



XOR Subsequences

by [vlade087](#)

Problem

Submissions

Leaderboard

Discussions

Editorial

Consider an array, A , of n integers ($A = a_0, a_1, \dots, a_{n-1}$). We take all consecutive subsequences of integers from the array that satisfy the following:

$$\{a_i, a_{i+1}, \dots, a_{j-1}, a_j\}, \text{ where } 0 \leq i \leq j < n$$

For example, if $n = 3$ our subsequences will be:

1. a_0
2. a_1
3. a_2
4. a_0, a_1
5. a_1, a_2
6. a_0, a_1, a_2

For each subsequence, we apply the bitwise XOR (\oplus) operation on all the integers and record the resultant value. Since there are $n \times \frac{(n+1)}{2}$ subsequences, this will result in $n \times \frac{(n+1)}{2}$ numbers.

Given array A , find the XOR sum of every subsequence of A and determine the frequency at which each number occurs. Then print the number and its respective frequency as two space-separated values on a single line.

Input Format

The first line contains an integer, n , denoting the size of the array.

Each line i of the n subsequent lines contains a single integer describing element a_i .

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq a_i < 2^{16}$

Output Format

Print 2 space-separated integers on a single line. The first integer should be the number having the highest frequency, and the second integer should be the number's frequency (i.e., the number of times it appeared). If there are multiple numbers having maximal frequency, choose the smallest one.

Sample Input

```
4
2
1
1
3
```

Sample Output

1 3

Explanation

Let's find the XOR sum for all consecutive subsequences. We'll refer to the frequency of some number x as $f(x)$, and keep a running sum for each frequency:

1. $2 = 2$, frequencies: $f(2) = 1$
2. $1 = 1$, frequencies: $f(1) = 1$ and $f(2) = 1$
3. $1 = 1$, frequencies: $f(1) = 2$ and $f(2) = 1$
4. $3 = 3$, frequencies: $f(1) = 2$, $f(2) = 1$, and $f(3) = 1$
5. $2 \oplus 1 = 3$, frequencies: $f(1) = 2$, $f(2) = 1$, and $f(3) = 2$
6. $1 \oplus 1 = 0$, frequencies: $f(0) = 1$, $f(1) = 2$, $f(2) = 1$, and $f(3) = 2$
7. $1 \oplus 3 = 2$, frequencies: $f(0) = 1$, $f(1) = 2$, $f(2) = 2$, and $f(3) = 2$
8. $2 \oplus 1 \oplus 1 = 2$, frequencies: $f(0) = 1$, $f(1) = 2$, $f(2) = 3$, and $f(3) = 2$
9. $1 \oplus 1 \oplus 3 = 3$, frequencies: $f(0) = 1$, $f(1) = 2$, $f(2) = 3$, and $f(3) = 3$
10. $2 \oplus 1 \oplus 1 \oplus 3 = 1$, frequencies: $f(0) = 1$, $f(1) = 3$, $f(2) = 3$, and $f(3) = 3$

Our maximal frequency is **3**, and the integers **1**, **2**, and **3** all have this frequency. Because more than one integer has this frequency, we choose the smallest one, which is **1**. We then print the respective smallest number having the maximal frequency and the maximal frequency as a single line of space-separated values.

[f](#) [t](#) [in](#)

Submissions: 609

Max Score: 80

Difficulty: Advanced

Rate This Challenge:

☆☆☆☆☆

[More](#)

Current Buffer (saved locally, editable)
Java 7

```

1 import java.io.*;
2 import java.util.*;
3 import java.text.*;
4 import java.math.*;
5 import java.util.regex.*;
6
7 public class Solution {
8
9     public static void main(String[] args) {
10         /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
11     }
12 }

```

Line: 1 Col: 1

[Upload Code as File](#)
☐ Test against custom input

Run Code

Submit Code

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)