**Non Repeating Numbers :-**

Medium Accuracy: 36.9% Submissions: 120K+ Points: 4

Given an array A containing 2\*N+2 positive numbers, out of which 2\*N numbers exist in pairs whereas the other two number occur exactly once and are distinct. Find the other two numbers. Return in increasing order.

**Example 1:**

**Input:**

N = 2

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

**Output:**

3 4

**Explanation:**

3 and 4 occur exactly once.

**Example 2:**

**Input:**

N = 1

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

**Output:**

1 3

**Explanation:**

1 3 occur exactly once.

**Your Task:**  
You do not need to read or print anything. Your task is to complete the function **singleNumber()**which takes the array as input parameter and returns a list of two numbers which occur exactly once in the array. The list must be in ascending order.

**Expected Time Complexity:** O(N)  
**Expected Space Complexity:**O(1)

**Constraints:**  
1 <= length of array <= 106  
1 <= Elements in array <= 5 \* 106

**Code :-**

//{ Driver Code Starts

#include<bits/stdc++.h>

using namespace std;

// } Driver Code Ends

class Solution

{

public:

vector<int> singleNumber(vector<int> nums){

int xorval=0;

// total xor value

for(auto &item:nums)

xorval ^= item;

// to calculate rightmost set bit

int lsb = xorval & ~(xorval - 1);

int first=0, second=0;

for(auto &item:nums){

if(item & lsb)

first ^= item;

else

second ^= item;

}

return {min(first,second), max(first, second)};

}

};

//{ Driver Code Starts.

int main(){

int T;

cin >> T;

while(T--)

{

int n;

cin >> n;

vector<int> v(2 \* n + 2);

for(int i = 0; i < 2 \* n + 2; i++)

cin >> v[i];

Solution ob;

vector<int > ans = ob.singleNumber(v);

for(auto i: ans)

cout << i << " ";

cout << "\n";

}

return 0;

}

// } Driver Code Ends

**T.C :- O(N)**

**S.C :- O(1)**