Problem 1442: Count Triplets That Can Form Two Arrays of Equal XOR

Problem Information

Difficulty: Medium
Acceptance Rate: 84.86%
Paid Only: No
Tags: Array, Hash Table, Math, Bit Manipulation, Prefix Sum

Problem Description

Given an array of integers `arr`.

We want to select three indices `i`, `j` and `k` where `(0 <= i < j <= k < arr.length)`.

Let's define `a` and `b` as follows:

* `a = arr[i] ^ arr[i + 1] ^ ... ^ arr[j - 1]` * `b = arr[j] ^ arr[j + 1] ^ ... ^ arr[k]`

Note that **^** denotes the **bitwise-xor** operation.

Return _the number of triplets_ (`i`, `j` and `k`) Where `a == b`.

Example 1:

Input: arr = [2,3,1,6,7] **Output:** 4 **Explanation:** The triplets are (0,1,2), (0,2,2), (2,3,4) and (2,4,4)

Example 2:

Input: arr = [1,1,1,1,1] **Output:** 10

Constraints:

* `1 <= arr.length <= 300` * `1 <= arr[i] <= 108`

Code Snippets

C++:

```
class Solution {
public:
int countTriplets(vector<int>& arr) {

}
};
```

Java:

```
class Solution {
public int countTriplets(int[] arr) {

}
}
```

Python3:

```
class Solution:
def countTriplets(self, arr: List[int]) -> int:
```