Problem 1310: XOR Queries of a Subarray

Problem Information

Difficulty: Medium
Acceptance Rate: 78.22%
Paid Only: No
Tags: Array, Bit Manipulation, Prefix Sum

Problem Description

You are given an array `arr` of positive integers. You are also given the array `queries` where `queries[i] = [lefti, righti]`.

For each query `i` compute the **XOR** of elements from `lefti` to `righti` (that is, `arr[lefti] XOR arr[lefti + 1] XOR ... XOR arr[righti]`).

Return an array `answer` where `answer[i]` is the answer to the `ith` query.

Example 1:

Input: arr = [1,3,4,8], queries = [[0,1],[1,2],[0,3],[3,3]] **Output:** [2,7,14,8] **Explanation:** The binary representation of the elements in the array are: 1 = 0001 3 = 0011 4 = 0100 8 = 1000 The XOR values for queries are: [0,1] = 1 xor 3 = 2 [1,2] = 3 xor 4 = 7 [0,3] = 1 xor 3 xor 4 xor 8 = 14 [3,3] = 8

Example 2:

Input: arr = [4,8,2,10], queries = [[2,3],[1,3],[0,0],[0,3]] **Output:** [8,0,4,4]

Constraints:

* `1 <= arr.length, queries.length <= 3 * 104` * `1 <= arr[i] <= 109` * `queries[i].length == 2` * `0 <= lefti <= righti < arr.length`

Code Snippets

C++:

```
class Solution {
public:
vector<int> xorQueries(vector<int>& arr, vector<vector<int>>& queries) {

}
};
```

Java:

```
class Solution {
public int[] xorQueries(int[] arr, int[][] queries) {

}
}
```

Python3:

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