

704. 二分查找.

给定一个 n 个元素有序的（升序）整型数组 `nums` 和一个目标值 `target`，写一个函数搜索 `nums` 中的 `target`，如果目标值存在返回下标，否则返回 `-1`。

示例 1:

```
1 输入: nums = [-1,0,3,5,9,12], target = 9
2 输出: 4
3 解释: 9 出现在 nums 中并且下标为 4
4
```

示例 2:

```
1 输入: nums = [-1,0,3,5,9,12], target = 2
2 输出: -1
3 解释: 2 不存在 nums 中因此返回 -1
4
```

提示:

1. 你可以假设 你可以假设 `nums` 中的所有元素是不重复的。
2. n 将在 $[1, 10000]$ 之间。
3. `nums` 的每个元素都将在 $[-9999, 9999]$ 之间。

分析：（这里参考代码随想录的）这道题目的前提是数组为有序数组，同时题目还强调数组中无重复元素，因为一旦有重复元素，使用二分查找法返

回的元素下标可能不是唯一的，这些都是使用二分法的前提条件，当大家看到题目描述满足如上条件的时候，可要想一想是不是可以用二分法了。

二分查找涉及的很多的边界条件，逻辑比较简单，但就是写不好。例如到底是 `while(left < right)` 还是 `while(left <= right)`，到底是 `right = middle` 呢，还是要 `right = middle - 1` 呢？

大家写二分法经常写乱，主要是因为对区间的定义没有想清楚，区间的定义就是不变量。要在二分查找的过程中，保持不变量，就是在 `while` 寻找中每一次边界的处理都要坚持根据区间的定义来操作，这就是循环不变量规则。写二分法，区间的定义一般为两种，左闭右闭即 `[left, right]`，或者左闭右开即 `[left, right)`。

下标:

0

1

2

3

4

5

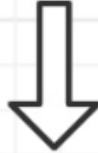
6

1	2	3	4	7	9	10
---	---	---	---	---	---	----

L = 0

M = 3

R = 6



1	2	3
---	---	---

L = 0

M = 1

R = 2

代码如下:

```
1 class Solution {
2 public:
3     int search(vector<int>& nums, int target) {
4         int left=0;//定义左区间
5         int right=nums.size()-1;//左闭右闭，所以右区间为nums的size即地址最后的的下一位减1
6         while(left<=right)//遍历
7         {
8             int middle=left+(right-left)/2;//取中间
9             if(nums[middle]>target)//取中间值右边空间
10            {
11                right=middle-1;
12            }
13            else if(nums[middle]<target)
14            {
15                left=middle+1;
16            }
17            else
18            {
19                return middle;
20            }
21        }
22        return -1;
23    }
```

```
23     }  
24 };
```

69. x 的平方根

简单

相关标签

相关企业

提示

给你一个非负整数 x ，计算并返回 x 的 **算术平方根**。

由于返回类型是整数，结果只保留 **整数部分**，小数部分将被 **舍去**。

注意：不允许使用任何内置指数函数和算符，例如 `pow(x, 0.5)` 或者 `x ** 0.5`。

示例 1：

输入： $x = 4$ **输出：**2

示例 2：

输入： $x = 8$ **输出：**2 **解释：**8 的算术平方根是 2.82842...，由于返回类型是整数，小数部分将被舍去。

分析：二分法；