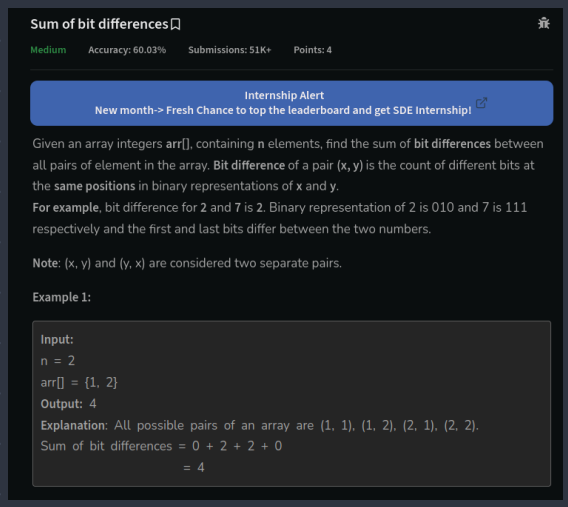
Sum of bit differences



Sum of bit differences

Medium Accuracy: 60.03% Submissions: 51K+ Points: 4

Internship Alert
New month-> Fresh Chance to top the leaderboard and get SDE Internship!

Given an array integers **arr**[], containing **n** elements, find the sum of **bit differences** between all pairs of element in the array. **Bit difference** of a pair **(x, y)** is the count of different bits at the **same positions** in binary representations of **x** and **y**.
For example, bit difference for 2 and 7 is 2. Binary representation of 2 is 010 and 7 is 111 respectively and the first and last bits differ between the two numbers.

Note: (x, y) and (y, x) are considered two separate pairs.

Example 1:

Input:
n = 2
arr[] = {1, 2}
Output: 4
Explanation: All possible pairs of an array are (1, 1), (1, 2), (2, 1), (2, 2).
Sum of bit differences = 0 + 2 + 2 + 0
= 4

→ Naive Approach :-

↳ Generate all the pairs using for loop

↳ XOR the pairs to get different element

↳ Use __builtin_popcount() to get the count of set bits.

```
class Solution{
public:

        long long sumBitDifferences(int arr[], int n) {
            long long ans = 0;

            for(int i = 0; i < n - 1; i++) {
                int j = i;
                while(j < n - 1) {
                    ans += __builtin_popcount(arr[i]^arr[j+1])*2;
                j++;
                }
            }
            return ans;
        }
};
```

Time Complexity : $O(n^2)$

Space Complexity : $O(1)$

→ This gives TLE, need $O(n)$ time complexity.

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

Constraints:
$1 <= n <= 10^5$
$1 <= arr[i] <= 10^5$

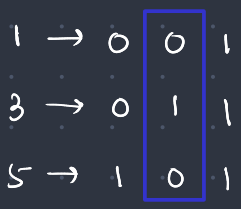
→ Optimal Approach:-

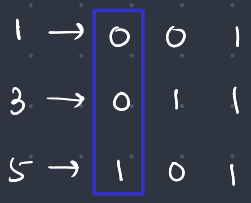
No. of zeros = 0
No. of ones = 3

Possible no. of bit difference if they make pair = 3*0 = 0

No. of zeros = 2
No. of ones = 1

Possible pairs giving bit difference = 2*1 = 2
(1,3) & (3,5)





No. of zeros = 2
No. of ones = 1

Possible pairs giving bit difference = 2
(1,5) & (3,5)

Total sum of bit difference = 2+2 = 4

∵ (x,y) & (y,x) are different

⇒ Ans = 4*2 = 8.

```
class Solution{
public:

    long long sumBitDifferences(int arr[], int n) {
        long long ans=0;
        for(int i=0;i<32;i++){
            long long  z=0;
             long long o=0;
            for(int j=0;j<n;j++){
                if((1<<i & arr[j]) ==0)z++;
                else o++;
            }
            ans+=(z*o);
        }
        return 2*ans;
    }
};
```

Time Complexity : O(N)
Space Complexity : O(1)