You are given a **0-indexed** array of **positive** integers nums.

In one **operation**, you can swap any two **adjacent** elements if they have the **same** number of

set bits

. You are allowed to do this operation **any** number of times (**including zero**).

Return true *if you can sort the array, else return* false.

**Example 1:**

Input: nums = [8,4,2,30,15]  
Output: true  
Explanation: Let's look at the binary representation of every element. The numbers 2, 4, and 8 have one set bit each with binary representation "10", "100", and "1000" respectively. The numbers 15 and 30 have four set bits each with binary representation "1111" and "11110".  
We can sort the array using 4 operations:  
- Swap nums[0] with nums[1]. This operation is valid because 8 and 4 have one set bit each. The array becomes [4,8,2,30,15].  
- Swap nums[1] with nums[2]. This operation is valid because 8 and 2 have one set bit each. The array becomes [4,2,8,30,15].  
- Swap nums[0] with nums[1]. This operation is valid because 4 and 2 have one set bit each. The array becomes [2,4,8,30,15].  
- Swap nums[3] with nums[4]. This operation is valid because 30 and 15 have four set bits each. The array becomes [2,4,8,15,30].  
The array has become sorted, hence we return true.  
Note that there may be other sequences of operations which also sort the array.

**Example 2:**

Input: nums = [1,2,3,4,5]  
Output: true  
Explanation: The array is already sorted, hence we return true.

**Example 3:**

Input: nums = [3,16,8,4,2]  
Output: false  
Explanation: It can be shown that it is not possible to sort the input array using any number of operations.

**Constraints:**

* 1 <= nums.length <= 100
* 1 <= nums[i] <= 28