Problem 3336: Find the Number of Subsequences With Equal GCD

Problem Information

Difficulty: Hard
Acceptance Rate: 0.00%
Paid Only: No

Problem Description

You are given an integer array

nums

.

Your task is to find the number of pairs of

non-empty

subsequences

(seq1, seq2)

of

nums

that satisfy the following conditions:

The subsequences

seq1

and

seq2

are

disjoint

, meaning

no index

of

nums

is common between them.

The

GCD

of the elements of

seq1

is equal to the GCD of the elements of

seq2

.

Return the total number of such pairs.

Since the answer may be very large, return it

modulo

10

9

+ 7

.

Example 1:

Input:

nums = [1,2,3,4]

Output:

10

Explanation:

The subsequence pairs which have the GCD of their elements equal to 1 are:

([

1

, 2, 3, 4], [1,

2

,

3

, 4])

([

1

, 2, 3, 4], [1,

2

,

3

,

4

])

([

1

, 2, 3, 4], [1, 2,

3

,

4

])

([

1

,

2

, 3, 4], [1, 2,

3

,

4

])

([

1

, 2, 3,

4

], [1,

2

,

3

, 4])

([1,

2

,

3

, 4], [

1

, 2, 3, 4])

([1,

2

,

3

, 4], [

1

, 2, 3,

4

])

([1,

2

,

3

,

4

], [

1

, 2, 3, 4])

([1, 2,

3

,

4

], [

1

, 2, 3, 4])

([1, 2,

3

,

4

], [

1

,

2

, 3, 4])

Example 2:

Input:

nums = [10,20,30]

Output:

2

Explanation:

The subsequence pairs which have the GCD of their elements equal to 10 are:

([

10

, 20, 30], [10,

20

,

30

])

([10,

20

,

30

], [

10

, 20, 30])

Example 3:

Input:

nums = [1,1,1,1]

Output:

50

Constraints:

1 <= nums.length <= 200

1 <= nums[i] <= 200

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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

Python:

```
class Solution(object):
def subsequencePairCount(self, nums):
"""
:type nums: List[int]
:rtype: int
"""
```

JavaScript:

```
/**
* @param {number[]} nums
* @return {number}
*/
var subsequencePairCount = function(nums) {

};
```

TypeScript:

```
function subsequencePairCount(nums: number[]): number {

};
```

C#:

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

}
}
```

C:

```
int subsequencePairCount(int* nums, int numsSize) {

}
```

Go:

```
func subsequencePairCount(nums []int) int {

}
```

Kotlin:

```
class Solution {
fun subsequencePairCount(nums: IntArray): Int {

}
}
```

Swift:

```
class Solution {
func subsequencePairCount(_ nums: [Int]) -> Int {

}
}
```

Rust:

```
impl Solution {
pub fn subsequence_pair_count(nums: Vec<i32>) -> i32 {

}
}
```

Ruby:

```
# @param {Integer[]} nums
# @return {Integer}
def subsequence_pair_count(nums)

end
```

PHP:

```
class Solution {

/**
* @param Integer[] $nums
* @return Integer
*/
function subsequencePairCount($nums) {

}
}
```

Dart:

```
class Solution {
int subsequencePairCount(List<int> nums) {

}
```

```
}
```

Scala:

```
object Solution {
def subsequencePairCount(nums: Array[Int]): Int = {

}
}
```

Elixir:

```
defmodule Solution do
@spec subsequence_pair_count(nums :: [integer]) :: integer
def subsequence_pair_count(nums) do

end
end
```

Erlang:

```
-spec subsequence_pair_count(Nums :: [integer()]) -> integer().
subsequence_pair_count(Nums) ->
.
```

Racket:

```
(define/contract (subsequence-pair-count nums)
(-> (listof exact-integer?) exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Find the Number of Subsequences With Equal GCD
* Difficulty: Hard
* Tags: array, dp, math
*
```

```
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) or O(n * m) for DP table
*/

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

}
};
```

Java Solution:

```
/**
* Problem: Find the Number of Subsequences With Equal GCD
* Difficulty: Hard
* Tags: array, dp, math
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) or O(n * m) for DP table
*/

class Solution {
public int subsequencePairCount(int[] nums) {

}
}
```

Python3 Solution:

```
"""
Problem: Find the Number of Subsequences With Equal GCD
Difficulty: Hard
Tags: array, dp, math

Approach: Use two pointers or sliding window technique
Time Complexity: O(n) or O(n log n)
Space Complexity: O(n) or O(n * m) for DP table
"""
```

```
class Solution:
def subsequencePairCount(self, nums: List[int]) -> int:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def subsequencePairCount(self, nums):
"""
:type nums: List[int]
:rtype: int
"""
```

JavaScript Solution:

```
/**
* Problem: Find the Number of Subsequences With Equal GCD
* Difficulty: Hard
* Tags: array, dp, math
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) or O(n * m) for DP table
*/

/**
* @param {number[]} nums
* @return {number}
*/
var subsequencePairCount = function(nums) {

};
```

TypeScript Solution:

```
/**
* Problem: Find the Number of Subsequences With Equal GCD
* Difficulty: Hard
* Tags: array, dp, math
```

```
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) or O(n * m) for DP table
*/

function subsequencePairCount(nums: number[]): number {

};
```

C# Solution:

```
/*
* Problem: Find the Number of Subsequences With Equal GCD
* Difficulty: Hard
* Tags: array, dp, math
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) or O(n * m) for DP table
*/

public class Solution {
public int SubsequencePairCount(int[] nums) {

}
}
```

C Solution:

```
/*
* Problem: Find the Number of Subsequences With Equal GCD
* Difficulty: Hard
* Tags: array, dp, math
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) or O(n * m) for DP table
*/

int subsequencePairCount(int* nums, int numsSize) {
```

```
}
```

Go Solution:

```
// Problem: Find the Number of Subsequences With Equal GCD
// Difficulty: Hard
// Tags: array, dp, math
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(n) or O(n * m) for DP table

func subsequencePairCount(nums []int) int {

}
```

Kotlin Solution:

```
class Solution {
fun subsequencePairCount(nums: IntArray): Int {

}
}
```

Swift Solution:

```
class Solution {
func subsequencePairCount(_ nums: [Int]) -> Int {

}
}
```

Rust Solution:

```
// Problem: Find the Number of Subsequences With Equal GCD
// Difficulty: Hard
// Tags: array, dp, math
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
```

```
// Space Complexity: O(n) or O(n * m) for DP table

impl Solution {
pub fn subsequence_pair_count(nums: Vec<i32>) -> i32 {

}
}
```

Ruby Solution:

```
# @param {Integer[]} nums
# @return {Integer}
def subsequence_pair_count(nums)

end
```

PHP Solution:

```
class Solution {

/**
* @param Integer[] $nums
* @return Integer
*/
function subsequencePairCount($nums) {

}
}
```

Dart Solution:

```
class Solution {
int subsequencePairCount(List<int> nums) {

}
}
```

Scala Solution:

```
object Solution {
def subsequencePairCount(nums: Array[Int]): Int = {
```

```
}
}
```

Elixir Solution:

```
defmodule Solution do
@spec subsequence_pair_count(nums :: [integer]) :: integer
def subsequence_pair_count(nums) do

end
end
```

Erlang Solution:

```
-spec subsequence_pair_count(Nums :: [integer()]) -> integer().
subsequence_pair_count(Nums) ->
.
```

Racket Solution:

```
(define/contract (subsequence-pair-count nums)
(-> (listof exact-integer?) exact-integer?)
)
```