

2024-06-29 - Handout – Bit Manipulation

Q1. Single Number III

Link: <https://leetcode.com/problems/single-number-iii/>

Given an integer array `nums`, in which exactly two elements appear only once and all the other elements appear exactly twice. Find the two elements that appear only once. You can return the answer in **any order**.

You must write an algorithm that runs in linear runtime complexity and uses only constant extra space.

Example 1:

Input: `nums = [1,2,1,3,2,5]`

Output: `[3,5]`

Explanation: `[5, 3]` is also a valid answer.

Example 2:

Input: `nums = [-1,0]`

Output: `[-1,0]`

Q2. Pseudo-palindromic paths in a binary tree

Link: <https://leetcode.com/problems/pseudo-palindromic-paths-in-a-binary-tree>

Given a binary tree where node values are digits from 1 to 9. A path in the binary tree is said to be **pseudo-palindromic** if at least one permutation of the node values in the path is a palindrome.

*Return the number of **pseudo-palindromic** paths going from the root node to leaf nodes.*

Input: `root = [2,3,1,3,1,null,1]`

Output: 2

Explanation: The figure above represents the given binary tree. There are three paths going from the root node to leaf nodes: the red path `[2,3,3]`, the green path `[2,1,1]`, and the path `[2,3,1]`. Among these paths only red path and green path are pseudo-palindromic paths since the red path `[2,3,3]` can be rearranged in `[3,2,3]` (palindrome) and the green path `[2,1,1]` can be rearranged in `[1,2,1]` (palindrome).

Input: `root = [2,1,1,1,3,null,null,null,null,1]`

Output: 1

Explanation: The figure above represents the given binary tree. There are three paths going from the root node to leaf nodes: the green path `[2,1,1]`, the path `[2,1,3,1]`, and the path `[2,1]`. Among these paths only the green path is pseudo-palindromic since `[2,1,1]` can be rearranged in `[1,2,1]` (palindrome).

Q3. Count triplets that can form two arrays of equal XOR

Link: <https://leetcode.com/problems/count-triplets-that-can-form-two-arrays-of-equal-xor/>

Given an array of integers arr.

We want to select three indices i, j and k where $(0 \leq i < j \leq k < \text{arr.length})$.

Let's define a and b as follows:

- $a = \text{arr}[i] \oplus \text{arr}[i + 1] \oplus \dots \oplus \text{arr}[j - 1]$
- $b = \text{arr}[j] \oplus \text{arr}[j + 1] \oplus \dots \oplus \text{arr}[k]$

Note that \oplus denotes the **bitwise-xor** operation.

Return *the number of triplets* (i, j and k) Where $a == b$.

Example 1:

Input: arr = [2,3,1,6,7]

Output: 4

Explanation: The triplets are (0,1,2), (0,2,2), (2,3,4) and (2,4,4)

Example 2:

Input: arr = [1,1,1,1,1]

Output: 10

Q4. Maximum XOR of two numbers in an array

Link: <https://leetcode.com/problems/maximum-xor-of-two-numbers-in-an-array>

Given an integer array nums, return *the maximum result of* $\text{nums}[i] \oplus \text{nums}[j]$, where $0 \leq i \leq j < n$.

Example 1:

Input: nums = [3,10,5,25,2,8]

Output: 28

Explanation: The maximum result is $5 \oplus 25 = 28$.

Example 2:

Input: nums = [14,70,53,83,49,91,36,80,92,51,66,70]

Output: 127.