Good Morning Everyone !!

YOU WILL NEVER HAVE THIS DAY AGAIN, SO MAKE IT COUNT.

26th Nov.

Wed → 7 AM
Fri → 7 AM
Sat → 8:30 PM

Today's content

→ Calculate sum of XOR of all pairs.

→ Max AND pair.

→ 1 Array H.W problem.

Problem solving Session →

Flip, Absolute maximum, Next Permutation, Number of Digit One

Q) Given N +ve elements. Calculate sum of XOR of all pairs.

arr[3] = { 3 1 5 }
(indices 0 1 2)

$\Rightarrow$ {3,3} + {3,1} + {3,5}
{1,3} + {1,1} + {1,5}
{5,3} + {5,1} + {5,5}

idea 1. $\rightarrow$ Consider all the pairs & calculate ans.
T.C $\rightarrow$ $O(N^2)$, S.C $\rightarrow$ $O(1)$

```
sum = 0
for(i=0; i<N; i++){
    for(j=0; j<N; j++){
        sum += a[i]^a[j]
    }
}
```

arr[5] = { 3 5 6 8 2 }
(indices 0 1 2 3 4)

3^3	3^5	3^6	3^8	3^2
5^3	5^5	5^6	5^8	5^2
6^3	6^5	6^6	6^8	6^2
8^3	8^5	8^6	8^8	8^2
2^3	2^5	2^6	2^8	2^2

idea2.:
Iterate on the upper half and calculate the sum.

return 2 * sum.

T.C $\rightarrow$ $O(N^2)$ S.C $\rightarrow$ $O(1)$

idea-3 contribution?

	3	2	1	0		
10 $\rightarrow$	1	0	1	0	$\Rightarrow$	$2^3 + 2^1$
+ 9 $\rightarrow$	1	0	0	1	$\Rightarrow$	$2^3 + 2^0$
+ 7 $\rightarrow$	0	1	1	1	$\Rightarrow$	$2^2 + 2^1 + 2^0$

— + 2×2^1 + 2×2^0

2 → 0010 , 3 → 0011 , 5 → 0101 , 6 → 0110 , 8 → 1000

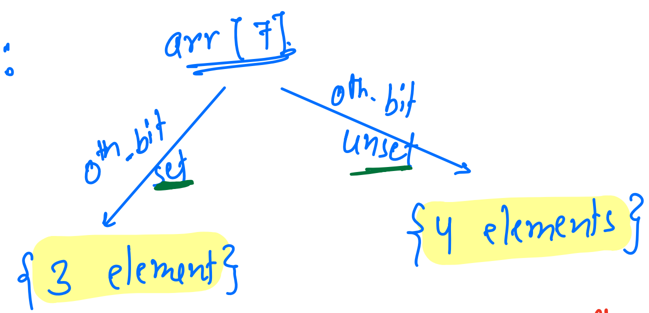
	2^3	2^2	2^1	2^0
3^5 =	0	1	1	0
3^6 =	0	1	0	1
3^8 =	1	0	1	1
3^2 =	0	0	0	1
5^6 =	0	0	1	1
5^8 =	1	1	0	1
5^2 =	0	1	1	1
6^8 =	1	1	1	0
6^2 =	0	1	0	0
8^2 =	1	0	1	0
sum:	$4 * 2^3$	$6 * 2^2$	$6 * 2^1$	$6 * 2^0$

sum: 32 + 24 + 12 + 6 = 74.

ans = 2 * sum = 2 * 74 = 148.

$$
\begin{array}{rl}
a = & \_ \; \_ \; \_ \; \_ \; \overset{0^{th}}{0/1} \,/\, \text{different} \\
\hat{} \;\; b = & \_ \; \_ \; \_ \; \_ \; 1/0 \,/\, \text{different} \\
\hline
 & 1 \\
\hline
\end{array}
$$

idea :



Q) In how many XOR pairs 0th bit will be set ?

idea : If we take one element from set and another element from unset, their xor of 0th bit = 1.

Total pairs = 3 * 4 = 12.

→ We are still considering all upper half pairs & then traversing on all the bits.

