

LintCode领扣题解 (/problem) / 山脉序列中的最大值 · Maximum Number in Mountain Sequence

山脉序列中的最大值 · Maximum Number in Mountain Sequence

中文

LintCode 版权所有 (/problem/?tags=lintcode-copyright) 二分法 (/problem/?tags=binary-search)

描述

给 n 个整数的山脉数组，即先增后减的序列，找到山顶（最大值）

① 数组严格递增，严格递减

样例

例1:

输入: nums = [1, 2, 4, 8, 6, 3]  
输出: 8

例2:

输入: nums = [10, 9, 8, 7],  
输出: 10

在线评测地址: <https://www.lintcode.com/problem/maximum-number-in-mountain-sequence/> (<https://www.lintcode.com/problem/maximum-number-in-mountain-sequence/>)

收起题目描述 ^

语言类型 ALL (17) java (9) python (4) cpp (4) 上传题解

令狐冲  
更新于 11/7/2020, 8:40:09 PM

二分法，判断山脉趋势，按照取数递归左边或者右边即可。山顶的条件是第一个使得 `nums[i] > nums[i + 1]` 的 `i`。当然也可以反过来，最后一个使得 `nums[i] > nums[i - 1]` 的 `i`

python java



invitation/show-solutions



```
/**
 * 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: a mountain sequence which increase firstly and then decrease
    @return: then mountain top
    """
    def mountainSequence(self, nums):
        if not nums:
            return -1

        # find first index i so that nums[i] > nums[i + 1]
        start, end = 0, len(nums) - 1
        while start + 1 < end:
            mid = (start + end) // 2
            # mid + 1 保证不会越界
            # 因为 start 和 end 是 start + 1 < end
            if nums[mid] > nums[mid + 1]:
                end = mid
            else:
                start = mid

        return max(nums[start], nums[end])
```

👍 获赞 7

💬 1 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, 7:03:48 AM

二分法, 判断山脉趋势, 按照取数递归左边或者右边即可。

```
/**
 * 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
 */
// version 1: 二分法
public class Solution {
    /**
     * @param nums a mountain sequence which increase firstly and then decrease
     * @return then mountain top
     */
    public int mountainSequence(int[] nums) {
        // Write your code here
        if (nums == null || nums.length == 0) {
            return 0;
        }
        int start = 0;
        int end = nums.length - 1;
        while (start + 1 < end) {
            int mid = start + (end - start) / 2;
            if (nums[mid] > nums[mid + 1]) {
                end = mid;
            } else {
                start = mid;
            }
        }
        return Math.max(nums[start], nums[end]);
    }
}

// version 2: 一个比较啰嗦的版本, 本质也是2分法, 每次取两个点
public class Solution {
    /**
     * @param nums a mountain sequence which increase firstly and then decrease
     * @return then mountain top
     */
    public int mountainSequence(int[] nums) {
        // Write your code here
        int left = 0, right = nums.length - 1;
        while (left + 1 < right) {
            int m1 = left + (right - left) / 2;
            int m2 = right - (right - m1) / 2;
            if (nums[m1] < nums[m2]) {
                left = m1 + 1;
            } else if (nums[m1] > nums[m2]) {
                right = m2 - 1;
            } else {
                left = m1;
                right = m2;
            }
        }
        return nums[left] > nums[right] ? nums[left] : nums[right];
    }
}
```

👍 获赞 3

💬 4 条评论

**DDBear**

更新于 8/20/2020, 2:08:10 AM

# 算法：二分

## 算法思路

- 由于本题数据是具有部分单调性，我们可以考虑用二分法来解决
- 并且本题保证数组严格递增或严格递减，所以相邻两个数必不相等
- 我们可以通过  $a[mid]$  和  $a[mid+1]$  的大小关系来判断  $mid$  是在左侧还是右侧

## 代码思路

1. 设置左边界等于0，右边界等于  $numsLen - 1$
2. 对于  $mid$  所指向的数，若  $nums[mid] > nums[mid + 1]$  则表示 $mid$ 指向的数在最大值右侧或最大值，那么  $right = mid$ ，否则  $left = mid$
3. 不断重复 2 直到  $left + 1 == right$  退出
4.  $left$  和  $right$  指向的数中较大的一个即最大值，将其返回

# 复杂度分析

N表示数组 `nums` 长度

- 空间复杂度： $O(N)$
- 时间复杂度： $O(\log N)$

[java](#)[c++](#)[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
 */

public class Solution {

    /**
     * @param nums: a mountain sequence which increase firstly and then decrease
     * @return: then mountain top
     */

    public int mountainSequence(int[] nums) {

        if (nums == null || nums.length == 0) {

            return 0;

