Problem 2576: Find the Maximum Number of Marked Indices

Problem Information

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

Problem Description

You are given a

0-indexed

integer array

nums

.

Initially, all of the indices are unmarked. You are allowed to make this operation any number of times:

Pick two

different unmarked

indices

i

and

j

such that

2 * nums[i] <= nums[j]

, then mark

i

and

j

.

Return

the maximum possible number of marked indices in

nums

using the above operation any number of times

.

Example 1:

Input:

nums = [3,5,2,4]

Output:

2

Explanation:

In the first operation: pick i = 2 and j = 1, the operation is allowed because 2 * nums[2] <= nums[1]. Then mark index 2 and 1. It can be shown that there's no other valid operation so the answer is 2.

Example 2:

Input:

nums = [9,2,5,4]

Output:

4

Explanation:

In the first operation: pick i = 3 and j = 0, the operation is allowed because 2 * nums[3] <= nums[0]. Then mark index 3 and 0. In the second operation: pick i = 1 and j = 2, the operation is allowed because 2 * nums[1] <= nums[2]. Then mark index 1 and 2. Since there is no other operation, the answer is 4.

Example 3:

Input:

nums = [7,6,8]

Output:

0

Explanation:

There is no valid operation to do, so the answer is 0.

Constraints:

1 <= nums.length <= 10

5

1 <= nums[i] <= 10

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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

Python:

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

JavaScript:

```
/**
* @param {number[]} nums
* @return {number}
*/
```

```
var maxNumOfMarkedIndices = function(nums) {

};
```

TypeScript:

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

};
```

C#:

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

}
}
```

C:

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

}
```

Go:

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

}
```

Kotlin:

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

}
}
```

Swift:

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

```
}
}
```

Rust:

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

}
}
```

Ruby:

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

end
```

PHP:

```
class Solution {

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

}
}
```

Dart:

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

}
}
```

Scala:

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

}
}
```

Elixir:

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

end
end
```

Erlang:

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

Racket:

```
(define/contract (max-num-of-marked-indices nums)
(-> (listof exact-integer?) exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Find the Maximum Number of Marked Indices
* Difficulty: Medium
* Tags: array, greedy, sort, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/
```

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

}
};
```

Java Solution:

```
/**
* Problem: Find the Maximum Number of Marked Indices
* Difficulty: Medium
* Tags: array, greedy, sort, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

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

}
}
```

Python3 Solution:

```
"""
Problem: Find the Maximum Number of Marked Indices
Difficulty: Medium
Tags: array, greedy, sort, search

Approach: Use two pointers or sliding window technique
Time Complexity: O(n) or O(n log n)
Space Complexity: O(1) to O(n) depending on approach
"""

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

Python Solution:

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

JavaScript Solution:

```
/**
* Problem: Find the Maximum Number of Marked Indices
* Difficulty: Medium
* Tags: array, greedy, sort, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

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

};
```

TypeScript Solution:

```
/**
* Problem: Find the Maximum Number of Marked Indices
* Difficulty: Medium
* Tags: array, greedy, sort, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

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

```
};
```

C# Solution:

```
/*
* Problem: Find the Maximum Number of Marked Indices
* Difficulty: Medium
* Tags: array, greedy, sort, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

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

}
}
```

C Solution:

```
/*
* Problem: Find the Maximum Number of Marked Indices
* Difficulty: Medium
* Tags: array, greedy, sort, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

int maxNumOfMarkedIndices(int* nums, int numsSize) {

}
```

Go Solution:

```
// Problem: Find the Maximum Number of Marked Indices
// Difficulty: Medium
```

```
// Tags: array, greedy, sort, search
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

func maxNumOfMarkedIndices(nums []int) int {

}
```

Kotlin Solution:

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

}
}
```

Swift Solution:

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

}
}
```

Rust Solution:

```
// Problem: Find the Maximum Number of Marked Indices
// Difficulty: Medium
// Tags: array, greedy, sort, search
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

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

}
}
```

Ruby Solution:

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

end
```

PHP Solution:

```
class Solution {

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

}
}
```

Dart Solution:

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

}
}
```

Scala Solution:

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

}
}
```

Elixir Solution:

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

```
end
end
```

Erlang Solution:

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

Racket Solution:

```
(define/contract (max-num-of-marked-indices nums)
(-> (listof exact-integer?) exact-integer?)
)
```