Problem 3309: Maximum Possible Number by Binary Concatenation

Problem Information

Difficulty: Medium
Acceptance Rate: 65.17%
Paid Only: No
Tags: Array, Bit Manipulation, Enumeration

Problem Description

You are given an array of integers `nums` of size 3.

Return the **maximum** possible number whose _binary representation_ can be formed by **concatenating** the _binary representation_ of **all** elements in `nums` in some order.

Note that the binary representation of any number _does not_ contain leading zeros.

Example 1:

Input: nums = [1,2,3]

Output: 30

Explanation:

Concatenate the numbers in the order `[3, 1, 2]` to get the result `"11110"`, which is the binary representation of 30.

Example 2:

Input: nums = [2,8,16]

Output: 1296

Explanation:

Concatenate the numbers in the order `[2, 8, 16]` to get the result `"10100010000"`, which is the binary representation of 1296.

Constraints:

* `nums.length == 3` * `1 <= nums[i] <= 127`

Code Snippets

C++:

```
class Solution {
public:
int maxGoodNumber(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int maxGoodNumber(int[] nums) {

}
}
```

Python3:

```
class Solution:
def maxGoodNumber(self, nums: List[int]) -> int:
```