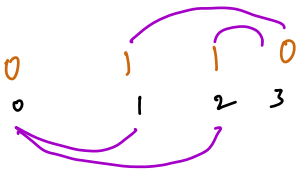
Question: No. of pairs with XOR value as 1 in a binary [0/1] array

(i, j) == (j, i) [i < j]

$x \wedge y = 1$

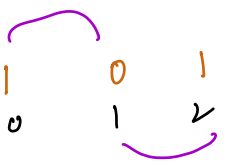
Ex1:



(0,1) (1,3)
(0,2) (2,3)

} 4

Ex2:



(0,1)
(1,2)

} 2 pairs

Brute force:

```
ans = 0;
for (i=0; i<N; i++) {
    for (j = i+1; j<N; j++) {
        if (A[i] ^ A[j] == 1)
            ans++;
    }
}
return ans;
```

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

S.C: $O(1)$

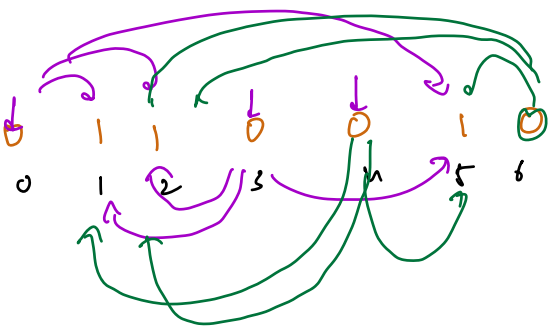
Efficient Approach

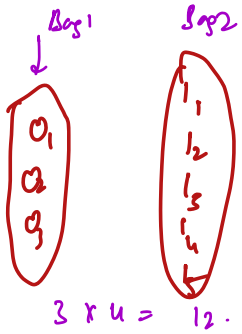
Find no. of (0,1) pairs in the array as their XOR value will be 1

$a \wedge b = 1$

a, b are different in b

Ex:





$3 \times 4 = 12$.

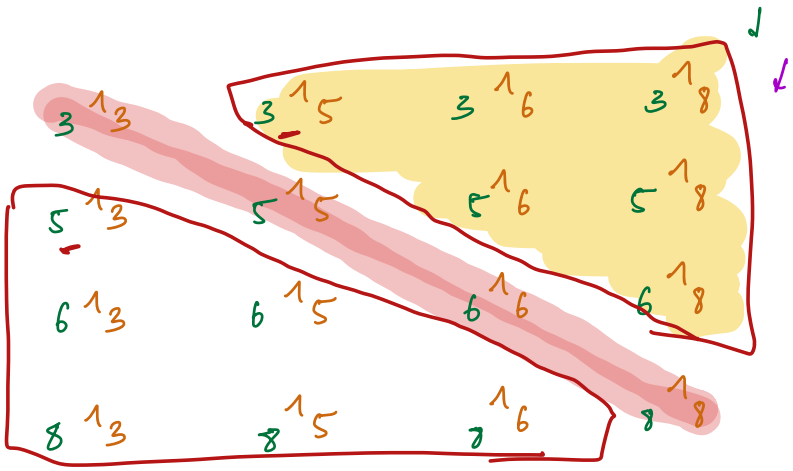
$3 + 6 + 6 + 3 = 12$

Ans : #0's x #1's

Question: Find sum of XOR of all the pairs (i, j) [N^2 pairs]

