1) Generate the missing no.

```
ans = 0;
for (i=0; i<32; i++) {
    count = 0;
    for (j=0; j<N; j++) {
        if (checkBit(A[j], i)) {
            count++;
        }
    }
    if (count % 3 == 1) {
        ans = ans | (1<<i);
    }
}
return ans;
```

XOR ⇔ Toggle $0 \wedge 1 = 1$, $1 \wedge 1 = 0$

					i					
	8	7	6	5	4	3	2	1	0	
ans	0	0	0	0	0	0	0	0	0	
M =	0	0	0	0	1	0	0	0	0	OR
	0	0	0	0	1	0	0	0	0	

$M = (1 << i)$

Q Given an array of N elements
Google Return the max & value of any pair

max (A[i] & A[j]) ∀ i, j (i != j)

A: 27, 18, 20,

27 & 18 | 27 & 20 | 18 & 20

27: 11011
18: 10010
10010 = 18 → Max = ans

27: 11011
20: 10100
10000 = 16

18: 10010
20: 10100
10000 = 16

Solⁿ ▷ Brute force

Calculate & for all pairs & take max

```
ans = INT_MIN
for (i = 0; i < N; i++) {
    for (j = i+1; j < N; j++) {
        andValue = A[i] & A[j];
        ans = max(ans, andValue);
}
```

}
return ans;

T.C. = $O(N^2)$

0000 > 0111

Optimize

A: 26, 13, 23, ~~28~~, 27, 7, 25, 27

	4	3	2	1	0
26 :	1	1	0	1	~~0~~
13 :	~~0~~	1	1	0	1
23 :	1	~~0~~	1	1	1
28 :	1	1	1	~~0~~	0
27 :	1	1	0	1	1
7 :	~~0~~	0	1	1	1
25 :	1	1	0	~~0~~	1
27	1	1	0	1	1
ans =	1	1	0	1	

28 -

-1

⇒ ans

$P1 = (i_1 \& j_1)$

⇓

and (4) = 1

$P_2 = (i_2 \& j_2)$

⇓

and (4) = 0

Code

for (i = 31 ; i >= 0 ; i--) {

```
count = 0;
for (j = 0; j < N; j++) {
    if (A[j] != -1) {
    if (A[j] & (1<<i) > 0) {
        count++;
    }
    if (count >= 2) {
        break;
    }
    }
}
=> if (count >= 2) {   // we can eliminate
    for (j = 0; j < N; j++) {
        if (A[j] != -1) {
        if (A[j] & (1<<i) == 0) {
            A[j] = -1;
        }
        }
    }
}
}
```

Ans => calculate

$T.C. = O(32 * N) \simeq (N \log N)$

$O(N)$

2^{32}

$\log N = 32$

Binary Representation of -ve no's

1 plot

father ⇒ N plots

⇓

1 child

Father = 0

Child ⇒ all +ve no's

_ _ _ _ _ _ _ _ _

(1/0)

32 bits

⇒ 2^{32} possibilities = N

$2^{32} - 1$ ⇒ +ve elements

1 ⇒ 0

Unsigned int ⇒ only +ve + 0

$[0, 2^{32}-1]$

0 Father $\left(\frac{2^{32}}{2}\right)$ $\frac{2^{32}}{2}$

Girl (-ve) 2^{31}

Boy (+ve) $2^{31} - 1$

$2^{31} \ldots -3, -2, -1, 0, 1, 2, 3, \ldots\ldots, 2^{31-1}$

$2^{31} \Downarrow -2^{31}$

Signed int (32 bits) $\Rightarrow \left[-2^{31}, 2^{31} - 1\right]$

MSB

$\underline{\quad}$ $\underline{\quad}$ $\underline{\quad}$ .... $\underline{\quad}$ $\underline{\quad}$ $\underline{\quad}$

2^{31} 2^{30} 2^{29} 2^{2} 2^{1} 2^{0}

Sign bit (MSB)
- 0 $\Rightarrow$ +ve
- 1 $\Rightarrow$ -ve

N Bit no. $\Rightarrow$ (N-1 bits) + 1 Bit

(N-1 bits) $\Downarrow$ Value

1 Bit $\Downarrow$ Sign

Assume 8 bits (Easy understanding)

10 : 0 0 0 0 1 0 1 0

$\Downarrow$ Sign bit

-10 : 1 0 0 0 1 0 1 0

$\Downarrow$ Sign bit

$x = 0$, $x = x \times -1$

$= -0$

0 : 0 0 0 0 0 0 0 0

-0 : 1 0 0 0 0 0 0 0 = 0

$$\begin{array}{r} 10 \\ +(-10) \\ \hline 0 \end{array} \qquad \begin{array}{r} 00001010 \\ 10001010 \\ \hline 10010100 \end{array} \rightarrow 0 ?? \times$$

1s compliment ($1 \rightarrow 0$, $0 \rightarrow 1$)

