

[LintCode领扣题解 \(/problem\)](#) / 二分查找 · First Position of Target

## 二分查找 · First Position of Target

中文

[LintCode 版权所有 \(/problem/?tags=lintcode-copyright\)](#)[数组 \(/problem/?tags=array\)](#)[二分法 \(/problem/?tags=binary-search\)](#)

### 描述

给定一个排序的整数数组（升序）和一个要查找的整数 `target`，用  $O(\log n)$  的时间查找到 `target` 第一次出现的下标（从0开始），如果 `target` 不存在于数组中，返回 `-1`。

### 样例

样例 1:

输入: `[1,4,4,5,7,7,8,9,9,10]`, 1  
输出: 0

样例解释:  
第一次出现在第0个位置。

样例 2:

输入: `[1, 2, 3, 3, 4, 5, 10]`, 3  
输出: 2

样例解释:  
第一次出现在第2个位置

样例 3:

输入: `[1, 2, 3, 3, 4, 5, 10]`, 6  
输出: -1

样例解释:  
没有出现6, 返回-1

### 挑战

如果数组中的整数个数超过了  $2^{32}$ ，你的算法是否会出错？

在线评测地址: <https://www.lintcode.com/problem/first-position-of-target/> (<https://www.lintcode.com/problem/first-position-of-target/>)

[收起题目描述 ^](#)

语言类型

[ALL \(16\)](#)[python \(8\)](#)[java \(4\)](#)[cpp \(3\)](#)[javascript \(1\)](#)[上传题解](#)

令狐冲

更新于 10/13/2020, 8:51:58 PM

二分查找的模版，找第一次出现的位置。注意可能出现多次，所以当找到这个数字的时候不能直接结束，而要将 `end` 移动到 `mid` 处。直到最后缩小到只有一个或两个数字时，优先判断 `start`，再判断 `end`。

时间复杂度  $O(\log_2 n)$

```
/**
 * 本参考程序由九章算法用户提供。版权所有，转发请注明出处。
```

\* - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。  
\* - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版  
\* - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD  
\* - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课  
\* - 更多详情请见官方网站: [http://www.jiuzhang.com/?utm\\_source=code](http://www.jiuzhang.com/?utm_source=code)  
\*/

// version 1: with jiuzhang template

```
class Solution {
    /**
     * @param nums: The integer array.
     * @param target: Target to find.
     * @return: The first position of target. Position starts from 0.
     */
    public int binarySearch(int[] nums, int target) {
        if (nums == null || nums.length == 0) {
            return -1;
        }

        int start = 0, end = nums.length - 1;
        while (start + 1 < end) {
            int mid = start + (end - start) / 2;
            if (nums[mid] == target) {
                end = mid;
            } else if (nums[mid] < target) {
                start = mid;
                // or start = mid + 1
            } else {
                end = mid;
                // or end = mid - 1
            }
        }

        if (nums[start] == target) {
            return start;
        }
        if (nums[end] == target) {
            return end;
        }
        return -1;
    }
}
```

// version 2: without jiuzhang template

```
class Solution {
    /**
     * @param nums: The integer array.
     * @param target: Target to find.
     * @return: The first position of target. Position starts from 0.
     */
    public int binarySearch(int[] nums, int target) {
        if (nums == null || nums.length == 0) {
            return -1;
        }

        int start = 0, end = nums.length - 1;
        while (start < end) {
            int mid = start + (end - start) / 2;
            if (nums[mid] == target) {
                end = mid;
            } else if (nums[mid] < target) {
                start = mid + 1;
            } else {
                end = mid - 1;
            }
        }
    }
}
```

```
    if (nums[start] == target) {
        return start;
    }

    return -1;
}
```

👍 获赞 3

💬 2 条评论

**你的口袋题库**  
2000+ 算法真题、国内外名企题库免费开放



九章算法APP

令狐冲

更新于 6/9/2020, 7:03:50 AM

二分查找的模版, 找第一次出现的位置。注意可能出现多次, 所以当找到这个数字的时候不能直接结束, 而要将end移动到mid处。直到最后缩小到只有一个或两个数字时, 优先判断start, 再判断end。

