Tanmeet Kaur
23BCS11761
KRG 3A

Test cases :
input 1 : A = [1, 3, 5]
output 1 : 8
input 2 : A = [2, 3]
output 2 : 2

We define f [X, Y] as number of different corresponding bits in the binary representation of X & Y. for example, f(2,7) = 2, since the binary representation of 2 & 7 are 010 and 111, respectively. The first and the third bit differ, so f (2, 7) = 2

You are given an array of N positive integers, $A_1, A_2, \dots A_N$. Find sum of $f(A_i, A_j)$ for all pairs (i, j) such that $1 \le i, j \le N$. Return the answer modulo $10^9 + 7$.

⇒ Given
f (x, y) = no. of different bits in binary of x & y
[checking every bits → TC = (N^2) → TLE]
To solve this, we can use the concept of bit manipulation - Try checking the bits of x & y.

Let's look at a particular k^{th} bit
if K^{th} bit of x is 1 & K^{th} bit of y is 0.
or vice versa
so ans = 2 for pair (x, y) & (y, x)

↳ Let cnt 1 = nos having bit K = 1
cnt 2 = nos having bit K = 0

contribution = 2 × cnt 1 × cnt 0.
for (x, y) and (y, x)

→ Steps to solve :

For every bit (0 -> 31)

1) Count numbers having that bit set.
2) Compute pairs.
3) Add to answer.

→ Code:

```
#include <bits/stdc++.h>
using namespace std;

const long long MOD = 1e9 + 7;

long long totalPairs (vector<int> &A) {
    long long n = A.size();
    long long ans = 0;

    for (int bit = 0; bit < 32; bit ++) {
        long long cnt1 = 0
        for (int i = 0; i < n; i ++) {
            if (A[i] & (1 << bit)) {
                cnt1 ++;
            }
        }
        long long Contribution = (cnt1 * cnt0) % MOD
        contribution = (2 * contribution) % MOD

        ans = (ans + contribution) % MOD

    }
return ans;
}
```