2, 4, 6

10 0 0
11

2 ^ 2 = 4

= 2

Sambhav Mahajan 23BCS11290
KRG-2B

→ Input: A = [1, 3, 5]
Output: 8
Input: [2, 3]
Output: 2

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 since binary representation of 2 and 7 are 0 1 0 and 1 1 1, respectively. The first and third bit differs so f(2, 7) = 2.
You are given an array of N integers, A1, A2, ..., AN. Find sum of f(Ai, Aj) for all pairs i, j such that 1 ≤ i, j ≤ N. Return the answer modulo $10^9 + 7$

approach 1 (unoptimized)

just iterate two loops and add arr[i] ⊗ arr[j] count of bits

→

~~int cnt = 0~~

Pseudo Code :

```
long long cnt = 0
int n; input n.
int arr[n];
for(int i=0; i<n; i++) {
    for(int j=0; j<n; j++) {
        if(i==j) continue;
        cnt += pop_count(arr[i] ^ arr[j]);
    }
}
print cnt %((int)(1e9+7));
```

time Complexity : $O(n^2)$
Space Complexity : $O(1)$

Approach 2

for every bit, for each number, count set bit,
now for each (i, j) differing bit,
SetCount * UnSetCount is the count of differing bits pair for that bit

pseudocode: int n; int arr(n), input arr;

```
long long cnt = 0;
for(int b = 0; b < 32; b++) {
    long long k = 0;
    for(int i = 0; i < n; i++) {
        if(arr(i) & (1 << b)) {
            k++;
        }
    }
    cnt += (k)(n-k);
}
cnt *= 2;
print cnt % (int)(1e9 + 7);
```

time Complexity : O(n)
Space Complexity : O(1)