Name: Gaurav Sharma UID: 23BCS10857
, Section: 23 BCS KRG-3A , Subject : CC

_/_/_

Q→ we define $f(x,y)$ as no. of different corresponding bits in the binary representation of x and y for example $f(2,7) = 2$

given an array of n integers need to return all pairs possible for i, j, where corresponding bits are not same share sum

- Test cases:

I/p = [2, 3]   O/p = 2   explaination

$$\begin{array}{ccc} 0 & 1 & 0 \\ 0 & 1 & 1 \end{array} \qquad \begin{array}{ccc} 0 & 1 & 1 \\ 0 & 1 & 0 \end{array}$$

$1 + 1$

$= 2$ o/p

Same for $[1, 3, 5] = 8$ ← Sol^n

Solution → ~~Har element pai pairs ke bits ko compare~~
for every possible pair in the array check or compare the bits as the bits are different increase the count, complexity will be high possibly $O(n^2)$

Better solution →
Count no. of 0,s & 1's for each position of bits & then compare them.

Pseudocode:

Sum of different bits (A, n) :
~~total~~ := 0
for bit 0 to 31 :

___/___/___

Count 1 ← 0

for i = 0 to n-1 ;
    if (A[i] And (1 << bit)) ≠ 0 ;
        Count1 = Count1 + 1

    Count0 = n - Count1
    total = total (Count1 × Count0)

return 2 × total ;

Explaination :

→ Every no. is represented in binary, we process each bit position from 0 to 31

→ for a fixed bit we count How many array elements have the bit set (Count1)

→ Remaining elements already have bit unset
Count0 = (n - Count1)

→ Every pair having one '1' & one '0' contributes 1 to the answer.

→ So Contribution of one bit = Count1 × Count0
we sum this for all bits

→ Since Problem Consider ordered Pairs means (i, j) & (j, i) so we we multiply answer by 2