

Name: Mohammad Ashraf Khan

UID: 28BCS11123

Section: 28BCS-FS-KRG-3A

11

Q1. Test cases:

Input 1: \rightarrow Output 1: id 30ms 0 ms

$A = [2, 3, 5]$ question (f) 8 ms 0 ms

Input 2: \rightarrow Output 2: 10 ms 2 ms

$A = [2, 3]$ 2

id 0.44 ms 0 ms

Problem statement:

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 differs, so $f(2, 7) = 2$.

Question 1: Given two integers i, j such that

You are given an array of N positive integers, $A_1, A_2, A_3, \dots, A_N$. Find the sum of $f(A_i, A_j)$ for all pairs

(i, j) such that $1 \leq i, j \leq N$. Refer to the answer modulo $10^9 + 7$.

constraint:

$1 \leq N \leq 10^5$; range of each array element

$1 \leq A[i] \leq 2^{31}$ (i.e. integer values are limited)

i (0 to $N-1$) j (0 to $N-1$)

Soln.

Approach:

Count total different bits for all ordered pairs (i, j) .

as in problem statement: $f(2, 7)$:

$2 \rightarrow 010$; other bit (A) ; 1st bit ; 2nd bit

$7 \rightarrow 111$; $2 \rightarrow 0$; different ; same ; diff

Total different bits $f(2, 7) = 2$

Bitwise intuition:

Instead of comparing every pair (which would be too slow) we solve it (bit by bit).

most freight transportation; almost

8814180988 1910

AE-00931 29-23888 000014200

2020-02-11 10:30

For a fixed bit position k :

- Count + 1 = no. of having ^{bit} 1 R 8 set = A
 - Count 0 = no. of 0 having bits unset

and for this bit:

- Every pair where one number has bit=1 (set) and the other has bit=0 (unset)
 - contributes 1 to d answer.

Number of such ordered pairs:

CountL x Count0

But since (i,j) and (j,i) both counted:
 $\Rightarrow \text{contributions} = 2 \times \text{CountL} \times \text{Count0}$

C++ Implementation: It has two parts (i) & (ii)

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

```
#define MOD 1000000007; 301-33 1972
```

longlong solve (vector<int> & A) {

long long N = A.size();

long long ans = 0;

```
for (int bit = 0; bit < 31; bit++)
```

long long count = 0;

```
long long count = 0;  
for (count i = 0; i < N; i++) {
```

```
for (int i = 0; i < N; i++)
```

if (A[i] & (1LL << bit)) {

for (int i = 0; i < count; i++) {

3

3 100 100 100

long long count0 = N - count1

long long contribution = (2L

long long continuation = continuation continuation continuation

~~800: - (800-100) = 700~~

$\Delta m\% = (\Delta m\% + \text{contribution}) \% \cdot M$

The outcome is 30% lower per year.

turn around

```

int main() {
    ios::sync_with_stdio(false);
    cin.tie(NULL); int N; cin >> N;
    vector<int> A(N);
    for (int i=0; i<N; i++) {
        cin >> A[i];
    }
    cout << value(A) << "\n";
    return 0;
}

```

Time complexity :

$$O(31 \times N) = O(N)$$

dry run for ~~A[0..3]~~ A = [2, 3]

2 → 10

3 → 11

<u>bit 0</u>	2 → 10	count 1 = 1
	3 → 11	count 0 = 1

$$\begin{aligned} \text{contribution} &= 2 \times \text{count } 1 \times \text{count } 0 \\ &= 2 \times 1 \times 1 = 2 \end{aligned}$$

<u>bit 1</u>	2 → 10	count 1 = 2
	3 → 11	count 0 = 0

$$\text{contribution} = 2 \times 2 \times 0 = 0$$

Higher bits 2 to 30.

~~Count 1 = 0~~

Count 0 = ...

$$\text{contribution} = 2 \times 0 \times \dots = 0$$

$$\text{ans} = 2 + 0$$

$$= 2 \quad \checkmark$$