0, 7

23BCS 10109 7, 2 Paramjeet

we define (x, y) as number of different corresponding bits in a binary ~~search~~ representation of x and y.

→ [1, 3, 5] → 8
[2, 7] → 2

→ Brute force

→ we iterate over the loop 1 → n
→ then we iterate from i+1 to n inner loop
→ then compare all the element (i, j) then check if their bits are same or ~~not~~ count the different bits.
→ then at the end double the count then return the count.

```
int count=0
for( int i=0 , i< n-1, i++){  ~~int count=0;~~
  for( int j= i+1, j<n , j++){

        count += __builtin__popcount(i^j)

  }
}
return 2* count;
```

3 * 1 101
11,

optimal Soln

→ Take a array of 31 size that all the index represent the bits from 1→31
→ for every element in original array if the bits is set then add 1 to its corresponding index of the array bit.
→ After iterating all over the array all the bits that are set ~~in~~ the element in the array are reflecting to the array.
→ Then iterate over the ~~bits~~ freq array for all the non. zero element formula become.

$$freq[i] * (N - f[i]);$$

→ [1, 3, 5]

for 3

$3 * (3-3)$

$= 0$

→ then next

$0 + 1 * (3-1) + 1 * (3-1)$

$= 0 + 2 + 2$

$= 4$

→ Double it = 4 * 2 = 8 (Because we are counting both for (Ai, Aj) & (Aj, Ai

return the Ans 28.

time complexity = O(N) * O(31)

Space complexity = O(31).