Search Insert Position

Easy

3778301Add to ListShare

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:

```
Input: nums = [1], target = 0

Output: 0
```

Solution

```
class Solution{

  public int searchInsert(int[] nums, int target) {

    if(target>nums[nums.length-1]){

        return nums.length;
```

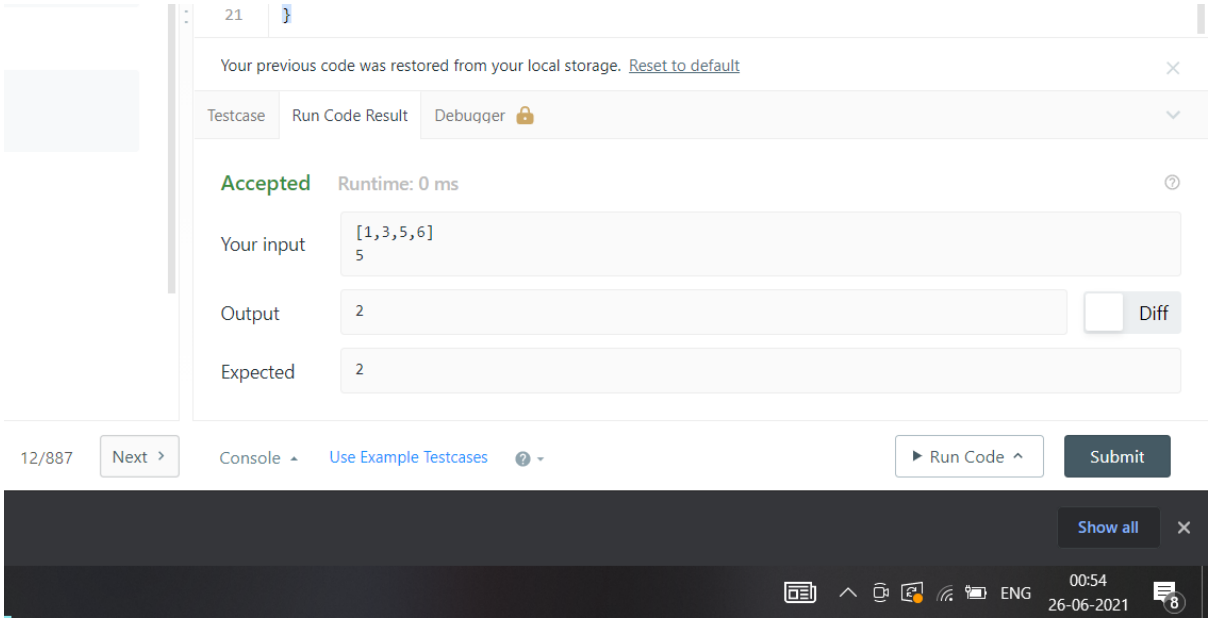
```
    }

    int l=0;
    int r=nums.length-1;

    while(l<r){
        int m = l+(r-l)/2;
        if(target>nums[m]){
            l=m+1;
        }else{
            r=m;
        }
    }

    return l;
}
}
```

Output


21 }
Your previous code was restored from your local storage. Reset to default
Testcase
Run Code Result
Debugger
Accepted
Runtime: 0 ms
Your input
[1,3,5,6]
5
Output
2
Diff
Expected
2

Show all
ENG
00:54
26-06-2021
8