

LeetCode 第 35 号问题：搜索插入位置

本文首发于公众号「图解面试算法」，是 [图解 LeetCode](#) 系列文章之一。

同步博客：<https://www.algomooc.com>

题目来源于 LeetCode 第 35 号问题：搜索插入位置。

题目

给定一个排序数组和一个目标值，在数组中找到目标值，并返回其索引。如果目标值不存在于数组中，返回它将会被按顺序插入的位置。 你可以假设数组中无重复元素。

示例 1:

```
输入: [1,3,5,6], 5
输出: 2
```

示例 2:

```
输入: [1,3,5,6], 2
输出: 1
```

示例 3:

```
输入: [1,3,5,6], 7
输出: 4
```

示例 4:

```
输入: [1,3,5,6], 0
输出: 0
```

思路解析

暴力循环法

这个题看起来就是很简单的，就是一道考验查找算法的题目。最简单的就是暴力查找了。

思路

遍历这个数组，然后如果当前值和目标值target一致或小于目标值target，那么就return 当前下标。这种解法的时间复杂度是O(N)

动画理解

代码实现

```
//时间复杂度: O(n)
//空间复杂度: O(1)
```

```
class Solution {
    public int searchInsert(int[] nums, int target) {
        int i=0;
        for(;i<nums.length;i++){
            if (nums[i]>=target){
                break;
            }
        }
        return i;
    }
}
```

二分法

思路

除了暴力法，我们在排序数组中查找值还可以用的一种方法是二分法，思路还是和改良的暴力循环法一样，先找到左右边界，然后计算，每次可以省出一半的时间。时间复杂度为 $O(\log n)$

代码实现

```
//时间复杂度:  $O(\log(n))$ 
//空间复杂度:  $O(1)$ 
class Solution {
    public int searchInsert(int[] nums, int target) {
        if (target>nums[nums.length-1]) {
            return nums.length;
        }
        int left=0;
        int right=nums.length-1;
        while (left < right) {
            int mid = (left + right) / 2;
            if (nums[mid] < target) {
                left = mid + 1;
            } else {
                right = mid;
            }
        }
        return left;
    }
}
```