begin{aligned}
 & -2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^1 + 2^0 \\
 &= -128 + 64 + 32 + 16 + 8 + 2 + 1 \\
 &= -5
 \end{aligned}$$

2's complement

NOTE → 2's complement is how the -ve no. is stored

- ① Binary rep. of absolute value say N
- ② $\sim N$
- ③ $+1$

Q> -3 in 8 bit representation.

Step 1 0 0 0 0 0 0 1 1

Step 2 1 1 1 1 1 1 0 0

Step 3 7 6 5 4 3 2 1 0
 1 1 1 1 1 1 0 1

$$\begin{aligned}
 & -2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^0 \\
 &= -128 + 64 + 32 + 16 + 8 + 4 + 1 \\
 &= -128 + 125 = -3
 \end{aligned}$$

-10 in 8 bit representation

step 1 0 0 0 0 1 0 1 0

step 2 1 1 1 1 0 1 0 1

step 3 1 1 1 1 0 1 1 0



$$-2^7 + 2^6 + 2^5 + 2^4 + 2^2 + 2^1$$

$$= -128 + 64 + 32 + 16 + 4 + 2$$

$$= -128 + 80 + 37$$

$$= -128 + 118 = -10$$

Break 22:50

Range of Datatypes

# bits	min	max	Range
2	$\begin{array}{c} \underline{1} \ \underline{0} \\ \downarrow \\ -2^1 + 0 = -2 \end{array}$	$\begin{array}{c} \underline{0} \ \underline{1} \\ \downarrow \\ 1 \end{array}$	$[-2, 1]$
3	$\begin{array}{c} \underline{1} \ \underline{0} \ \underline{0} \\ -2^2 = -4 \end{array}$	$\begin{array}{c} \underline{0} \ \underline{1} \ \underline{1} \\ 3 \end{array}$	$[-4, 3]$
4	$\begin{array}{c} \underline{1} \ \underline{0} \ \underline{0} \ \underline{0} \\ -2^3 = -8 \end{array}$	$\begin{array}{c} \underline{0} \ \underline{1} \ \underline{1} \ \underline{1} \\ 7 \end{array}$	$[-8, 7]$
8	$\begin{array}{c} \overset{7}{\underline{1}} \ \overset{6}{\underline{0}} \ \overset{5}{\underline{0}} \ \overset{4}{\underline{0}} \ \overset{3}{\underline{0}} \ \overset{2}{\underline{0}} \ \overset{1}{\underline{0}} \ \overset{0}{\underline{0}} \\ -2^7 \end{array}$	$\begin{array}{c} \overset{7}{\underline{0}} \ \overset{6}{\underline{1}} \ \overset{5}{\underline{1}} \ \overset{4}{\underline{1}} \ \overset{3}{\underline{1}} \ \overset{2}{\underline{1}} \ \overset{1}{\underline{1}} \ \overset{0}{\underline{1}} \\ 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 \\ = 2^7 - 1 \end{array}$	$[-128, 127]$
32	$\begin{array}{c} -2^{31} \\ \downarrow \\ \approx -2 * 10^9 \end{array}$	$\begin{array}{c} 2^{31} - 1 \\ \downarrow \\ \approx 2 * 10^9 \end{array}$	$[-2 * 10^9, 2 * 10^9]$
64	$\begin{array}{c} -2^{63} \\ \downarrow \\ 2^{10} = 1024 \approx 1000 \\ 2^{10} * 2^{10} * 2^{10} * 2^{10} * 2^{10} * 2^{10} \approx (10^3)^6 \approx 10^{18} \\ - 8 * 10^{18} \end{array}$	$\begin{array}{c} +2^{63} - 1 \end{array}$	Range $-8 * 10^{18}$ to $+8 * 10^{18}$

Importance of Constraints ***

```
int a = 105
int b = 106
```

int range $[-2 \times 10^9 \text{ to } 2 \times 10^9]$

```
int c = a * b
      = 1011 → overflow of integer
```

```
long c = a * b
        = 1011
```

1st operation $a * b$

$\text{int} * \text{int} = \text{int}$

The result will be int

This result will be stored in c

```
long c = (long) a * b
```

$\text{long} * \text{int} = \text{long}$

$= 10^{11}$ // correct value

steps → a will be typecasted to long

→ $\text{long } a * \text{int } b$

→ temp result will be stored as long

→ c is assigned temp result.

Constraints → $N \rightarrow [1 \text{ to } 10^5]$

$A[i] \rightarrow [1 \text{ to } 10^6]$

Cannot be
stored in int

```
int sum = 0
```

```
for i → 0 to N-1
```

```
    sum += A[i]
```

```
print(sum) // garbage value
```

$N = 10^5$

$A[i] = 10^6$

$10^5 * 10^6 = 10^{11}$

```
long sum = 0
for i → 0 to N-1
|   sum += A[i]
|
|
print(sum) // right value
```

No confusion code

```
long sum = 0
for i → 0 to N-1
|   long x = A[i]
|   sum += x
|
|
print(sum) // right value
```

HW → Take simple code examples and print the results.

Doubt session

(result % MOD) $\rightarrow 10^9 + 7$

$0 \leq x \% 15 \rightarrow < 15$

oracle jdk i/o \rightarrow ArrayList

Array support \rightarrow int[]

How to solve problem ?

\rightarrow Read the question

\rightarrow Understand input \rightarrow output

\rightarrow Bruteforce

\rightarrow your code should give right result on simple tests

\rightarrow optimise

Exceeded 25 mins on a question

\rightarrow Hint 1

\rightarrow Hint 2

\rightarrow Video soln

\rightarrow TA { video call }

\rightarrow whatsapp

\rightarrow Text me