LEETCODE PROBLEM – 35

35. Search Insert Position

Easy Topics Companies

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
```

Constraints:

- `1 <= nums.length <= 10^4`
- `-10^4 <= nums[i] <= 10^4`
- `nums` contains **distinct** values sorted in **ascending** order.
- `-10^4 <= target <= 10^4`

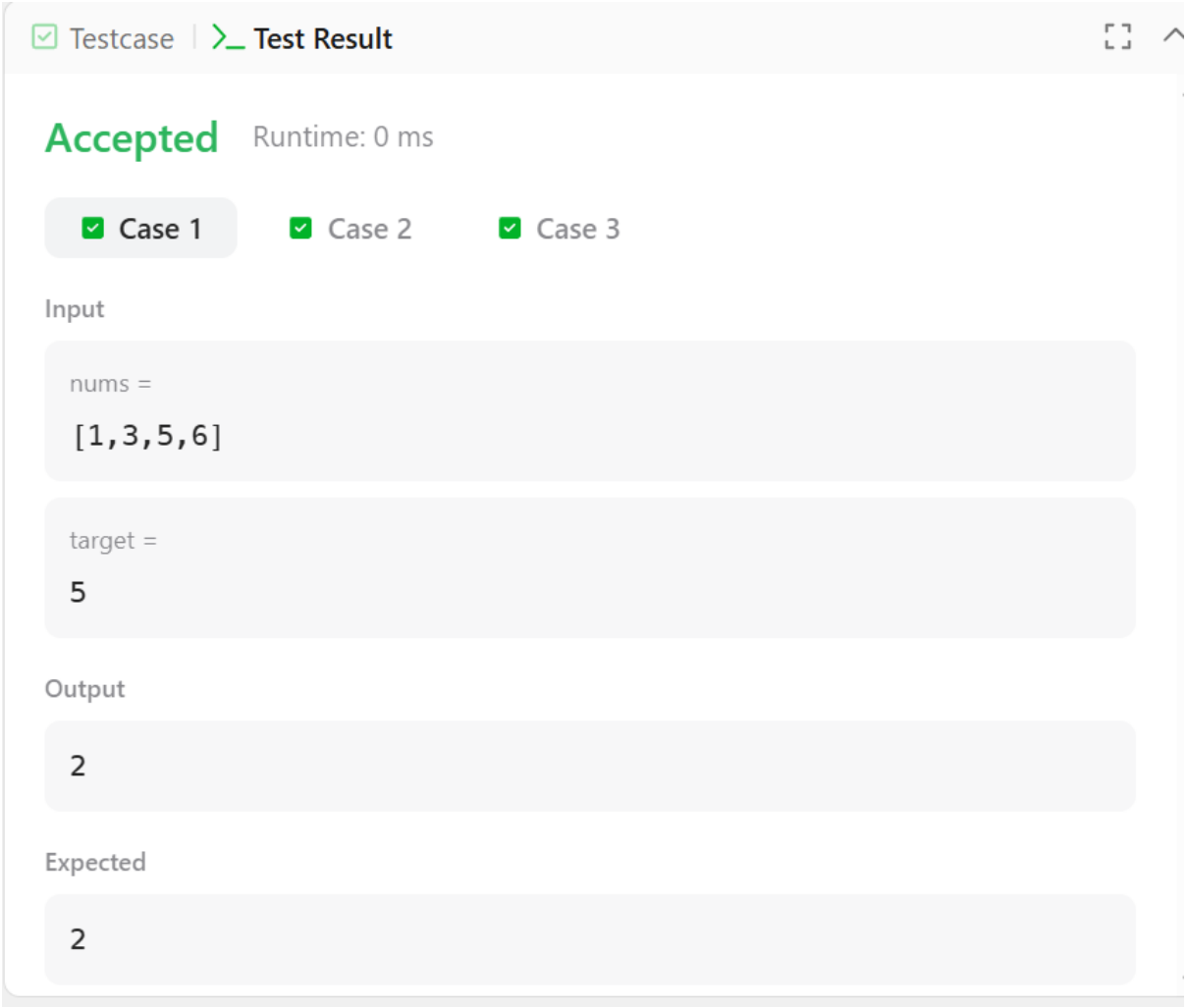
</> Code

C Auto

```c
int searchInsert(int* nums, int numsSize, int target) {
    int low = 0, high = numsSize - 1, mid;

    while (low <= high) {
        mid = (low + high) / 2;

        if (nums[mid] == target)
            return mid;
        else if (nums[mid] < target)
            low = mid + 1;
        else
            high = mid - 1;
    }
    return low;
}
```


Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
nums =
[1,3,5,6]
target =
5
Output
2
Expected
2