T.C → $O(32 \cdot N^2)$

$2 \rightarrow 0010$, $3 \rightarrow 0011$, $5 \rightarrow 0101$, $6 \rightarrow 0110$, $8 \rightarrow 1000$

contribution

In how many elements → $\begin{cases} 0^{th} \text{ bit set : } 2 \\ 0^{th} \text{ bit unset : } 3 \end{cases}$

xor pairs for 0^{th} bit is set $= 6$

$6 * 2^0$

+

In how many elements $\begin{cases} 1^{st} \text{ bit set : } 3 \\ 1^{st} \text{ bit unset : } 2 \end{cases}$

xor pairs for 1^{st} bit is set $= 6$

$6 * 2^1$

+

In how many elements $\begin{cases} 2^{nd} \text{ bit set : } 2 \\ 2^{nd} \text{ bit unset : } 3 \end{cases}$

xor pairs for 2^{nd} bit is set $= 6$

$6 * 2^2$

+

In how many elements $\begin{cases} 3^{rd} \text{ bit set : } 1 \\ 3^{rd} \text{ bit unset : } 4 \end{cases}$

xor pairs for 3^{rd} bit is set $= 4$

$4 * 2^3$

sum = 74

[ans = 2 * sum]

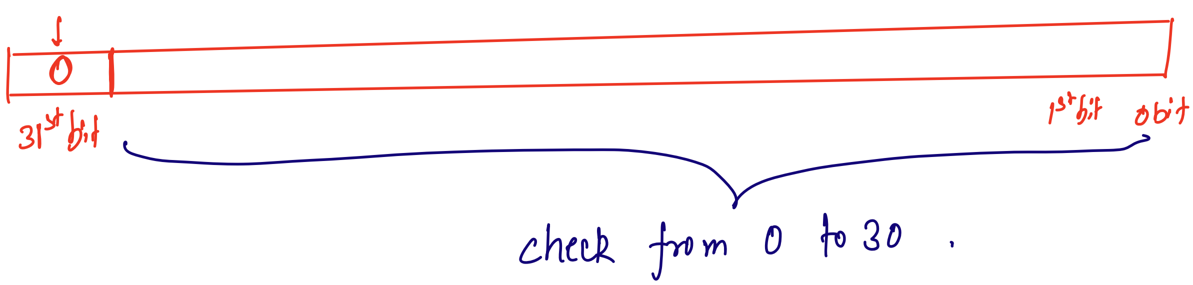
pseudo-code-

```
sum = 0

for( i = 0 ;  i <= 30  ;  i++) {
        // for every bit position, get no. of elements
           for which ith bit is set & unset

        c = 0
        for( j = 0 ; j < N ; j++) {
              if (checkBit ( arr[j], i) ){
                    c++
              }
        }

        //in how many xor-pairs ith-bit set.
                 =  c * (N-c)

        sum += c * (N-c) * 2^i
                           (1<<i)
}

return 2* sum ;
```

total elements = N
set = c
unset = N-c

$T \cdot C \rightarrow O(N)$
$S \cdot C \rightarrow O(1)$

int. for +ve elements



Q) Given N +ve array elements, choose 2 indices (i, j) such that $i \neq j$ and $\{arr[i] \,\&\, arr[j]\}$ is maximum.

↳ bitwise AND

eg → arr[3] : { 27 18 20 } {ans → 18}

(27,18) = 18 (27,20) = 16 (18,20) = 16

27 → 1 1 0 1 1
18 → 1 0 0 1 0
20 → 1 0 1 0 0

eg → arr[4] : { 21 18 24 17 }

21 : 1 0 1 0 1
18 : 1 0 0 1 0
24 : 1 1 0 0 0
17 : 1 0 0 0 1

(21 & 17) → 17
(24 & 21) → 16
(21 & 18) → 16
(18 & 24) → 16

idea 1 : Consider all pairs such that (i != j) and find maximum bitwise AND pair.

idea 2. Select max and second max. Calculate bitwise AND. ✗

1 0 0 0 0 → 2^4

0 1 1 1 1 → $2^3 + 2^2 + 2^1 + 2^0 = 2^4 - 1$

Max AND.
Give preference to leftmost bit

Eg → arr[7] = {26 13 23 28 27 7 25}

	4	3	2	1	0
26 :	1	1	0	1	0
~~13~~0 :	0	0	0	0	0
~~23~~0 :	0	0	0	0	0
~~28~~0 :	0	0	0	0	0
27 :	1	1	0	1	1
~~7~~0 :	0	0	0	0	0
~~25~~0 :	0	0	0	0	0
count set bits	5	4	1 {no pair}	2	1
ans :	1	1	0	1	0

{No discard}

$\underline{2}^4 + \underline{2}^3 + \underline{2}^1 = \underline{\underline{26}}$.

pseudo.code

```
ans = 0

for( i = 30 ; i >= 0 ; i--){
        //count of elements for which ith bit is set
        c = 0
        for( j = 0 ; j < N ; j++){
            if (checkBit( arr[j], i)){
                c += 1
            }
        }

        if ( c >= 2){
            //We can pick a pair for which ith bit
              will be set in ans.
                ans = ans + 2^i
            or { ans = ans | (1<<i) }        ] set-ith bit in ans

            //Discard all elements with ith bit → unset.

            for ( j = 0 ; j < N ; j++){
                if (!checkBit(arr[j], i)) { arr[j] = 0}
            }
}
return ans ;
```

$T.C \rightarrow O(N)$
$S.C \rightarrow O(1)$

1. Maximize Bitwise AND of triplets.
2. Maximize Bitwise AND of quadruplets.
3. Google: Calculate how many pairs are there for which bitwise AND is maximum.

