Date |
Page |

Experiment - 04

Name: Tanay Manish Netani | Subject: competitive coding
UID: 23BCS13761 | Date: 18.2.26
Class: 23BCS_KRG-2_B

Q) We define f(x,y) as a number of different corresponding bits in the binary representation of X and Y. Eg: f(2,7) = 2, as 2 and 7 are 010 and 111, respectively. The 1st and 3rd bit differ, so f(2,7) are 2.
You are given a array of N positive integers $A_1, A_2, \ldots, A_n$. find sum of f(Ai, Aj) for all pairs i, j such that 1 ≤ i and j ≤ N, return answer modulo 10^9+7.
Input: 1st and only argument of input has single integer array A
output: return single integer denoting sum

Test cases ① arr=[1, 3, 5] O/P = 8	arr = [2,3] O/P = 2
Explanation: 1 = 01, 3 = 11, 5 = 101 Sum = 1+2+1 = 4 O/P = 4×2 = 8 (1st bit differ in 1 and 3 = 1, 1st and 2nd bit differ in 3 and 5 = 2, 3rd bit different in 1 and 5 = 1)	Explanation: 2 = 10, 3 = 11 3rd bit differ ∴ O/P = 1×2 = 2

Approach: (Vector cross product)
① run 2 loops i and j, i = 0, j = 0, n = no. of array elements (start at 0) [i: 0 to 31 - 1 (array)]
② Compare arr[j] value and (1<<i) != 0
③ find number of differing bits in arr[i] and arr[j]. (if ② is true, count1++)
~~add no. of differing bits (diff) to sum, sum += diff~~
④ j++
⑤ repeat ② to ④ till j = n, then i++, till i = 31
⑥ sum = (sum + (2 × count1 × count0) % M) % M, count0 = n - count1;
⑦ output = sum
⑧ End

Date
Page

Code:

```
Public int Solve {
  Public int four(int[] a){
     long M= 1000000007;
     int n= A.length;
     long sum = 0;
     for(int i=0; i<31; i++){
       long count1= 0;
       for(int j=0; j<n; j++){
         if((A[j] & (1<<i))!= 0){
           count1++;
         }
       }
       long count0 = n- count1;
     sum = (sum + (2^count1 * count0)%M) %M;
     }
     return (int) sum;
  }
}
```