35. Search Insert Position

Easy 3980 307 Add to List Share

Given a sorted array of distinct integers and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.

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

Example 1:

```
Input: nums = [1,3,5,6], target = 5
Output: 2
```

Example 2:

```
Input: nums = [1,3,5,6], target = 2
Output: 1
```

Example 3:

```
Input: nums = [1,3,5,6], target = 7
Output: 4
```

Example 4:

```
Input: nums = [1,3,5,6], target = 0
Output: 0
```

Example 5:

```
```

```python
class Solution:
    def searchInsert(self, nums, target): # works even if there are duplicates.
        l , r = 0, len(nums)-1
        while l <= r:
            mid=(l+r) // 2
            if nums[mid] < target:
                l = mid+1
            else:
                if nums[mid]== target and nums[mid-1]!=target:
                    return mid
                else:
                    r = mid-1
        return l
```

Your previous code was restored from your local storage. Reset to default

Testcase | Run Code Result | Debugger

Accepted Runtime: 33 ms

Your input
[1,3,5,6]
5

Output 2 Diff

Expected 2