→ After doing all the operations, count no. of non-zero elements in the array = x.

$$ans = {}^{x}C_2 = \frac{x(x-1)}{2}$$

{selecting 2 items from x available items.}

Eg → arr[7] = {26 13 23 28 27 7 25 26 31}

	4	3	2	1	0	
28 :	1	1	1	0	0	(a)
~~26~~ 0 :	0	0	0	0	0	
~~13~~ 0 :	0	0	0	0	0	
~~23~~ 0 :	0	0	0	0	0	
28 :	1	1	1	0	0	(b)
0 ~~27~~ :	0	0	0	0	0	
0 ~~7~~ :	0	0	0	0	0	
0 ~~25~~ :	0	0	0	0	0	
0 ~~26~~ :	0	0	0	0	0	
31 :	1	1	1	1	1	(c)
set-bits	8	7	3	1 {pair}	1 {pair}	
ans :	1	1	1	0	0	

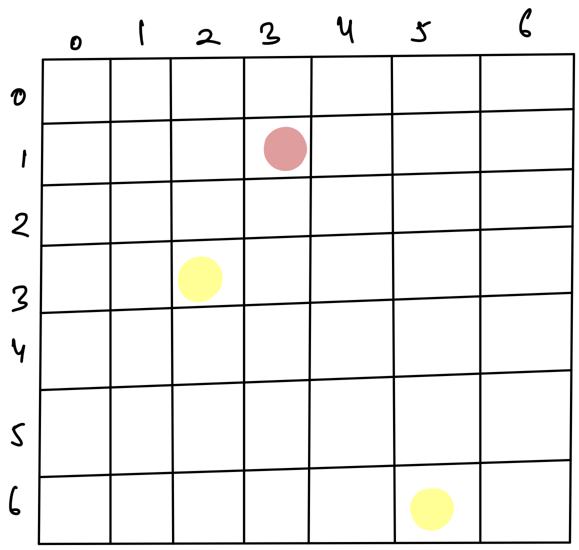
[a & b
b & c
a & c]

→ ${}^{3}C_2$

Max Sum Square Sub-matrix

Given mat[N][N]. find B*B sub-matrix such that sum of all elements in sub-matrix is maximum. [$1 \le B \le N$]

N = 7
B = 4



idea → check every sub-matrix of B*B, iterate on that sub-matrix & get sum.

Finally, get overall maximum sum.

[B*B] [4*4]

TL BR

TL 4-1 BR

0,0 3,3

3,2 6,5

1,3 4,6

{ i, j   i+B-1, j+B-1 }

4,5   7,8

Index Out of Bound!

① pf[][]

pseudo-code:

Step-1. Create pf[][].

```
for ( i = 0 ; i < N ; i++ ) {
    for ( j = 0 ; j < N ; j++ ) {
        // TL → i, j
        // BR → i+B-1 , j+B-1
        if ( i+B-1 < N  &&  j+B-1 < N ) {
            // use pf[][] to get sum of this
               submatrix
        }
    }
}
```

→ [Max sub-matrix sum] → 1st attempt

→ [2-D Kadane's]

27 / 4

↑ dvd   ↑ div

$$dvd = div * quo + remainder.$$

$$remainder = dvd - (div * quo)$$

$$= dvd - div * \{\text{closest factor of div} \le dvd\}$$

⇑

{quotient}

27 → {1 1 0 1 1} (bit positions 4 3 2 1 0), div = 4.

$4*1 \le 27$.   ⇒ $(1<<1)$ [1]

$4*2 \le 27$ → 2^1 [10]

$4*4 \le 27$ → 2^2 [100]

$4*8 \le 27$ → 2^3 [1000]

$4*6 \le 27$ → $2^2+2^1 = 6$.   [110]

$4*7 \not\le 27$ → $2^2+2^1+2^0$ → 7   [111]

1	1 0 0
1 0	1 1 0 (6).
1 0 0	1 1 1
1 0 0 0	