

# Problem 3644: Maximum K to Sort a Permutation

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 34.59%

**Paid Only:** No

**Tags:** Array, Bit Manipulation

## Problem Description

You are given an integer array `nums` of length `n`, where `nums` is a **permutation** of the numbers in the range `[0..n - 1]`.

You may swap elements at indices `i` and `j` **only if** `nums[i] AND nums[j] == k`, where `AND` denotes the bitwise AND operation and `k` is a **non-negative** integer.

Return the **maximum** value of `k` such that the array can be sorted in **non-decreasing** order using any number of such swaps. If `nums` is already sorted, return 0.

**Example 1:**

**Input:** `nums = [0,3,2,1]`

**Output:** 1

**Explanation:**

Choose `k = 1`. Swapping `nums[1] = 3` and `nums[3] = 1` is allowed since `nums[1] AND nums[3] == 1`, resulting in a sorted permutation: `[0, 1, 2, 3]`.

**Example 2:**

**Input:** `nums = [0,1,3,2]`

**\*\*Output:\*\*** 2

**\*\*Explanation:\*\***

Choose `k = 2`. Swapping `nums[2] = 3` and `nums[3] = 2` is allowed since `nums[2] AND nums[3] == 2`, resulting in a sorted permutation: `[0, 1, 2, 3]`.

**\*\*Example 3:\*\***

**\*\*Input:\*\*** nums = [3,2,1,0]

**\*\*Output:\*\*** 0

**\*\*Explanation:\*\***

Only `k = 0` allows sorting since no greater `k` allows the required swaps where `nums[i] AND nums[j] == k`.

**\*\*Constraints:\*\***

\* `1 <= n == nums.length <= 105` \* `0 <= nums[i] <= n - 1` \* `nums` is a permutation of integers from `0` to `n - 1`.

## Code Snippets

**C++:**

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

    }

};
```

**Java:**

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

```
}  
}
```

### Python3:

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