

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`

</> Code

C   Auto

    

```
1 int searchInsert(int* nums, int numsSize, int target) {
2     int left = 0;
3     int right = numsSize - 1;
4
5     while (left <= right) {
6         int mid = left + (right - left) / 2;
7
8         if (nums[mid] == target) {
9             return mid;          // target found
10        } else if (nums[mid] < target) {
11            left = mid + 1;        // search right half
12        } else {
13            right = mid - 1;       // search left half
14        }
15    }
16
17    // left is the correct insert position
18    return left;
19 }
20
```

☒ 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

 [Contribute a testcase](#)