Bit-Manipulation

video-16

Leetcode -1442 Medium

Facebook, Instagram → codestorywithMIK

Twitter → cswithMIK

→ codestorywithMIK

1442. Count Triplets That Can Form Two Arrays of Equal XOR

Medium | Topics | Companies | Hint

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:- arr = { 2, 3, 1, 6, 7 }

Output = 4

i	j	K
0	1	2
0	2	2
2	3	4
2	4	4

1. Brute Force:

for (i)
  for (j)
    for (k)

Thought Process

i, j, K

a1, a2, a3, a4, a5

a1, a2, a3, a4

a1 ∧ a2 ∧ a3 ∧ a4 = 0

arr = { 2, 3, 1, 6, 7 }

(K − i) triplets
(2 − 0) = 2

a ∧ b = 0

1 0
1 1
0 1
0 1
0

XOR [i : j-1] = a

XOR [j : K] = b

a == b

0 1 2 3 4

{ 2 3 1 6 7 }

[2, 3, 1, 6, 7]

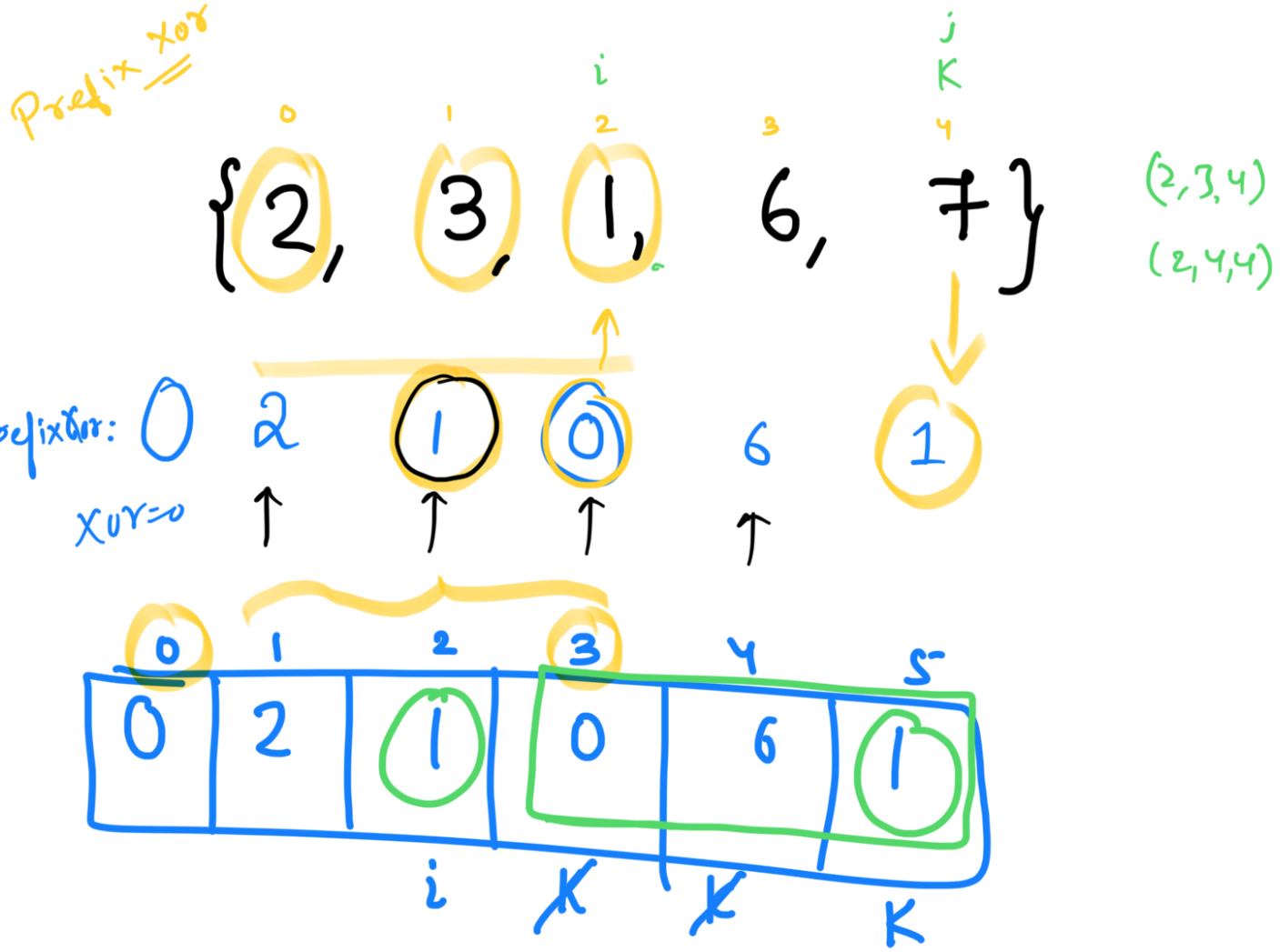
$O(n^3)$ { for (i

for (K = i+1

(i to K)

Result = (K - i) .

XOR Sum



PrefixXor[K] == PrefixXor[i]

count = K - i - 1 ;

= 3 - 0 - 1 = 2

K - i - 1 = 5 - 2 - 1

= 2

$O(n^3) \rightarrow O(n^2)$