A = 3 5 6 8

0 1 2 3



pairs : $O(N^2)$

Brute force

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

Observations

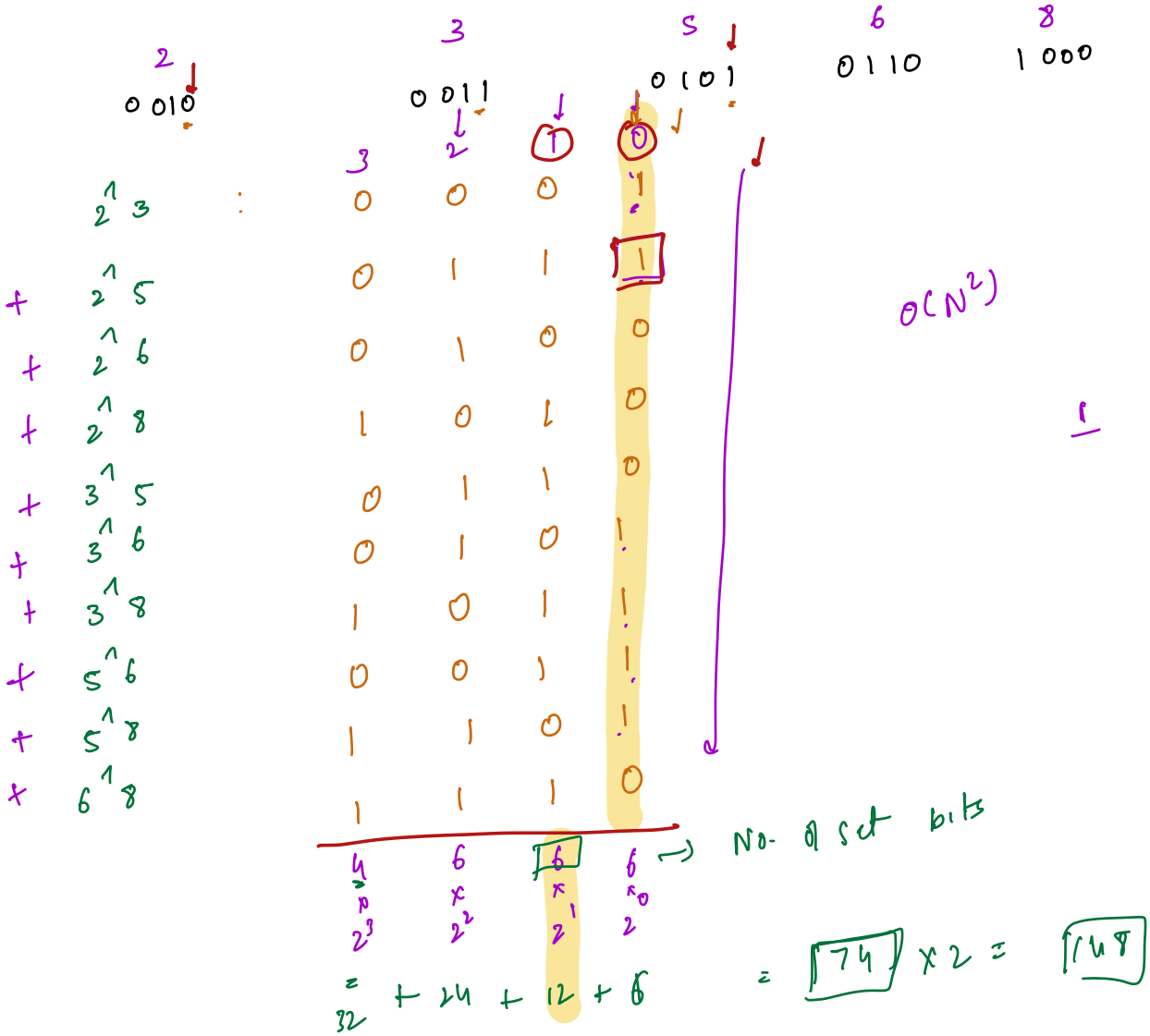
1) A[i] ^ A[i] == 0;
2) A[i] ^ A[j] == A[j] ^ A[i]

```
ans = 0;
for(i=0;  i<N; i++){
      for(j = i+1; j<N; j++){
          ans += A[i] ^ A[j];
      }
}
return  2 * ans;
```

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

S.C : $O(1)$

Efficient Approach



$O(N^2)$

Naive Approach:

