5866. GCD Sort of an Array

My Submissions (/contest/weekly-contest-257/problems/gcd-sort-of-an-array/submissions/)

Back to Contest (/contest/weekly-contest-257/)

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.*

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Hard

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`

JavaScript

```
function DJSet(n) {
    let parent = [];
    for (let i = 0; i < n; i++) parent[i] = i;
    let rank = Array(n).fill(0);
    return { find, union, equiv, getParent }
    function find(x) {
        return x == parent[x] ? x : parent[x] = find(parent[x]);
    }
    function union(x, y) {
        let rx = find(x), ry = find(y);
        if (rank[rx] < rank[ry]) {
```

```
            parent[rx] = ry;
        } else if (rank[rx] > rank[ry]) {
            parent[ry] = rx;
        } else {
            parent[ry] = rx;
            rank[rx]++;
        }
        return rx == ry;
    }
    function equiv(x, y) {
        return find(x) == find(y);
    }
    function getParent() {
        return parent;
    }
}

const N = 1e5;
const gcdSort = (nums) => {
    let sa = [...nums];
    sa.sort((x, y) => x - y);
    let n = nums.length;
    let a = Array(N + 1).fill(0);
    let visit = Array(N + 1).fill(0);
    for (const x of nums) a[x] = 1;
    let ds = new DJSet(N);
    for (let i = 2; i <= N; i++) {
        if (visit[i] == 0) {
            for (let j = i; j <= N; j += i) {
                visit[j] = 1;
                if (a[j] > 0) ds.union(i, j);
            }
        }
    }
    for (let i = 0; i < n; i++) {
        if (!ds.equiv(nums[i], sa[i])) return false;
    }
    return true;
};
```

☐ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/549724828/) More Details > (/submissions/detail/549724828/)

Share your acceptance!