Name - Abhishek Sharma UID- 23BCS12330 Grp- KRG-3A

Experiment - 4

Test Case-: A = [1, 3, 5]

Q-) We define f(X, Y) as no. of different corresponding bits in binary representation of X & Y. For ex => f(2, 7) = f(2, 7) = 2 => First & third bit differ.

You are given an array of N positive integers. Find sum of f(A_i, A_j) for all pairs such that 1 ≤ i, j ≤ N.

Ans) Brute Force Approach-:

Check every pair & count no. of different bits b/w them.

- Two numbers differ at a bit position if their XOR has 1.

Steps -> i) Loop through all pairs (i<j)

ii) Compute -:

xor = A[i] ^ A[j]

iii) Count no. of set bit

iv) Add it to total answer

Code =>

```
public class Main {
    public static void main(String[] args) {
 // Ex- int[] arr = new int[]{1, 3, 5};
        int n = arr.length;
        int totcount = 0;
        for (int i = 0; i<n; i++) {
            for (int j = 0; j<n; j++) {
                if (i != j) {
                int xor = arr[i] ^ arr[j]
            .bitCount(Xor);
        totCount += cnt;
      }
   } }
return totCount;
```

T.C = O(N^2)

⇒ optimal Approach :-

⇒ For every bit position from 0 to 32 & count no. of zeros & ones.

ii) Then add count0 * count1 to tol answer.

code :-

```
long answer = 0;
for(int bit = 0 ; bit<32 ; bit++){
    long countone = 0;

    for(int i=0 ; i<N; i++){
        if(A[i] &(1<<bit)){
            countone++;
    } }
    long countZeros = N-countOne;
    answer = (answer + (countones *
                        countzeros)%
                        1e9+7)%
                        1e9+7);
}
return answer;
```

T.C = O(N * 32)

S.C = O(1)