Problem 1998: GCD Sort of an Array

Problem Information

Difficulty: Hard
Acceptance Rate: 48.01%
Paid Only: No
Tags: Array, Math, Union Find, Sorting, Number Theory

Problem Description

You are given an integer array `nums`, and you can perform the following operation **any** number of times on `nums`:

* Swap the positions of two elements `nums[i]` and `nums[j]` if `gcd(nums[i], nums[j]) > 1` where `gcd(nums[i], nums[j])` is the **greatest common divisor** of `nums[i]` and `nums[j]`.

Return `true` _if it is possible to sort_`nums` _in**non-decreasing** order using the above swap method, or _`false` _otherwise._

Example 1:

Input: nums = [7,21,3] **Output:** true **Explanation:** We can sort [7,21,3] by performing the following operations: - Swap 7 and 21 because gcd(7,21) = 7. nums = [_**21**_ ,_**7**_ ,3] - Swap 21 and 3 because gcd(21,3) = 3. nums = [_**3**_ ,7,_**21**_]

Example 2:

Input: nums = [5,2,6,2] **Output:** false **Explanation:** It is impossible to sort the array because 5 cannot be swapped with any other element.

Example 3:

Input: nums = [10,5,9,3,15] **Output:** true We can sort [10,5,9,3,15] by performing the following operations: - Swap 10 and 15 because gcd(10,15) = 5. nums = [_**15**_ ,5,9,3,_**10**_] - Swap 15 and 3 because gcd(15,3) = 3. nums = [_**3**_ ,5,9,_**15**_ ,10] - Swap 10 and 15 because gcd(10,15) = 5. nums = [3,5,9,_**10**_ ,_**15**_]

Constraints:

* `1 <= nums.length <= 3 * 104` * `2 <= nums[i] <= 105`

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public boolean gcdSort(int[] nums) {

}
}
```

Python3:

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