Problem 1095: Find in Mountain Array

Problem Information

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

Problem Description

(This problem is an

interactive problem

.)

You may recall that an array

arr

is a

mountain array

if and only if:

arr.length >= 3

There exists some

i

with

0 < i < arr.length - 1

such that:

arr[0] < arr[1] < ... < arr[i - 1] < arr[i]

arr[i] > arr[i + 1] > ... > arr[arr.length - 1]

Given a mountain array

mountainArr

, return the

minimum

index

such that

mountainArr.get(index) == target

. If such an

index

does not exist, return

-1

.

You cannot access the mountain array directly.

You may only access the array using a

MountainArray

interface:

MountainArray.get(k)

returns the element of the array at index

k

(0-indexed).

MountainArray.length()

returns the length of the array.

Submissions making more than

100

calls to

MountainArray.get

will be judged

Wrong Answer

. Also, any solutions that attempt to circumvent the judge will result in disqualification.

Example 1:

Input:

mountainArr = [1,2,3,4,5,3,1], target = 3

Output:

2

Explanation:

3 exists in the array, at index=2 and index=5. Return the minimum index, which is 2.

Example 2:

Input:

mountainArr = [0,1,2,4,2,1], target = 3

Output:

-1

Explanation:

3 does not exist in

the array,

so we return -1.

Constraints:

3 <= mountainArr.length() <= 10

4

0 <= target <= 10

9

0 <= mountainArr.get(index) <= 10

9

Code Snippets

C++:

```
/**
* // This is the MountainArray's API interface.
```

```
* // You should not implement it, or speculate about its implementation
* class MountainArray {
* public:
* int get(int index);
* int length();
* };
*/

class Solution {
public:
int findInMountainArray(int target, MountainArray &mountainArr) {

}
};
```

Java:

```
/**
* // This is MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* interface MountainArray {
* public int get(int index) {}
* public int length() {}
* }
*/

class Solution {
public int findInMountainArray(int target, MountainArray mountainArr) {

}
}
```

Python3:

```
# """
# This is MountainArray's API interface.
# You should not implement it, or speculate about its implementation
# """
#class MountainArray:
# def get(self, index: int) -> int:
# def length(self) -> int:
```

```
class Solution:
def findInMountainArray(self, target: int, mountainArr: 'MountainArray') ->
int:
```

Python:

```
# """
# This is MountainArray's API interface.
# You should not implement it, or speculate about its implementation
# """
#class MountainArray(object):
# def get(self, index):
# """
# :type index: int
# :rtype int
# """
#
# def length(self):
# """
# :rtype int
# """

class Solution(object):
def findInMountainArray(self, target, mountainArr):
"""
:type target: integer
:type mountain_arr: MountainArray
:rtype: integer
"""
```

JavaScript:

```
/**
* // This is the MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* function MountainArray() {
* @param {number} index
* @return {number}
* this.get = function(index) {
* ...
* };
*
```

```
* @return {number}
* this.length = function() {
* ...
* };
* };
*/

/**
* @param {number} target
* @param {MountainArray} mountainArr
* @return {number}
*/
var findInMountainArray = function(target, mountainArr) {

};
```

TypeScript:

```
/**
* // This is the MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* class MountainArray {
* get(index: number): number {}
*
* length(): number {}
* }
*/

function findInMountainArray(target: number, mountainArr: MountainArray) {

};
```

C#:

```
/**
* // This is MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* class MountainArray {
* public int Get(int index) {}
* public int Length() {}
* }
*/
```

```
class Solution {
public int FindInMountainArray(int target, MountainArray mountainArr) {

}
}
```

C:

```
/**
* ********************************************************************
* // This is the MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* ********************************************************************
*
* int get(MountainArray *, int index);
* int length(MountainArray *);
*/

int findInMountainArray(int target, MountainArray* mountainArr) {

}
```

Go:

```
/**
* // This is the MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* type MountainArray struct {
* }
*
* func (this *MountainArray) get(index int) int {}
* func (this *MountainArray) length() int {}
*/

func findInMountainArray(target int, mountainArr *MountainArray) int {

}
```

Kotlin:

```
/**
* // This is MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* class MountainArray {
* fun get(index: Int): Int {}
* fun length(): Int {}
* }
*/

class Solution {
fun findInMountainArray(target: Int, mountainArr: MountainArray): Int {

}
}
```

Swift:

```
/**
* // This is MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* interface MountainArray {
* public func get(_ index: Int) -> Int {}
* public func length() -> Int {}
* }
*/

class Solution {
func findInMountainArray(_ target: Int, _ mountainArr: MountainArray) -> Int
{

}
}
```

Rust:

```
/**
* // This is the MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* struct MountainArray;
* impl MountainArray {
* fn get(index:i32)->i32;
* fn length()->i32;
* };
```

```
*/

impl Solution {
pub fn find_in_mountain_array(target: i32, mountainArr: &MountainArray) ->
i32 {

}
}
```

Ruby:

```
# This is MountainArray's API interface.
# You should not implement it, or speculate about its implementation
# class MountainArray
# def get(index):
#
# end
#
# def length()
#
# end
# end

# @param {int} int
# @param {MountainArray} mountain_arr
# @return {int}
def findInMountainArray(target, mountainArr)

end
```

PHP:

```
/**
* // This is MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* class MountainArray {
* function get($index) {}
* function length() {}
* }
*/

class Solution {
```

```
/**
* @param Integer $target
* @param MountainArray $mountainArr
* @return Integer
*/
function findInMountainArray($target, $mountainArr) {

}
}
```

