Assessment – 7 [Panda Village]

Editorial By – Joydeep Biswas

Difficulty:- Easy

Prerequisites:- Properties of XOR

Problem Understanding:- Let it be very simple. We are given two arrays of length N (N must be Odd). Lets consider each array as a sequence of number.A =$X_1, X_2, \ldots, X_n$.

And B = $Y_1, Y_2, \ldots, Y_n$. We need to find if there exists a sequence C which is basically any permutation of the sequence B for which $(A_1 \oplus C_1) = (A_2 \oplus C_2) = (A_3 \oplus C_3) = \ldots = (A_4 \oplus C_4)$ (i.e: Xor of each element A_i and C_i must same for any valid i).

And finally if there is any such sequence C then we have to print "YES" with the XOR of all elements of sequence C.

Solution Approach:-After understand the problem the basic (brute-force) approach that we can follow is that simply derive all permutation of sequence B and then check one by one if it matches the condition or not. But the main fact for which this approach will not work is that the number of operation and the constrains given will not satisfies. In simple word, here the N is given 1<= N <= 10^5. And we as we know in permutation with N number of elements it will cost N! operations, which is very big and will not fit within time limit. So the max permutation we can calculate safely is 10^8 (some cases it leads to 10^9).

So from here we are sure that the brute-force approach will not work for this problem. So not the question is which method or tricks or properties we should follow in order to get a A.C(All Correct).

First lets see, some basic property of XOR operator.

1. *Commutative*: A ⊕ B = B ⊕ A
   This is clear from the definition of XOR: it doesn't matter which way round you order the two inputs.

2. *Associative*: A ⊕ (B ⊕ C) = (A ⊕ B) ⊕ C
   This means that XOR operations can be chained together and the order doesn't matter. If you aren't convinced of the truth of this statement, try drawing the truth tables.

3. *Identity element*: A ⊕ 0 = A
   This means that any value XOR'd with zero is left unchanged.

4. *Self-inverse*: A ⊕ A = 0
   This means that any value XOR'd with itself gives zero.

Assessment – 7 [Panda Village]

These four properties are enough to solve this question.

General observation:- We can see the N is odd and we are talking about XOR operation in this problem, so the first thing clicks in mind is if there is a any relation between the odd number and XOR property.

If we think carefully then we can see there is pattern between this two. If we use property number 4 **self-inverse** and property number 3 **Identity element** and try to relate with question then we can see as N is odd

$(A_1 \oplus C_1) \oplus (A_2 \oplus C_2) \oplus (A_3 \oplus C_3) \oplus \ldots \oplus (A_4 \oplus C_4) = D$---------- (1) equation 1

Reason:-

((($X \oplus X) \oplus X) \oplus X) = ((0 \oplus X) \oplus X) = (X \oplus X) = 0$ [N is even]

(((($X \oplus X) \oplus X) \oplus X) \oplus X) = (((0 \oplus X) \oplus X) \oplus X) = ((X \oplus X) \oplus X) = (0 \oplus X) = X$ [N is odd]

So if you continue with this series then you can see if N is odd then ans is number it self and if N is even then ans is 0;

Now If we apply **Commutative** property then we can obtain from equation 1 that.

$(A_1 \oplus C_1) \oplus (A_2 \oplus C_2) \oplus (A_3 \oplus C_3) \oplus \ldots \oplus (A_4 \oplus C_4) = D$ this can be simplified as

$(A_1 \oplus A_2 \oplus A_3 \oplus \ldots\ldots \oplus A_n) \oplus (C_1 \oplus C_2 \oplus \ldots \oplus C_n) = D$

Now the main part begins as C is a permutation of B so we can write like this

$(A_1 \oplus A_2 \oplus A_3 \oplus \ldots\ldots \oplus A_n) \oplus (B_1 \oplus B_2 \oplus \ldots \oplus B_n) = D$ ----------(2) equation 2

No matter the what is the sequence because after all the XOR will perform with entire sequence so, which order or sequence you follow it will give the same XOR result(If we consider entire array 1 to N).

As we can see $A \oplus C = D$ we can write $C = A \oplus D$ at any valid ith instance.

We are almost done here :-

We have $C = A \oplus D$ at any valid ith instance.

And we know how to calculate D from ***equation 2*** [simply XORing all the element of array. And then from there we can calculate C_i by $A_i \oplus D$.

Assessment – 7 [Panda Village]

Now we have a new sequence C and have sequence A & B. Thus C is a permutation of B, so if all the element in B is same as C then we can say there is a sequence and we can simply XORing the elements and print it with a string “YES”.

For simplify the matching the elements we can sort both the array in ascending order or descending order and check by each element from 0 to N-1.

Original problem’s editorial :- https://discuss.codechef.com/t/xorgm-editorial/54449

Complexity :- O(n log n)

With this approach O(n log n) time takes only for sorting, otherwise it takes O(n)time for other traversing. So the total complexity will be **O(n log n)** .

Code:-

```
//
// By - Joydeep Biswas - 24/04/2020
//
//for 100 points
//
import java.io.*;
import java.util.*;
import java.lang.*;
class TestClass {
     static class FastReader
    {
        BufferedReader br;
        StringTokenizer st;

        public FastReader()
        {
            br = new BufferedReader(new
                     InputStreamReader(System.in));
        }

        String next()
        {
            while (st == null || !st.hasMoreElements())
            {
                try
                {
                    st = new StringTokenizer(br.readLine());
                }
                catch (IOException  e)
```

Assessment – 7 [Panda Village]

```
                {
                    e.printStackTrace();
                }
            }
            return st.nextToken();
        }

        int nextInt()
        {
            return Integer.parseInt(next());
        }

        long nextLong()
        {
            return Long.parseLong(next());
        }

        double nextDouble()
        {
            return Double.parseDouble(next());
        }

        String nextLine()
        {
            String str = "";
            try
            {
                str = br.readLine();
            }
            catch (IOException e)
            {
                e.printStackTrace();
            }
            return str;
        }
    }
    public static void main(String args[] ) throws Exception {
        FastReader s=new FastReader();
        int t = s.nextInt();
        while(t-->0){
            int n = s.nextInt();
            int[] x = new int[n];
            int[] y = new int[n];
            int[] p = new int[n];
            for(int i=0;i<n;i++){
                x[i] = s.nextInt();
            }
            for(int i=0;i<n;i++){
```

Assessment – 7 [Panda Village]

```
                y[i] = s.nextInt();
            }
            int d=0;
            for(int i=0;i<n;i++){
                d^=x[i]^y[i];
            }
            for(int i=0;i<n;i++){
                p[i] = x[i]^d;
            }
            Arrays.sort(y);
            Arrays.sort(p);
            int i=0;
            for(i=0;i<n;i++){
                if(y[i]!=p[i])
                    break;

            }
            if(i<n){
                System.out.print("-1"+"\n");
            }
            else{
                int temp=0;
                for(i=0;i<n;i++){
                    temp^= y[i];
                }
                System.out.print("YES "+temp+"\n");
            }
        }

    }
}
```