Exp-4

Ques → We define $f(x,y)$ as no of diff corresponding bits in binary representation of x & y. for ex – $f(2,7) = 2$ since 010 & 111. The first & 3rd bit differ, so $f(2,7)=2$.

Given arr of N +ve integers. find sum of $f(A_i, A_j)$ for all

output → A = [2,3] output: 2 $f(2,2) + f(2,3) + f(3,2) + f(3,3) = 0 + 1 + 1 + 0 = 2$

a) A = [1,3,5] output: 8 → $f(1,1) + f(1,3) + f(1,5) + f(3,1) + f(3,3) + f(3,5) + f(5,1) + f(5,3) + f(5,5)$

⇒ 8

→ Brute Force :→ 1) Try all possible pairs of A_i, A_j.
2) sum $(f(A_i, A_j))$ $\forall i,j \in [1,n]$.

Code →
```
int f(x, y) {
    int n = x^y, count=0;
    while(n!=0) {
        if(n%10==1) count++;
        n = n/10;
    }
    return count;
}

int main() {
    int n;
    cin >> n;
    vector<int> arr(n);
    for(int i=0; i<n; i++) {
        cin >> arr[i];
    }
    int ans=0;
    for(int i=0; i<n; i++) {
        for(int j=i+1; j<n; j++) {
            ans += f(i,j);
        }
    }
    cout << ans;
```

Optimized Code:-

```
#include <bits/stdc++.h>
using namespace std;    #define long long LL
int solve(vector<int> &A){
        long long MOD = 1e9+7;
        long long LL n = A.size();
          LL ans=0;
        for(int b=0; b<32; b++){
            for(int i=0; i<n; i++){
                if(A[i] &(i<<b)) one++;
            }
        }
        long long zeros= n - ones;
        ans += ((one+zeros)%MOD)*2))%MOD
    }
}
```