时间复杂度  $O(\log_2 n)$

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
# python 2
class Solution:
    # @param nums: The integer array
    # @param target: Target number to find
    # @return the first position of target in nums, position start from 0
    def binarySearch(self, nums, target):
        # write your code here
        left, right = 0, len(nums)-1
        while left + 1 < right:
            mid = (left + right) / 2
            if nums[mid] < target:
                left = mid
            else:
                right = mid
        if nums[left] == target:
            return left
        elif nums[right] == target:
            return right
        return -1;
```

👍 获赞 2

💬 5 条评论

令狐冲

更新于 6/9/2020, 7:04:28 AM

二分查找的模版, 找第一次出现的位置。注意可能出现多次, 所以当找到这个数字的时候不能直接结束, 而要将end移动到mid处。直到最后缩小到只有一个或两个数字时, 优先判断start, 再判断end。

时间复杂度 $O(\log_2 n)$

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution {
public:
    int binarySearch(vector<int> &A, int target) {
        if (A.size() == 0) {
            return -1;

            int start = 0;
            int end = A.size() - 1;
            int mid;

            while (start + 1 < end) {
                mid = start + (end - start) / 2;
                if (A[mid] == target) {
                    end = mid;
                } else if (A[mid] < target) {
                    start = mid;
                } else if (A[mid] > target) {
                    end = mid;
                }
            }

            if (A[start] == target) {
                return start;
            }
            if (A[end] == target) {
                return end;
            }

            return -1;
        }
    }
};
```

👍 获赞 0

💬 2 条评论



华助教

更新于 6/9/2020, 7:04:26 AM

二分查找的模版, 找第一次出现的位置。注意可能出现多次, 所以当找到这个数字的时候不能直接结束, 而要将end移动到mid处。直到最后缩小到只有一个或两个数字时, 优先判断start, 再判断end。

时间复杂度 $O(\log_2 n)$



invitation/shi



```
/**
 * 本参考程序由九章算法用户提供。版权所有，转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作，授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括：九章算法班 2020升级版，算法强化班，算法基础班，北美算法面试高频题班，Java 高级工程师 P6+ 小班课，面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括：系统设计 System Design，面向对象设计 OOD
 * - 专题及项目类课程包括：动态规划专题班，Big Data - Spark 项目实战，Django 开发项目课
 * - 更多详情请见官方网站：http://www.jiuzhang.com/?utm_source=code
 */
/**
 * @param nums: The integer array.
 * @param target: Target to find.
 * @return: The first position of target. Position starts from 0.
 */
const binarySearch = function (nums, target) {
    var left, right, mid, res;
    res = 0;
    left = 0;
    right = nums.length - 1;
    while (left <= right) {
        mid = Math.floor((left + right) / 2);
        if (nums[mid] === target) {
            res = mid;
        }
        if (nums[mid] >= target) {
            right = mid - 1;
        } else {
            left = mid + 1;
        }
    }
    if (nums[res] !== target) {
        return -1;
    }
    return res;
}
```

👍 获赞 0    💬 添加评论



九章-小原

更新于 6/9/2020, 7:04:00 AM

## 解题思路：

题目提到，给定的数组已经排序，若从小到大遍历数组查找target，则时间复杂度为 $O(n)$ ，n为数组长度。需要用一个 $O(\log n)$ 的时间复杂度去完成本题，那么需要用到二分查找。

二分查找常用于查找有序数组中目标数target的位置，用 left 和 right 记录target所在的区间端点，每次将区间的中间位置值和target作比较，然后移动区间端点。

## 算法流程：

- 将区间赋值为整个数组区间（left = 0, right = n - 1），取中间位置mid
- 若  $a[mid] < target$ ，则将区间缩小到原区间的右区间(left = mid + 1)
- 若  $a[mid] \geq target$ ，则将区间缩小至原区间的左区间(right = mid)
- 若left >= right 时，若  $a[right] = target$ 则返回right, 否则返回-1

