

Day 18

Assignment 2: Unique Elements Identification

Given an array of integers where every element appears twice except for two, write a function that efficiently finds these two non-repeating elements using bitwise XOR operations.

A)

Introduction:

In this task, we aim to design a method to identify two non-repeating elements from an array where each element appears exactly twice except for the two unique elements. We'll employ bitwise XOR operations to efficiently achieve this.

Explanation:

Calculate XOR of all elements:

We initialize a variable xorResult to 0. Then, we iterate through all elements of the array and calculate the XOR of all elements. This step is crucial as XORing all elements will result in the XOR of the two non-repeating elements.

Find the rightmost set bit:

We find the rightmost set bit in the XOR result. This can be done by performing a bitwise AND operation between xorResult and its two's complement. The result will have only the rightmost set bit preserved.

Divide elements into two groups:

We iterate through the array again, dividing the elements into two groups based on whether the rightmost set bit is set or not. If the bit is set, the element is XORed with one group, otherwise with the other group. After this step, each group will contain one of the non-repeating elements.

Print the non-repeating elements:

Finally, we print the non-repeating elements found in the previous step.

Java Implementation:

```
public class UniqueElementsIdentifier {
```

```

public static void findNonRepeatingElements(int[] arr) {
    int xorResult = 0;
    for (int num : arr) {
        xorResult ^= num;
    }

    int rightmostSetBit = xorResult & -xorResult;

    int group1 = 0, group2 = 0;
    for (int num : arr) {
        if ((num & rightmostSetBit) != 0) {
            group1 ^= num;
        } else {
            group2 ^= num;
        }
    }

    System.out.println("Non-repeating elements: " + group1 + " " + group2);
}

public static void main(String[] args) {
    int[] arr = {4, 2, 4, 5, 2, 3, 3, 1};
    findNonRepeatingElements(arr);
}

```

This implementation provides an efficient solution to identify the two non-repeating elements in the given array using bitwise XOR operations.