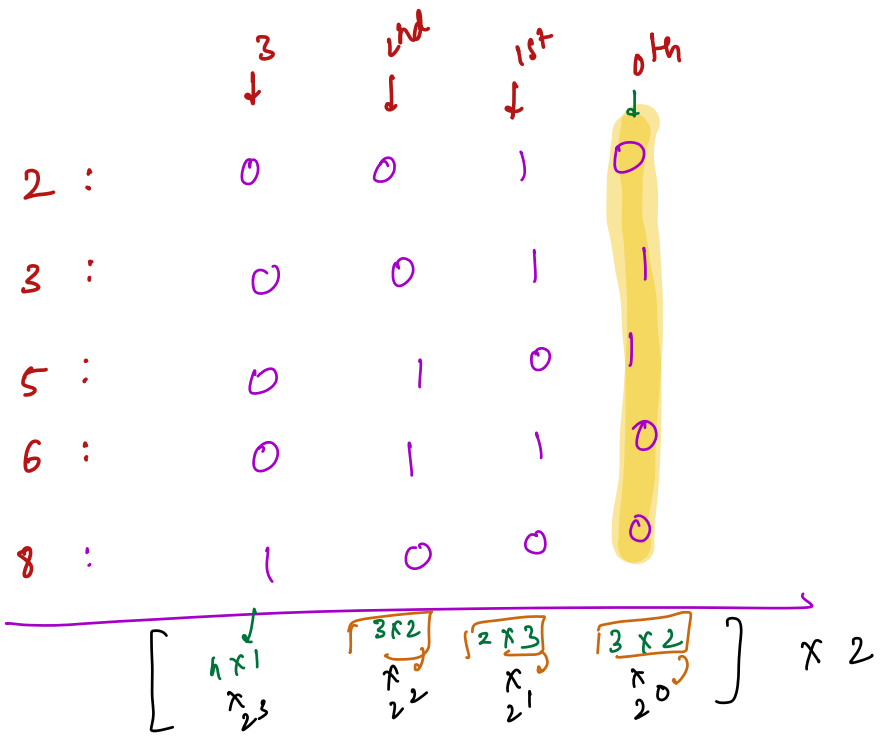
Iterate over all the pairs & find the no. of set bits at ith position

$O(N^2) \times \log(MAX)$

Approach2:

Task: Find the no of set bits at every ith position

0th bit: 0 1 1 0 0 ⇒ 2 × 3 = 6



ans = 0

log(Max) + 1

for(i=0; i < 32; i++){

log(MAX)

count = 0;

for(j = 0; j < N; j++){

O(N)

if(A[j] & 1<<i != 0)

count++;

}

#1's #0's 2^i

ans = (count) * (N-(count)) * (1<<i)

}

return ans * 2;

T.C: O(N log(MAX))

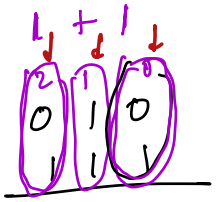
S.C: O(1)

1<<i = 2^i

Question: Different Bits Sum Pairwise

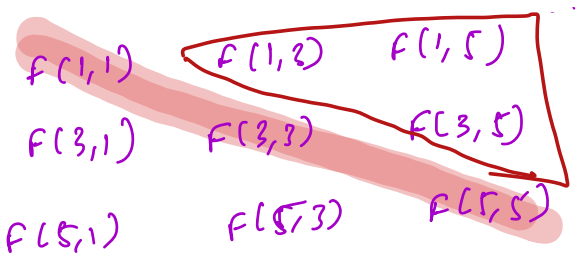
F(X, Y) : No. of different corresponding bits

F(2, 7) = 2



Find $\sum F(A_i, A_j)$ $\forall$ pairs (i, j)

A = 1 3 5



Brute force:

