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:

We're tasked with finding two unique elements in an array where every other element appears twice. We can achieve this efficiently using bitwise XOR operations.

Explanation:

Calculate XOR of all elements:

We initialize a variable xorResult to 0. Then, we traverse through the array and perform bitwise XOR of all elements with xorResult. After this step, xorResult will contain the XOR of the two unique elements.

Identify a set bit:

Find any set bit in xorResult. This bit indicates a difference between the two unique elements. We'll use this bit to partition the elements into two groups later.

Partition elements:

Traverse through the array again. For each element, check if the corresponding bit (found in the previous step) is set or not. Based on this, separate the elements into two groups: one group containing elements with the bit set, and the other with the bit unset.

XOR each group separately:

XOR all elements in each group separately. This will give us the two unique elements.

Java Implementation:

```
Package Day18;

public class UniqueElementsIdentifier {

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

            // Find any set bit in xorResult
```

```
        int rightmostSetBit = xorResult & -xorResult;

        // Partition elements into two groups based on the set bit
        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 follows the same logical steps but explains the process in a slightly different manner. Both implementations efficiently identify the two non-repeating elements using bitwise XOR operations in Java.