## 复杂度分析：

- 时间复杂度： $O(\log n)$

每次查找都将区间缩小至原来长度的一半，可见查找的最多次数为logn

- 空间复杂度:  $O(1)$

查找不需要开辟新的非常数级空间, 只需在原数组基础上进行查找即可

## 代码:

c++ java python

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution {
public:
    int binarySearch(vector<int> &nums, int target) {
        int left = 0;
        int right = nums.size() - 1;
        int mid;
        while (left < right) {
            //得到中间位置
            mid = (right + left) / 2;
            if (nums[mid] < target) {
                left = mid + 1;
            } else {
                right = mid;
            }
        }
        if (nums[right] == target) {
            return right;
        }
        return -1;
    }
};
```

👍 获赞 0    💬 添加评论



九章用户2WYHZZK

更新于 6/9/2020, 7:03:56 AM

百花齐放, 分享我的二分法模版, 用它实现了近10道二分法的题, 还没有发现问题。

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
public class Solution {

    public int binarySearch(int[] nums, int target) {
        if (nums == null || nums.length == 0)
            return -1;

        int left = 0, right = nums.length - 1;
        while (left <= right) {
            int mid = left + (right - left) / 2;
            if (target == nums[mid])
                right = mid - 1;
            else if (target < nums[mid])
                right = mid - 1;
            else
                left = mid + 1;
        }

        if (left < nums.length && nums[left] == target) // left = first position
            return left;
        return -1;
    }
}

```

👍 获赞 1    💬 添加评论



九章用户98DV6M

更新于 6/9/2020, 7:03:52 AM

用内置的index函数可以快速获取出现的第一次位置 但是缺点是速度很慢

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution:
    """
    @param nums: The integer array.
    @param target: Target to find.
    @return: The first position of target. Position starts from 0.
    """
    def binarySearch(self, nums, target):
        return nums.index(target) if target in nums else -1

```

👍 获赞 1    💬 添加评论

**九章用户CZ64MV**

更新于 6/9/2020, 7:03:52 AM

如果numsmid () >= target,则题解一定在左边, 否则题解在右边

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution:
    """
    @param nums: The integer array.
    @param target: Target to find.
    @return: The first position of target. Position starts from 0.
    """
    def binarySearch(self, nums, target):
        # write your code here
        if not nums or not target:
            return -1

        start, end = 0, len(nums) - 1
        while start + 1 < end:
            mid = (start + end) // 2
            if target <= nums[mid]:
                end = mid
            else:
                start = mid

        if nums[start] == target:
            return start
        if nums[end] == target:
            return end

        return -1
```

👍 获赞 1

💬 添加评论

**xqfiswhy**

更新于 6/9/2020, 7:04:26 AM

二分思想嘛,很简单就是中间往两头哈哈.然后要注意的是题目中没有提到的有重复数字的情况. 然后,,,对于代码中的问题就是我把py和scala给搞混淆了,以为py中没有break,哈哈,学到了



```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution:
    """
    @param nums: The integer array.
    @param target: Target to find.
    @return: The first position of target. Position starts from 0.
    """
    def binarySearch(self, nums, target):
        # write your code here
        if nums==None:
            return -1
        length=len(nums)
        if target<nums[0] or target>nums[length-1]:
            return -1

        left=0
        right=length-1
        find=False
        result=0
        while find!=True and left<=length-1 and right>=0 and left<=right:
            mid=(left+right)//2
            if target>nums[mid]:
                left=mid+1
            elif target<nums[mid]:
                right=mid-1
            else :
                result=mid
                find=True

        if find:
            flag=False
            while result>0 and flag==False:
                if nums[result-1]==target:
                    result-=1
                else :
                    flag=True
            return result
        else :
            return -1
```

👍 获赞 0

💬 添加评论

**九章用户16GH70**

更新于 6/9/2020, 7:04:24 AM

使用 recursion 解决此问题。recursion 终结情况1: start > end, 此时没有找到匹配数, return -1 recursion 终结情况2: start == end 且 nums[start] == target, 此条件必要, 不然会跳不出 nummid() == target 的循环。

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
public class Solution {
    /**
     * @param nums: The integer array.
     * @param target: Target to find.
     * @return: The first position of target. Position starts from 0.
     */
    public int binarySearch(int[] nums, int target) {
        // write your code here
        if (nums == null || nums.length == 0) {
            return -1;
        }

        return searchHelper(nums, target, 0, nums.length - 1);
    }

    private int searchHelper(int[] nums, int target, int start, int end) {
        // ending situation 1
        if (start > end) {
            return -1;
        }
        // ending situation 2
        if (start == end && nums[start] == target) {
            return start;
        }

