UID: 23BCS10087

Experiment - 4.

Name: Vanshit Kumar

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

You are given an array of N positive integers, $A_1, A_2, \ldots, A_N$. 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 $10^9 + 7$.

Constraints

$1 \le N \le 10^5$

$1 \le A[i] \le 2^{31} - 1$

Appraach →

① We will create a frequency array of counting for each position in binary representation that how many ~~bits~~ number are ~~set~~ having the bit set at that position.

② We will count the sum of total such pairs ~~at~~ for bit position i,

$$sm = \sum_{i=0}^{i=31} pos[i] \times [pos[i] + N]$$

③ finally, since i, j and j, i are two different pairs, we will double the answer

$$sm = sm \times 2$$

④ It returns the final answers.

Code →

```
#include <bits/stdc++.h>
const int MOD = 1000000007;
int main() {
    int n;
    cin >> N;

    vector<int> a(n);
    for (int i=0 ; i<n ; i++)
            cin >> a[i];
    int freq[31];
    for (int i=0 ; i<n ; i++ ) {
        for (int j=0 ; j<31; j++) {
            if (a[i] & (1<<j))
                  freq[j] += 1;
        }
    }
    int ans = 0;
    for (int i=0 ; i<31 ; i++) {
        int p = freq[i] x (N-freq[i]);
        ans = (ans + p) % MOD;
    }

    ans = (ans x 2) % MOD;

    cout << ans << endl;
    return 0;
}
```

high

while (low ≤ high)

0 1 0
0 0 1
1 0 1

1 x 2
1 x 2 x 1 (i<<2)

Test case : n = 3   a = [3, 1, 5]

ans = 8