702. Search in a Sorted Array of Unknown Size

Medium 600 32 Add to List Share

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 i^{th} index (**0-indexed**) of the secret array (i.e., `secret[i]`), or
- returns $2^{31} - 1$ 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:

- $1 <= secret.length <= 10^4$
- $-10^4 <= secret[i], target <= 10^4$
- `secret` is sorted in a strictly increasing order.

Accepted 56,722 | Submissions 81,047

Seen this question in a real interview before? Yes No

Companies

0 ~ 6 months 6 months ~ 1 year 1 year ~ 2 years

Microsoft 2 Facebook 2

Related Topics

Array Binary Search Interactive

Similar Questions

Binary Search Easy

Find the Index of the Large Integer Medium

Java Autocomplete

```java
class Solution {
  public int search(ArrayReader reader, int target) {
    if (reader.get(0) == target) return 0;

    // search boundaries
    int left = 0, right = 1;
    while (reader.get(right) < target) {
      left = right;
      right <<= 1;
    }

    // binary search
    int pivot, num;
    while (left <= right) {
      pivot = left + ((right - left) >> 1);
      num = reader.get(pivot);

      if (num == target) return pivot;
      if (num > target) right = pivot - 1;
      else left = pivot + 1;
    }

    // there is no target element
    return -1;
  }
}
```