

Problem 702: Search in a Sorted Array of Unknown Size

Problem Information

Difficulty: Medium

Acceptance Rate: 72.98%

Paid Only: Yes

Tags: Array, Binary Search, Interactive

Problem Description

This is an `**_interactive problem_**`.

You have a sorted array of `**unique**` elements and an `**unknown size**`. You do not have an access to the array but you can use the ``ArrayReader`` interface to access it. You can call ``ArrayReader.get(i)`` that:

`*` returns the value at the `ith` index (**0-indexed**) of the secret array (i.e., `secret[i]`), or *` returns `231 - 1` if the `i` is out of the boundary of the array.`

You are also given an integer ``target``.

Return the index ``k`` of the hidden array where ``secret[k] == target`` or return ``-1`` otherwise.

You must write an algorithm with ``O(log n)`` runtime complexity.

`**Example 1:**`

`**Input:** secret = [-1,0,3,5,9,12], target = 9 **Output:** 4 **Explanation:** 9 exists in secret and its index is 4.`

`**Example 2:**`

`**Input:** secret = [-1,0,3,5,9,12], target = 2 **Output:** -1 **Explanation:** 2 does not exist in secret so return -1.`

****Constraints:****

*`1 <= secret.length <= 104` *`-104 <= secret[i], target <= 104` *`secret` is sorted in a strictly increasing order.

Code Snippets

C++:

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

class Solution {
public:
    int search(const ArrayReader& reader, int target) {

    }
};
```

Java:

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

class Solution {
    public int search(ArrayReader reader, int target) {

    }
}
```

Python3:

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

class Solution:
def search(self, reader: 'ArrayReader', target: int) -> int:
```