**Sum of XOR of all pairs: -**

**Medium Accuracy: 45.14% Submissions: 19K+ Points: 4**

Given an array of **N** integers, find the **sum of xor**of all pairs of numbers in the array **arr**. In other words, select all possible pairs of **i** and **j** from **0** to **N-1 (i<j)** and determine **sum**of all **(arri xor arrj)**.

**Example 1:**

**Input :** arr[ ] = {7, 3, 5}

**Output :** 12

**Explanation:**

All possible pairs and there Xor

Value: ( 3 ^ 5 = 6 ) + (7 ^ 3 = 4)

+ ( 7 ^ 5 = 2 ) = 6 + 4 + 2 = 12

**Example 2:**

**Input :** arr[ ] = {5, 9, 7, 6}

**Output :** 47  
**Explanation:**All possible pairs and there Xor  
Value: (5 ^ 9 = 12) + (5 ^ 7 = 2)  
 + (5 ^ 6 = 3) + (9 ^ 7 = 14)  
 + (9 ^ 6 = 15) + (7 ^ 6 = 1)  
 = 12 + 2 + 3 + 14 + 15 + 1 = 47

**Your Task:**  
You do not have to take input or print output. You only need to complete the function **sumXOR()** that takes an array **(arr)**, sizeOfArray **(n)**, and **return** the **sum of xor of all pairs**of numbers in the array.

**Expected Time Complexity:** O(n).  
**Expected Auxiliary Space:** O(1).

**Constraints**  
2 ≤ n ≤ 105  
1 ≤ arr[i] ≤ 105

**Code: -**

//{ Driver Code Starts

// An efficient C++ program to compute

// sum of bitwise OR of all pairs

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

class Solution{

public:

long long int sum, c0, c1;

// Returns sum of bitwise OR

// of all pairs

long long int sumXOR(int arr[], int n)

{

//Complete the function

sum = 0;

for(int i = 0; i < 32; ++i){

c0 = 0, c1 = 0;

for(int j=0; j<n; ++j){

if(arr[j] & (1 << i)) ++c1;

else ++c0;

}

sum += (c1 \* c0) \* pow(2, i);

}

return sum;

}

};

//{ Driver Code Starts.

int main()

{

int t;

cin>>t;

while(t--)

{

int n ;

cin>>n;

int arr[n+1];

for( int i=0;i<n;i++)

cin>>arr[i];

Solution ob;

cout<<ob.sumXOR(arr, n)<<endl;

}

return 0;

}

// } Driver Code Ends

**T.C: - O(k \* n)**

**S.C: - O(1)**

**K = count of bits to store int**