

2997. Minimum Number of Operations to Make Array XOR Equal to K

Solved ●

Medium  Topics  Companies  Hint

You are given a **0-indexed** integer array `nums` and a positive integer `k`.

You can apply the following operation on the array **any** number of times:

- Choose **any** element of the array and **flip** a bit in its **binary** representation. Flipping a bit means changing a `0` to `1` or vice versa.

Return the **minimum** number of operations required to make the bitwise `XOR` of **all** elements of the final array equal to `k`.

**Note** that you can flip leading zero bits in the binary representation of elements. For example, for the number `(101)2` you can flip the fourth bit and obtain `(1101)2`.

Example 1:

Input: `nums = [2,1,3,4]`, `k = 1`

Output: 2

Explanation: We can do the following operations:

- Choose element 2 which is `3 == (011)2`, we flip the first bit and we obtain `(010)2 == 2`. `nums` becomes `[2,1,2,4]`.
- Choose element 0 which is `2 == (010)2`, we flip the third bit and we obtain `(110)2 = 6`. `nums` becomes `[6,1,2,4]`.

The XOR of elements of the final array is `(6 XOR 1 XOR 2 XOR 4) == 1 == k`.  
It can be shown that we cannot make the XOR equal to `k` in less than 2 operations.

Example 2:

Input: `nums = [2,0,2,0]`, `k = 0`

Output: 0

Explanation: The XOR of elements of the array is `(2 XOR 0 XOR 2 XOR 0) == 0 == k`. So no operation is needed.

Constraints:

- `1 <= nums.length <= 105`
- `0 <= nums[i] <= 106`
- `0 <= k <= 106`

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

Yes No

Accepted 92.8K Submissions 107.5K Acceptance Rate 86.4%

Topics	▼
Companies	▼
Hint 1	▼
Hint 2	▼
Similar Questions	▼
Discussion (94)	▼