Problem List

Submit

Description | Editorial | Solutions | Submissions

# 3199. Count Triplets with Even XOR Set Bits

Solved

Premium

Easy | Topics | Companies | Hint

Given three integer arrays `a`, `b`, and `c`, return the number of triplets `(a[i], b[j], c[k])`, such that the bitwise `XOR` of the elements of each triplet has an **even** number of set bits.

**Example 1:**

**Input:** a = [1], b = [2], c = [3]

**Output:** 1

**Explanation:**

The only triplet is `(a[0], b[0], c[0])` and their `XOR` is: $1 \text{ XOR } 2 \text{ XOR } 3 = 00_2$.

**Example 2:**

**Input:** a = [1,1], b = [2,3], c = [1,5]

**Output:** 4

**Explanation:**

Consider these four triplets:

- `(a[0], b[1], c[0])`: $1 \text{ XOR } 3 \text{ XOR } 1 = 011_2$
- `(a[1], b[1], c[0])`: $1 \text{ XOR } 3 \text{ XOR } 1 = 011_2$
- `(a[0], b[0], c[1])`: $1 \text{ XOR } 2 \text{ XOR } 5 = 110_2$
- `(a[1], b[0], c[1])`: $1 \text{ XOR } 2 \text{ XOR } 5 = 110_2$

**Constraints:**

- $1 \le a.length, b.length, c.length \le 100$
- $0 \le a[i], b[i], c[i] \le 100$

Seen this question in a real interview before? 1/5

Yes No

Accepted 1,674/2K | Acceptance Rate 82.9%

Topics

Companies

Hint 1

Discussion (2)



9 2 0 Online

Code

Python3 • Auto

```python
class Solution:
    def tripletCount(self, a: List[int], b: List[int], c: List[int]) -> int:

        count = 0

        for i in range(0, len(a)):
            for j in range(0, len(b)):
                for k in range(0, len(c)):
                    temp = bin(a[i] ^ b[j] ^ c[k]).count("1")
                    if temp % 2 == 0:
                        count += 1

        return count

```

Saved Ln 12, Col 1

Testcase | Test Result

You must run your code first