36. Number of Subarrays with xor K

Tuesday, April 25, 2023 4:30 PM

Problem Statement Suggest Edit

Given an array 'A' consisting of 'N' integers and an integer 'B', find the number of subarrays of array 'A' whose bitwise XOR(⊕) of all elements is equal to 'B'.

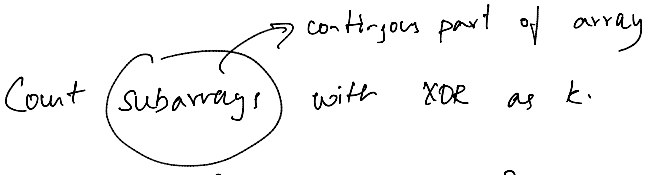
Example:

```
Input: 'N' = 4 'B' = 2
'A' = [1, 2, 3, 2]

Output: 3

Explanation: Subarrays have bitwise xor equal to '2' are: [1, 2, 3, 2],
[2], [2].
```

A subarray of an array is obtained by removing some(zero or more) elements from the front and back of the array.



arr[] = { 4 2 2 6 4 } k = 6

{4, 2} = 4 ^ 2 = 6.

{6} = 6

{2, 2, 6} = {2 ^ 2} = 0 ^ 6 = 6

{4, 2, 2, 6, 4} = {6} = 6

ans = 4

8	4	2	1
0	1	0	0 = 4
0	0	1	0 = 2
0	1	1	0 = 6

1) Brute Force:

arr[] = [4 2 2 6 4] k = 6

* Generate Subarrays

```
for (i = 0; i < n ; i++)
{
    for (int j = i ; j < n ; j++)
```

```
    for (int j=i ; j<n ; j++)
    {   XOR=0
        for (k = i ⟶ j)
            XOR = XOR ^ arr[j];
        if (XOR == k) cnt++;
    }
}
return cnt;
```

T.C ⇒ $O(n^2)$ S.C ⇒ $O(1)$

2) Better Solution :

- We don't need k - loop.

```
for (i=0; i<n ; i++)
{   XOR=0;
    for (j=i ; j<n ; j++)
    {   XR = XR^ arr[j];
        if ( XR == k) cnt++;
    }
```

}

return ans ;

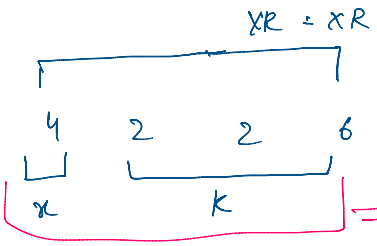
TC ⇒ $O(n^2)$ SC ⇒ $O(1)$

3) Optimal Solution

arr[] = [4 2 2 6 4] k = 6

- Every subarray has starting point and ending point

we will focus on ending point.



$4 \wedge 2 \wedge 2 \wedge 6 = 4 \wedge 6 = 2$

XR = 2 is there any subarray ending with '6' and having XOR of K.

$x \wedge k = XR$

$(x \wedge k) \wedge k = XR \wedge k$ (if XOR same element it's '0')

$x = XR \wedge k$ → this only possible when x is there whose value is $x = XR \wedge k$.

$= 2 \wedge 6$

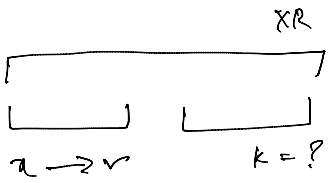
$x = 4$

So we have someone from start which is giving '4'

yes we have so now we can say that there exists some element which will give me the 'k'.

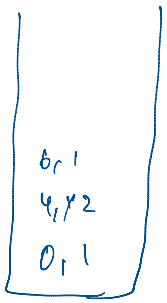
exists some element which will give me the 'k'.

we will try to move, and keep a track of pre XOR.



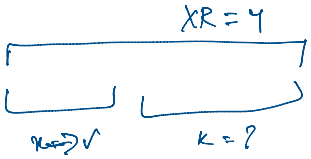
To keep a track we need hashmap dis.

arr [] = [4 2 2 6 4] k = 6



XR = 0 4, 2 (4^2) 6, 2 (6^2) = 4, 6 (6^4) = 2, 4 (2^4) = 6.

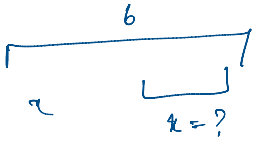
cnt = 0 1 3,



x = XR ^ k

= 4 ^ 6

= 2 (is there '2' previously ND)

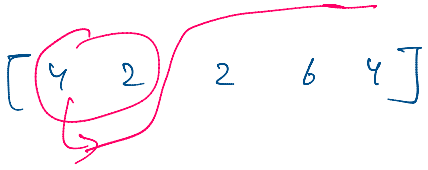


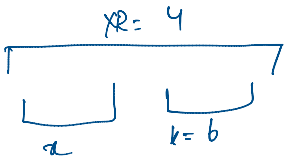
x = 6 ^ 6

x = 0 (yes value started)

so entire thing is subarray

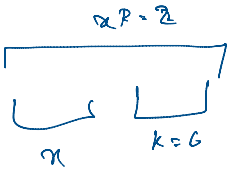
so entire thing is subarray





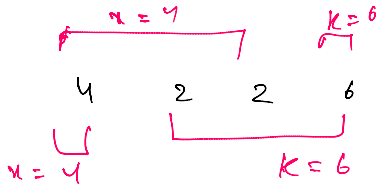
$x = 4 \wedge 6$

$x = = 2$ (No found as 2 in hashmap)

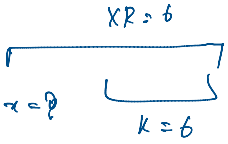


$x = 2 \wedge 6$

$= 4$ (4 is we have two element



(So two subbarray so increase cnt = 2)



$x = 6 \wedge 6 = 0$ (we have '0')

So entire array. cnt = 1

cnt = 4

T.C $\Rightarrow O(n \log n)$ S.C $\Rightarrow O(n)$

```cpp
#include<bits/stdc++.h>
int subarraysWithSumK(vector < int > a, int k) {
    int xr = 0;
    map<int, int> mpp;
    mpp[xr]++; // {0,1} is inserted at first
    int cnt = 0;
    for(int i = 0; i<a.size(); i++){
        xr = xr ^ a[i];
        // looking for k
        int x = xr ^ k;
        cnt += mpp[x];
        mpp[xr]++;
    }
    return cnt;
}
```