10 : 0 0 0 0 1 0 1 0

-10 1 1 1 1 0 1 0 1

0 : 0 0 0 0 0 0 0 0

-0 : 1 1 1 1 1 1 1 1

2's Compliment

10 : 0 0 0 0 1 0 1 0

1's Comp : 1 1 1 1 0 1 0 1

Add 1 : 0 + 1

2's Compl: 1 1 1 1 0 1 1 0

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

0

$1 \times (-2^7) + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3$

$+ 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

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

$= -10$

$0:\quad 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$

1's com $\quad 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1$

Add 1 $\quad\quad\quad\quad\quad\quad + 1$

$-0:\quad 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$

---

int x = 5

signed

x = -5;

print (x); → -5

unsigned int x = 5

x = -5 → Error

# Bits	Unsigned Min	Unsigned Max	Signed Min	Signed Max
1) __ __ (2)	0	$2^2 - 1$	$(1\,0) = -2^1$ (-2)	$(0\,1) = 2^1 - 1$ (1)
2) 5 Bit	0	$1\,1\,1\,1\,1$ $2^0+2^1+2^2+2^3+2^4 = 2^5-1$	$1\,0\,0\,0\,0$ ⇓ -2^4 (-16)	$0\,1\,1\,1\,1$ $2^0+2^1+2^2+2^3 = 2^4-1$ (15)

-2^4 (MSB) $\quad \underbrace{2^3 \quad 2^2 \quad 2^1 \quad 2^0}_{+ve}$

(N Bit)

$$\underline{-2^{N-1}} \quad 2^{N-2} \quad 2^{N-3} \quad \ldots\ldots \quad 2^{3} \quad 2^{2} \quad 2^{1} \quad 2^{0}$$

0 ... 1

$$\text{Min} = -2^{N-1}$$

$$\text{Max} = 2^0 + 2^1 + 2^2 \ldots\ldots 2^{N-2}$$

$$= 2^{N-1} - 1$$

$$\text{Range} \rightarrow \left[-2^{N-1},\ 2^{N-1}-1\right]$$

Approximations.

$$2^{10} = 1024 \simeq 10^3$$

1) $(2^{10})^3 \simeq (10^3)^3$

$$2^{30} \simeq \underline{10^9}$$

(multiply 4 on both side)

$$2^{32} \simeq \underline{(4\times 10^9)}$$

$$2^{31} \simeq 2\times 10^9 \left(\simeq 10^9\right) \Rightarrow \text{INT}$$

2) $(2^{10})^6 \simeq (10^3)^6$

$2^{60} \simeq 10^{18}$ (Multiply 8)

$8 \times 2^{60} \simeq 8 \times 10^{18}$

$2^{63} \simeq 8 \times 10^{18}$

$\Downarrow$ Long

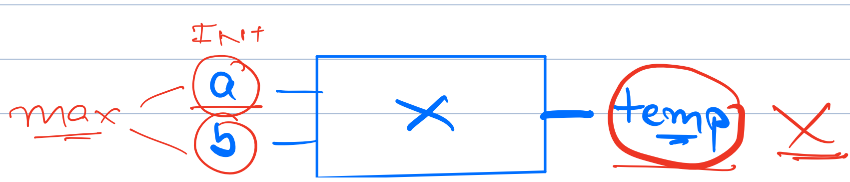
Overflow

int a = 10^5

int b = 10^6 → 10^{11}

int c = a × b ✗

long c = a × b ✗



INT INT

long c = long((a×b)) ✗

long c = (long) a × b ✓

long c = a× (long) b ✓

long c = (long) a × (long) b ✓

TLE

1 GHz Machine ⇒ 1B instructions / sec

10^9

only once

1

i = i+1 (2)

```
for (i = 1; i <= 100; i++) {
    print (i);    1
}
```

Iterations = 100

4 × 100 + 1 = 401 instructions.

Assume

1) 1 iteration = 10 instructions.

1 inst = $\frac{1}{10}$ iteration

10^9 instructions = 10^8 iterations

$N \simeq 10^5 \Rightarrow O(N^2)$ ✗

$\Downarrow$

10^{10}

2) 1 iteration = 100 instructions

10^9 instruction = 10^7 iteration

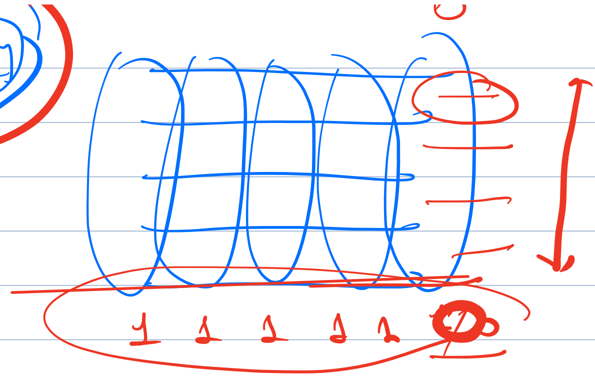
Doubt

21, 32, 45, 67, 82, 7, 3

$\Rightarrow$

T.C. = $O(N^3) \Rightarrow O(N^2)$

Subarray



Atleast 1 = All − None