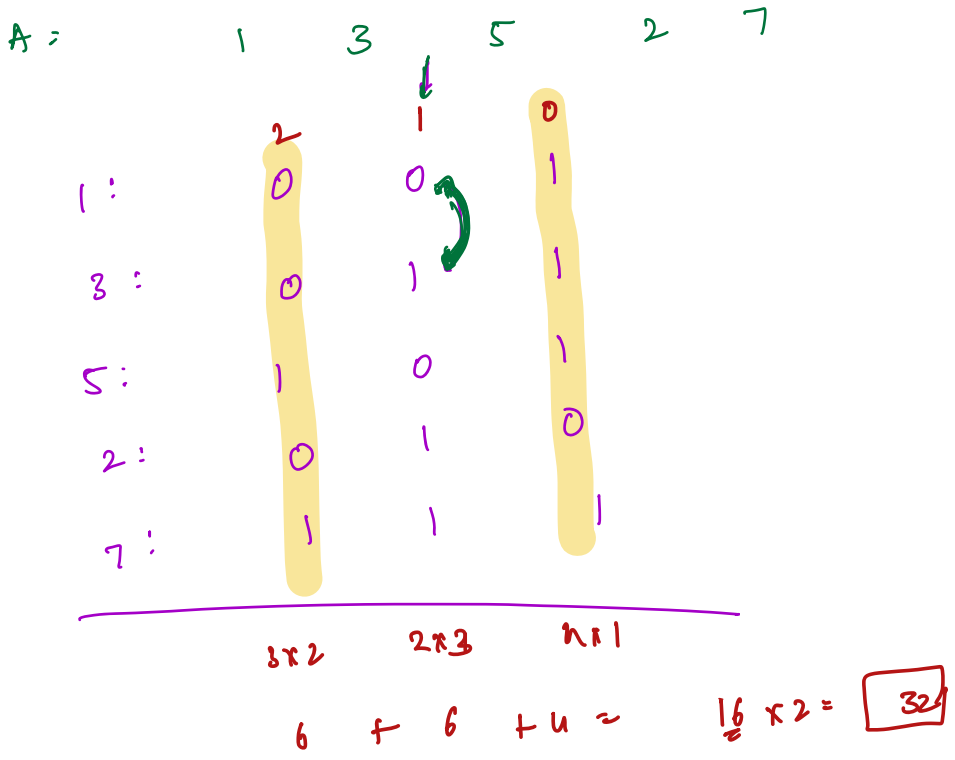
→ Consider all the pairs

→ for each pair, find no. of different bits

T.C: $O(N^2 \times \log(\text{Max}))$

S.C: $O(1)$

Efficient Approach



6 + 6 + 4 = 16 x 2 = 32

T.C: O(log(MAX) x N)

S.C: O(1)

Question: given an array of N positive integers, find the max AND value of any pair

