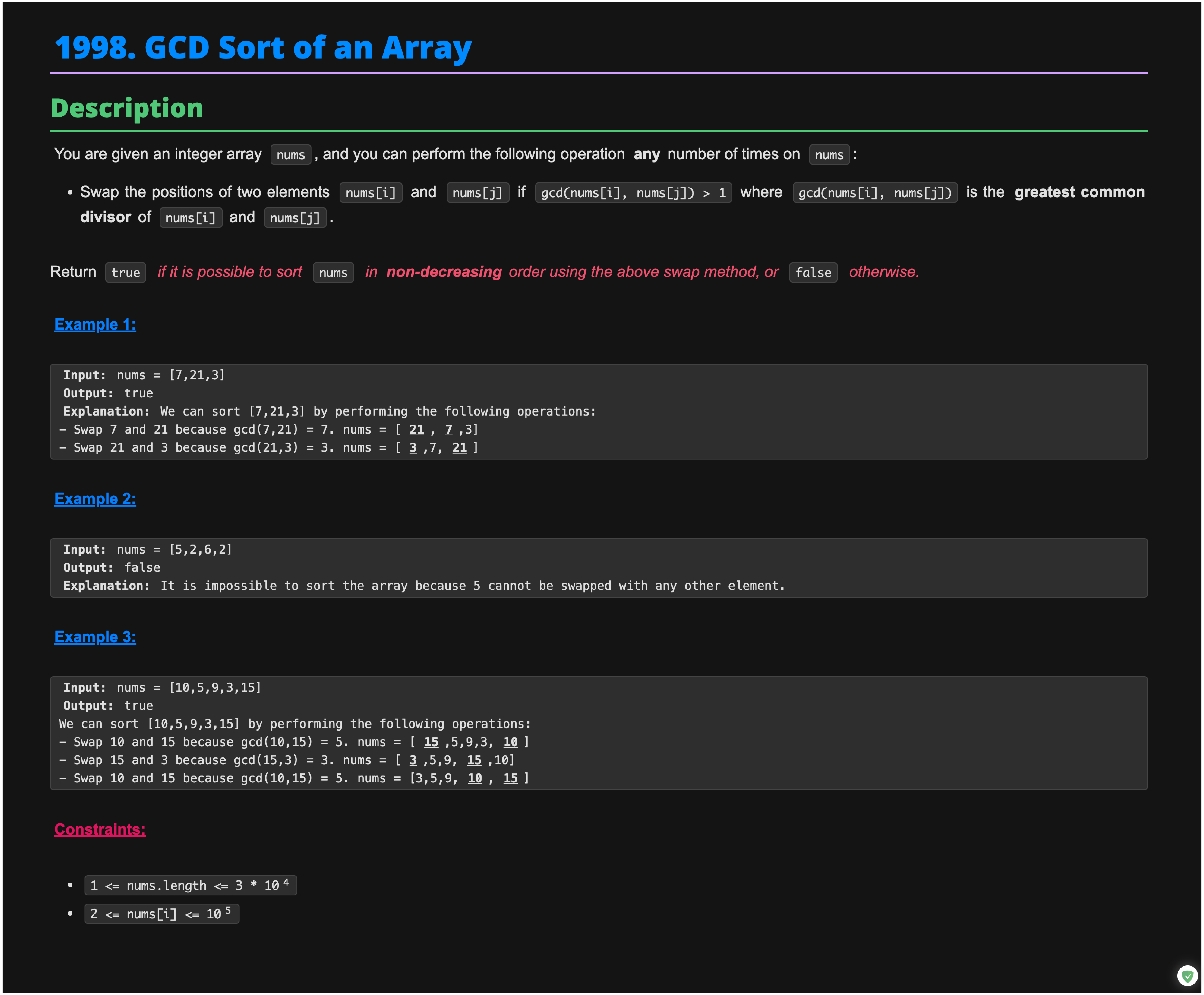

1998. GCD Sort of an Array

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 * 10`^{4}
- `2 <= nums[i] <= 10`^{5}