Scala:

```
/**
* // This is MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* class MountainArray {
* def get(index: Int): Int = {}
* def length(): Int = {}
* }
*/

object Solution {
def findInMountainArray(value: Int, mountainArr: MountainArray): Int = {

}
}
```

Solutions

C++ Solution:

```
/*
* Problem: Find in Mountain Array
* Difficulty: Hard
* Tags: array, 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
*/
```

```
/**
* // This is the MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* class MountainArray {
* public:
* int get(int index);
* int length();
* };
*/

class Solution {
public:
int findInMountainArray(int target, MountainArray &mountainArr) {

}
};
```

Java Solution:

```
/**
* Problem: Find in Mountain Array
* Difficulty: Hard
* Tags: array, 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
*/

/**
* // This is MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* interface MountainArray {
* public int get(int index) {
// TODO: Implement optimized solution
return 0;
}
* public int length() {
// TODO: Implement optimized solution
return 0;
```

```
}
* }
*/

class Solution {
public int findInMountainArray(int target, MountainArray mountainArr) {

}
}
```

Python3 Solution:

```
"""
Problem: Find in Mountain Array
Difficulty: Hard
Tags: array, 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
"""

# """
# This is MountainArray's API interface.
# You should not implement it, or speculate about its implementation
# """
#class MountainArray:
# def get(self, index: int) -> int:
# def length(self) -> int:

class Solution:
def findInMountainArray(self, target: int, mountainArr: 'MountainArray') ->
int:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
# """
# This is MountainArray's API interface.
# You should not implement it, or speculate about its implementation
```

```
# """
#class MountainArray(object):
# def get(self, index):
# """
# :type index: int
# :rtype int
# """
#
# def length(self):
# """
# :rtype int
# """

class Solution(object):
def findInMountainArray(self, target, mountainArr):
"""
:type target: integer
:type mountain_arr: MountainArray
:rtype: integer
"""
```

JavaScript Solution:

```
/**
* Problem: Find in Mountain Array
* Difficulty: Hard
* Tags: array, 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
*/

/**
* // This is the MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* function MountainArray() {
* @param {number} index
* @return {number}
* this.get = function(index) {
* ...
```

```
* };
*
* @return {number}
* this.length = function() {
* ...
* };
* };
*/

/**
* @param {number} target
* @param {MountainArray} mountainArr
* @return {number}
*/
var findInMountainArray = function(target, mountainArr) {

};
```

TypeScript Solution:

```
/**
* Problem: Find in Mountain Array
* Difficulty: Hard
* Tags: array, 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
*/

/**
* // This is the MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* class MountainArray {
* get(index: number): number {}
*
* length(): number {}
* }
*/

function findInMountainArray(target: number, mountainArr: MountainArray) {
```

```
};
```

C# Solution:

```
/*
* Problem: Find in Mountain Array
* Difficulty: Hard
* Tags: array, 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
*/

/**
* // This is MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* class MountainArray {
* public int Get(int index) {}
* public int Length() {}
* }
*/

class Solution {
public int FindInMountainArray(int target, MountainArray mountainArr) {

}
}
```

C Solution:

```
/*
* Problem: Find in Mountain Array
* Difficulty: Hard
* Tags: array, 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
*/
```

```
/**
* ************************************************************************
* // This is the MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* ************************************************************************
*
* int get(MountainArray *, int index);
* int length(MountainArray *);
*/

int findInMountainArray(int target, MountainArray* mountainArr) {

}
```

Go Solution:

```
// Problem: Find in Mountain Array
// Difficulty: Hard
// Tags: array, 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

/**
* // This is the MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* type MountainArray struct {
* }
*
* func (this *MountainArray) get(index int) int {}
* func (this *MountainArray) length() int {}
*/

func findInMountainArray(target int, mountainArr *MountainArray) int {

}
```

Kotlin Solution:

```
/**
* // This is MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* class MountainArray {
* fun get(index: Int): Int {}
* fun length(): Int {}
* }
*/

class Solution {
fun findInMountainArray(target: Int, mountainArr: MountainArray): Int {

}
}
```

Swift Solution:

```
/**
* // This is MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* interface MountainArray {
* public func get(_ index: Int) -> Int {}
* public func length() -> Int {}
* }
*/

class Solution {
func findInMountainArray(_ target: Int, _ mountainArr: MountainArray) -> Int
{

}
}
```

Rust Solution:

```
// Problem: Find in Mountain Array
// Difficulty: Hard
// Tags: array, 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
```

```
/**
* // This is the MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* struct MountainArray;
* impl MountainArray {
* fn get(index:i32)->i32;
* fn length()->i32;
* };
*/

impl Solution {
pub fn find_in_mountain_array(target: i32, mountainArr: &MountainArray) ->
i32 {

}
}
```

Ruby Solution:

```
# This is MountainArray's API interface.
# You should not implement it, or speculate about its implementation
# class MountainArray
# def get(index):
#
# end
#
# def length()
#
# end
# end

# @param {int} int
# @param {MountainArray} mountain_arr
# @return {int}
def findInMountainArray(target, mountainArr)

end
```

PHP Solution:

```
/**
* // This is MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* class MountainArray {
* function get($index) {}
* function length() {}
* }
*/

class Solution {
/**
* @param Integer $target
* @param MountainArray $mountainArr
* @return Integer
*/
function findInMountainArray($target, $mountainArr) {

}
}
```

Scala Solution:

```
/**
* // This is MountainArray's API interface.
* // You should not implement it, or speculate about its implementation
* class MountainArray {
* def get(index: Int): Int = {}
* def length(): Int = {}
* }
*/

object Solution {
def findInMountainArray(value: Int, mountainArr: MountainArray): Int = {

}
}
```