max(A[i] & A[j]) where i != j

A : 27 18 20

27: 1 1 0 1 1
18: 1 0 0 1 0
---
1 0 0 1 0 => 18

27: 1 1 0 1 1
20: 1 0 1 0 0
---
1 0 0 0 0 => 16

18: 1 0 0 1 0
20: 1 0 1 0 0
---
1 0 0 0 0 => 16

11 mins

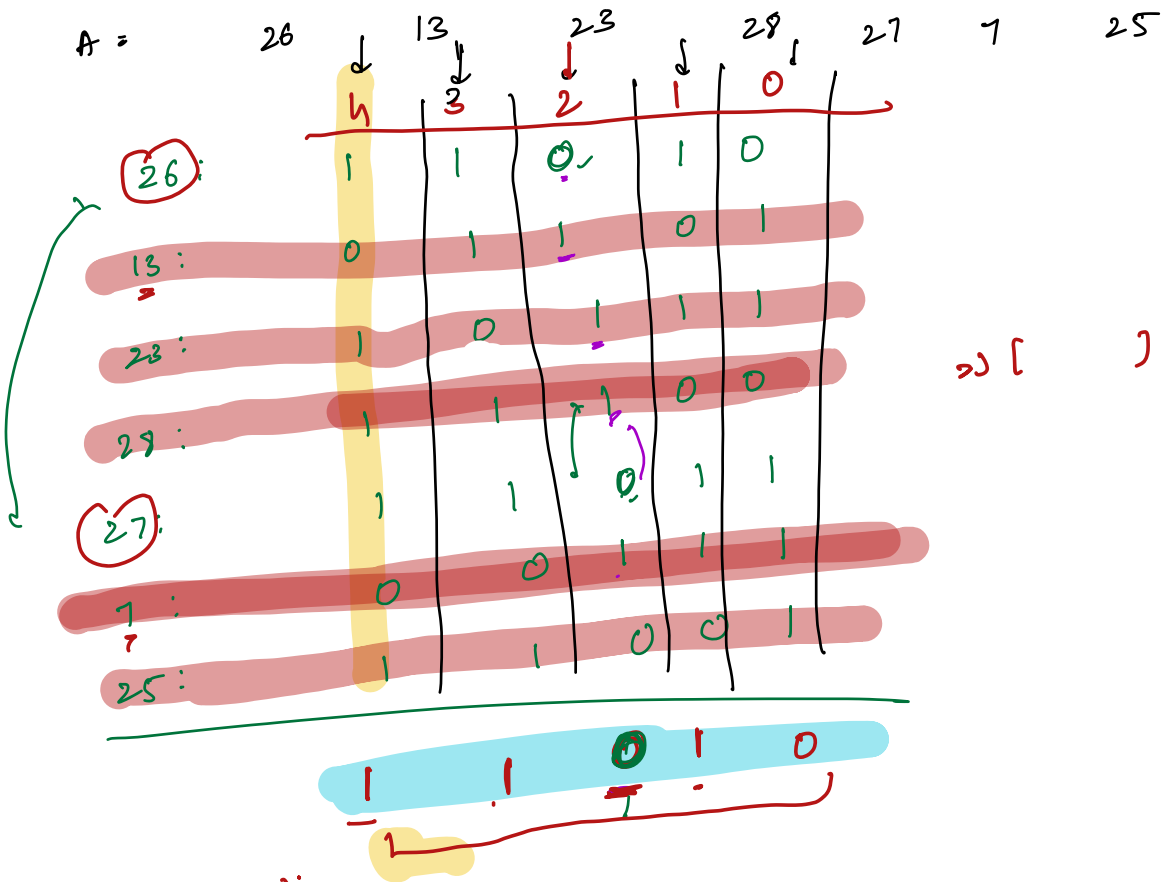
Brute force

consider all pairs

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

S.C: $O(1)$

Approach 2:

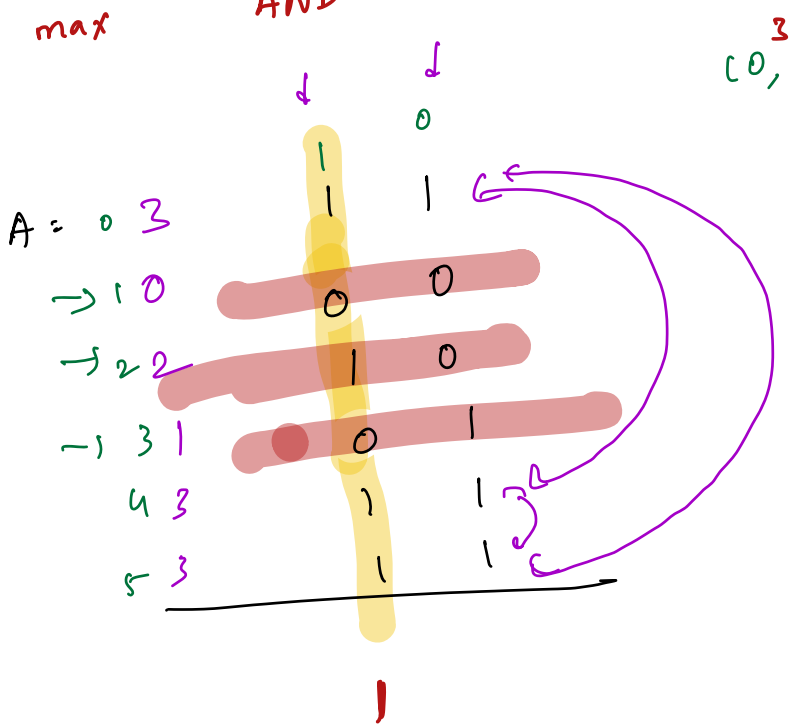



```
ans = 0;
for (i = log(MAX)+1;  i ≥ 0;  i--) {
    cnt = 0;
    for (j = 0; j < N; j++) {
        if (A[j] & (1<<i) != 0)
            cnt++;
    }                                   O(N)
    if (cnt ≥ 2) {
        ans = ans | (1<<i);
        for (k = 0; k < N; k++) {
            if (A[k] & (1<<i) == 0)
                A[k] = 0;
        }
    }                                   O(N)
}                                       log(MAX)
return ans;
```

Total T.C: $O(N \times \log(MAX))$

S.C: S.C: $O(1)$

Question: Find max AND No. of pairs with this value.