        int mid = start + (end - start) / 2;
        if (nums[mid] == target) {
            // continue doing recursion to find if there any same number before current mid
            return searchHelper(nums, target, start, mid);
        }
        if (nums[mid] < target) {
            return searchHelper(nums, target, mid + 1, end);
        }
        return searchHelper(nums, target, start, mid - 1);
    }
}
```

👍 获赞 0    💬 添加评论



cc189

更新于 6/9/2020, 7:04:21 AM

This is a test. This is a test. This is a test. This is a test. This is a test. This is a test.

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ /
Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution:
    """
    @param nums: The integer array.
    @param target: Target to find.
    @return: The first position of target. Position starts from 0.
    """
    def binarySearch(self, nums, target):
        l, r = 0, len(nums) - 1

        while r - 1 > l:
            mid = (r - l) // 2 + l
            if nums[mid] >= target: r = mid
            else: l = mid

        if nums[l] == target: return l
        if nums[r] == target: return r
        return -1;
```

👍 获赞 0

💬 添加评论



**MORVEL**

更新于 6/9/2020, 7:04:20 AM

使用递归解法进行二分搜索。Python语言。简介实在用不了30个字。

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
class Solution:
    """
    @param nums: The integer array.
    @param target: Target to find.
    @return: The first position of target. Position starts from 0.
    """
    def binarySearch(self, nums, target):
        # write your code here
        if nums is None:
            return -1
        else:
            start, end = self.searchHelper(nums, target, 0, len(nums)-1 )
            if nums[start] == target:
                return start
            elif nums[end] == target:
                return end
            else:
                return -1

    def searchHelper(self, nums, target, start, end):
        # Exit
        if start + 1 >= end:
            return start, end
        # Binary search: adjust start & end
        mid = start + (end - start) // 2
        if nums[mid] == target:
            end = mid
        elif nums[mid] > target:
            end = mid
        else:
            start = mid
        # Recurse
        return self.searchHelper(nums, target, start, end)
```

👍 获赞 0

💬 添加评论



jb

更新于 6/9/2020, 7:04:19 AM

```
class Solution: """ @param nums: The integer array. @param target: Target to find. @return: The first position of target. Position starts from 0. """ def
binarySearch(self, nums, target):
```

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
# write your code here
left, right = 0, len(nums)
while left + 1 < right:
    mid = (left + right) // 2
    if nums[mid] < target:
        left = mid
    else:
        right = mid
if nums[left] == target:
    return left
elif nums[right] == target:
    return right
return -1;

```

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution:
    """
    @param nums: The integer array.
    @param target: Target to find.
    @return: The first position of target. Position starts from 0.
    """
    def binarySearch(self, nums, target):
        # write your code here
        left, right = 0, len(nums)
        while left + 1 < right:
            mid = (left + right) // 2
            if nums[mid] < target:
                left = mid
            else:
                right = mid
        if nums[left] == target:
            return left
        elif nums[right] == target:
            return right
        return -1;

```

👍 获赞 0

💬 添加评论

**JianxiaGao**

更新于 6/9/2020, 7:04:18 AM

This solution is exactly based on Programming Pearl Column9: where there are only one comparison in loop and one if after loop.

```

/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution {
public:
    /**
     * @param nums: The integer array.
     * @param target: Target to find.
     * @return: The first position of target. Position starts from 0.
     */
    int binarySearch(vector<int> &nums, int target){
        int start = -1;
        int end = nums.size();
        if (end == 0) return -1;
        if (end == 1 && nums[0] != target) return -1;
        while(start+1 != end){
            auto mid = (start+end)/2;
            if(nums[mid] < target)
                start = mid;
            else
                end = mid;
        }
        if(end >= nums.size() || nums[end] != target) return -1;
        return end;
    }
};

```

👍 获赞 0

💬 添加评论

## 进阶课程

视频+互动

直播+互动

直播+互动

互动课

### 九章算法班 2021 版

8周时间精通 57 个核心高频考点, 9 招击破 FLAG、BATJ 算法面试。22....

### 系统架构设计 System Design

#### 2021 版

成为百万架构师必上。30 课时带你快速掌握18大系统架构设计知识点与面...

### 九章算法面试高频题冲刺班

每期更新 15% 题目, 考前押题, 一举拿下FLAG & BATJ Offer

### 面向对象设计 OOD

应届生及亚马逊面试必考, IT求职必备基础

