

Unique Number II

Unique Number II



Difficulty: Medium Accuracy: 36.9% Submissions: 163K+ Points: 4 Average Time: 15m

Given an array `arr[]` containing $2*n + 2$ positive numbers, out of which $2*n$ numbers exist in pairs whereas only two number occur exactly once and are **distinct**. Find the other two numbers. Return the answer in increasing order.

Examples:

Input: `arr[] = [1, 2, 3, 2, 1, 4]`
Output: `[3, 4]`
Explanation: 3 and 4 occur exactly once.

Input: `arr[] = [2, 1, 3, 2]`
Output: `[1, 3]`
Explanation: 1 and 3 occur exactly once.

Input: `arr[] = [2, 1, 3, 3]`
Output: `[1, 2]`
Explanation: 1 and 2 occur exactly once.

Constraints:

$2 \leq \text{arr.size()} \leq 10^6$
 $1 < \text{arr}[i] < 5 * 10^6$
`arr.size()` is even

[Try more examples](#)

Expected Complexities

Company Tags

Accolite Amazon Microsoft Samsung FactSet MakeMyTrip Google Qualcomm

Topic Tags

Approach:-

1. First we do XOR of all the numbers through which the XOR all the repeated numbers become zero and the remaining value will be the xor of remaining two numbers.
2. Now the problem arises how to get the two numbers from its XOR.
3. We will find the rightmost set bit & apply AND operation with all elements & divide the elements

into two parts:

- one which give 0 with the & operation & the other which doesn't. The answer will never be on the same side.

4. Then if do xor operation of the two parts I can see that I get the two numbers from two parts.

arr[] = {1, 2, 3, 2, 1, 4}

XOR of all elements
 $1 \oplus 2 \oplus 3 \oplus 2 \oplus 1 \oplus 4 = 3 \oplus 4 = 7$

$3 \Rightarrow 011$ $7 \Rightarrow 111$
 $4 \Rightarrow 100$

Right most bit set for 7
 $\downarrow (001)$

Formula: $\text{num} \& \sim \text{num}$

1	001	$\& 001 \neq 0$
2	010	$\& 001 \Rightarrow 0$
3	011	$\& 001 \neq 0$
2	010	$\& 001 \Rightarrow 0$
1	001	$\& 001 \neq 0$
4	100	$\& 001 \Rightarrow 0$

Two sets not equal to 0.

$4 \ 100 \ \& \ 001 \Rightarrow 0$

Two sets

Equal to 0	Not equal to 0
2	1
2	3
4	1
$2 \oplus 2 \oplus 4 = 4$	$1 \oplus 3 \oplus 1 = 3$

```

+ if num < 0, clear();
  if (4 > 3) {
    num.push_back(3);
    num.push_back(4);
  }
  else {
    num.push_back(4);
    num.push_back(3);
  }
  return num;

```

Note: Convert this numbers to variable & solve.

Note:- To find LSB of a number x , use the formula: -

$$LSB = n \& \sim n$$

↳ 2's complement of a number

class Solution {

public:

vector<int> singleNum(vector<int> &arr) {

public.

```
vector<int> singleNum(vector<int> &arr){
```

```
    int x = 0;
```

```
    for(int i : arr) x = x ^ i;
```

```
    int lsb = x & -x;
```

```
    vector<int> ans(2);
```

```
    int a = 0, b = 0;
```

```
    for(int i : arr){
```

```
        if(i & lsb) a = a ^ i;
```

```
        else b = b ^ i;
```

```
    }
```

```
    ans[0] = min(a, b);
```

```
    ans[1] = max(a, b);
```

```
    return ans; }
```

T.C : $O(n)$ S.C : $O(1)$