Question: Google

Find max $(A[i] \& A[j] \& A[k] \& A[l])$

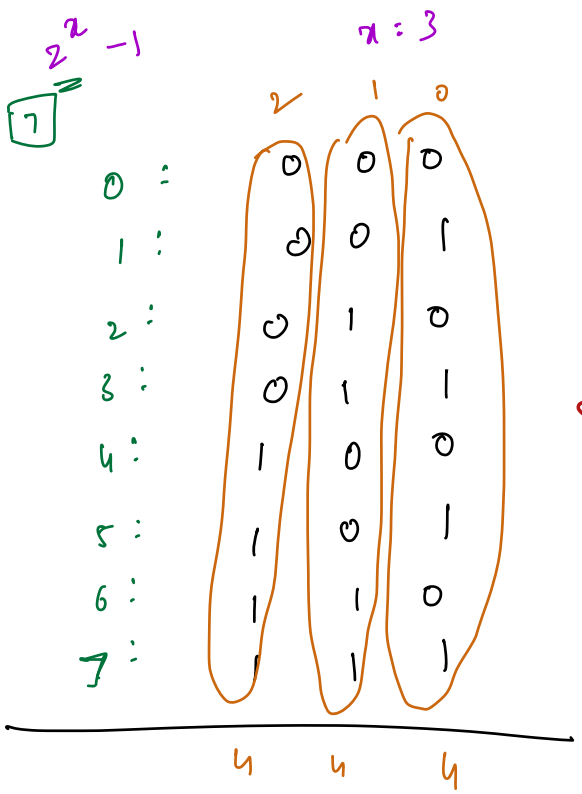
where $i \neq j \neq k \neq l$

N = 4

0 1
1 0
1 1
1 0 0

} 1+1+2+1 = 5

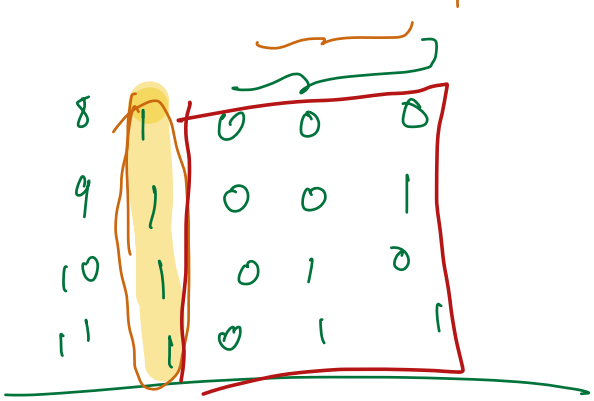
T.C : O(N x log N)



[1 → 2^x - 1]

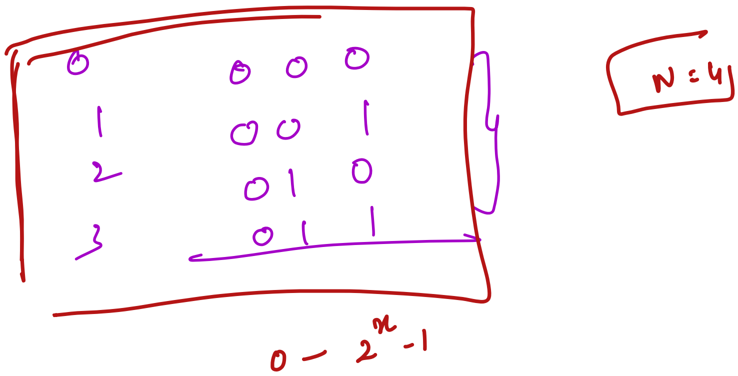
$\left(\frac{2^x}{2}\right) \times x$

⇒ 4 x 3 = 12



N = 11 → 8 → 11

12 + 4



0
1
2
3
0 0 0
0 0 1
0 1 0
0 1 1
4
N = 4
0 — $2^n - 1$