Problem 3285: Find Indices of Stable Mountains

Problem Information

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

Problem Description

There are

n

mountains in a row, and each mountain has a height. You are given an integer array

height

where

height[i]

represents the height of mountain

i

, and an integer

threshold

.

A mountain is called

stable

if the mountain just before it (

if it exists

) has a height

strictly greater

than

threshold

.

Note

that mountain 0 is

not

stable.

Return an array containing the indices of

all

stable

mountains in

any

order.

Example 1:

Input:

height = [1,2,3,4,5], threshold = 2

Output:

[3,4]

Explanation:

Mountain 3 is stable because

height[2] == 3

is greater than

threshold == 2

.

Mountain 4 is stable because

height[3] == 4

is greater than

threshold == 2

.

Example 2:

Input:

height = [10,1,10,1,10], threshold = 3

Output:

[1,3]

Example 3:

Input:

height = [10,1,10,1,10], threshold = 10

Output:

[]

Constraints:

2 <= n == height.length <= 100

1 <= height[i] <= 100

1 <= threshold <= 100

Code Snippets

C++:

```
class Solution {
public:
vector<int> stableMountains(vector<int>& height, int threshold) {

}
};
```

Java:

```
class Solution {
public List<Integer> stableMountains(int[] height, int threshold) {

}
}
```

Python3:

```
class Solution:
def stableMountains(self, height: List[int], threshold: int) -> List[int]:
```

Python:

```
class Solution(object):
def stableMountains(self, height, threshold):
"""
:type height: List[int]
:type threshold: int
:rtype: List[int]
"""
```

JavaScript:

```
/**
* @param {number[]} height
* @param {number} threshold
* @return {number[]}
*/
var stableMountains = function(height, threshold) {

};
```

TypeScript:

```
function stableMountains(height: number[], threshold: number): number[] {

};
```

C#:

```
public class Solution {
public IList<int> StableMountains(int[] height, int threshold) {

}
}
```

C:

```
/**
* Note: The returned array must be malloced, assume caller calls free().
*/
int* stableMountains(int* height, int heightSize, int threshold, int*
returnSize) {

}
```

Go:

```
func stableMountains(height []int, threshold int) []int {

}
```

Kotlin:

```
class Solution {
fun stableMountains(height: IntArray, threshold: Int): List<Int> {

}
}
```

Swift:

```
class Solution {
func stableMountains(_ height: [Int], _ threshold: Int) -> [Int] {

}
}
```

Rust:

```
impl Solution {
pub fn stable_mountains(height: Vec<i32>, threshold: i32) -> Vec<i32> {

}
}
```

Ruby:

```
# @param {Integer[]} height
# @param {Integer} threshold
# @return {Integer[]}
def stable_mountains(height, threshold)

end
```

PHP:

```
class Solution {

/**
* @param Integer[] $height
* @param Integer $threshold
* @return Integer[]
*/
function stableMountains($height, $threshold) {

}
}
```

Dart:

```
class Solution {
List<int> stableMountains(List<int> height, int threshold) {

}
}
```

Scala:

```
object Solution {
def stableMountains(height: Array[Int], threshold: Int): List[Int] = {

}
}
```

Elixir:

```
defmodule Solution do
@spec stable_mountains(height :: [integer], threshold :: integer) ::
[integer]
def stable_mountains(height, threshold) do

end
end
```

Erlang:

```
-spec stable_mountains(Height :: [integer()], Threshold :: integer()) ->
[integer()].
stable_mountains(Height, Threshold) ->
```

```
.
```

Racket:

```
(define/contract (stable-mountains height threshold)
(-> (listof exact-integer?) exact-integer? (listof exact-integer?))
)
```

Solutions

C++ Solution:

```
/*
* Problem: Find Indices of Stable Mountains
* Difficulty: Easy
* Tags: array
*
* 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:
vector<int> stableMountains(vector<int>& height, int threshold) {

}
};
```

Java Solution:

```
/**
* Problem: Find Indices of Stable Mountains
* Difficulty: Easy
* Tags: array
*
* 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 List<Integer> stableMountains(int[] height, int threshold) {

}
}
```

Python3 Solution:

```
"""
Problem: Find Indices of Stable Mountains
Difficulty: Easy
Tags: array

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 stableMountains(self, height: List[int], threshold: int) -> List[int]:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def stableMountains(self, height, threshold):
"""
:type height: List[int]
:type threshold: int
:rtype: List[int]
"""
```

JavaScript Solution:

```
/**
* Problem: Find Indices of Stable Mountains
* Difficulty: Easy
* Tags: array
*
```

```
* 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[]} height
* @param {number} threshold
* @return {number[]}
*/
var stableMountains = function(height, threshold) {

};
```

TypeScript Solution:

```
/**
* Problem: Find Indices of Stable Mountains
* Difficulty: Easy
* Tags: array
*
* 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 stableMountains(height: number[], threshold: number): number[] {

};
```

C# Solution:

```
/*
* Problem: Find Indices of Stable Mountains
* Difficulty: Easy
* Tags: array
*
* 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 IList<int> StableMountains(int[] height, int threshold) {

}
}
```

C Solution:

```
/*
* Problem: Find Indices of Stable Mountains
* Difficulty: Easy
* Tags: array
*
* 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
*/

/**
* Note: The returned array must be malloced, assume caller calls free().
*/
int* stableMountains(int* height, int heightSize, int threshold, int*
returnSize) {

}
```

Go Solution:

```
// Problem: Find Indices of Stable Mountains
// Difficulty: Easy
// Tags: array
//
// 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 stableMountains(height []int, threshold int) []int {

}
```

Kotlin Solution:

```
class Solution {
fun stableMountains(height: IntArray, threshold: Int): List<Int> {

}
}
```

Swift Solution:

```
class Solution {
func stableMountains(_ height: [Int], _ threshold: Int) -> [Int] {

}
}
```

Rust Solution:

```
// Problem: Find Indices of Stable Mountains
// Difficulty: Easy
// Tags: array
//
// 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 stable_mountains(height: Vec<i32>, threshold: i32) -> Vec<i32> {

}
}
```

Ruby Solution:

```
# @param {Integer[]} height
# @param {Integer} threshold
# @return {Integer[]}
def stable_mountains(height, threshold)

end
```

PHP Solution:

```
class Solution {

/**
* @param Integer[] $height
* @param Integer $threshold
* @return Integer[]
*/
function stableMountains($height, $threshold) {

}
}
```

Dart Solution:

```
class Solution {
List<int> stableMountains(List<int> height, int threshold) {

}
}
```

Scala Solution:

```
object Solution {
def stableMountains(height: Array[Int], threshold: Int): List[Int] = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec stable_mountains(height :: [integer], threshold :: integer) ::
[integer]
def stable_mountains(height, threshold) do

end
end
```

Erlang Solution:

```
-spec stable_mountains(Height :: [integer()], Threshold :: integer()) ->
[integer()].
```

```
stable_mountains(Height, Threshold) ->
.
```

Racket Solution:

```
(define/contract (stable-mountains height threshold)
(-> (listof exact-integer?) exact-integer? (listof exact-integer?))
)
```