        }

        int left = 0;

        int right = nums.length - 1;

        while (left + 1 < right) {

            int mid = left + (right - left) / 2;

            if (nums[mid] > nums[mid + 1]) {

                right = mid;

            } else {

                left = mid;

            }

        }

        return Math.max(nums[left], nums[right]);

    }

}
```

👍 获赞 1    💬 添加评论



令狐冲

更新于 6/9/2020, 7:04:28 AM

二分法，判断山脉趋势，按照取数递归左边或者右边即可。

```

/**
 * 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 a mountain sequence which increase firstly and then decrease
     * @return then mountain top
     */
    int mountainSequence(vector<int>& nums) {
        // Write your code here
        int left = 0, right = nums.size() - 1;
        while (left + 1 < right) {
            int m1 = left + (right - left) / 2;
            int m2 = right - (right - m1) / 2;
            if (nums[m1] < nums[m2]) {
                left = m1 + 1;
            } else if (nums[m1] > nums[m2]) {
                right = m2 - 1;
            } else {
                left = m1;
                right = m2;
            }
        }
        return nums[left] > nums[right] ? nums[left] : nums[right];
    }
};

```

👍 获赞 0

💬 添加评论



YU

更新于 6/9/2020, 7:03:46 AM

偷懒做法。。。。。。题目中要求找到山峰, 返回值是峰值, 等同于排序后找最后一个即可。

```

/**
 * 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: a mountain sequence which increase firstly and then decrease
     * @return: then mountain top
     */
    public int mountainSequence(int[] nums) {
        // write your code here
        Arrays.sort(nums);
        return nums[nums.length - 1];
    }
}

```

👍 获赞 5    💬 添加评论



Chichi

更新于 6/9/2020, 7:03:50 AM

课上说峰值的分解条件必须是下降的第一个数，否则会出现死循环。但是我采用分界条件为上升的最后一个数，submit之后仍然可以通过。用于是用了 `start + 1 < end`，所以永远不会出现死循环。

```
/**
 * 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: a mountain sequence which increase firstly and then decrease
     * @return: then mountain top
     */
    public int mountainSequence(int[] nums) {
        // write your code here
        if (nums == null || nums.length == 0) {
            return -1;
        }

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

        if (nums[end] > nums[start]) {
            return nums[end];
        }
        else {
            return nums[start];
        }
    }
}
```

👍 获赞 2    💬 添加评论



X同学

更新于 6/9/2020, 7:03:56 AM

山峰的满足条件是左右都比它小，所以选取mid，然后在while loop中选择它左右的作为start, end 最后在check一下array两端的值是不是更大

```
/**
 * 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: a mountain sequence which increase firstly and then decrease
     * @return: then mountain top
     */
    public int mountainSequence(int[] nums) {
        if(nums == null || nums.length == 0) {
            return -1;
        }
        int mid = (nums.length) / 2;
        while(mid > 0 && mid < nums.length - 1) {
            int left = mid - 1;
            int right = mid + 1;
            if(nums[mid] > nums[left] && nums[mid] > nums[right]) {
                return nums[mid];
            } else if (nums[mid] > nums[right]) {
                mid = left;
            } else {
                mid = right;
            }
        }
        int x;
        x = Math.max(nums[mid], nums[0]);
        x = Math.max(x, nums[nums.length - 1]);
        return x;
    }
}
```

👍 获赞 1

💬 添加评论



九章用户3DCQLO

更新于 6/9/2020, 7:03:54 AM

while(left&lt;=right) 模板 \_\_\_\_



```
/**
 * 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 int mountainSequence(int[] nums) {
    int left = 0;
    int right = nums.length-1;

    while(left<=right) {
        int mid = (left+right)/2;

        if(mid-1>=0 && nums[mid-1]>nums[mid]) {
            right = mid-1;
        }
        else if(mid+1<=nums.length-1 && nums[mid+1]>nums[mid]) {
            left = mid+1;
        }
        else {
            return nums[mid];
        }
    }

    return -1;
}
```

👍 获赞 1    💬 添加评论



**Jet**

更新于 6/9/2020, 7:03:52 AM

- 1.算法: 二分法 为什么用(条件): XXOO, 无target 找到分界点。注意, 因为int数取整特性, 总是取较小的值, 所以判断两数 时用numsi () 与 numsi + 1 ()
- 2.代码实现 注意优化: 3.时空复杂度分析 时间复杂度: O(logn) 空间复杂度:O(1) 4.相关的题做过有哪些:

```

/**
 * 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: a mountain sequence which increase firstly and then decrease
     * @return: then mountain top
     */
    int mountainSequence(vector<int> &nums) {
        // write your code here
        //don't have equal situation
        int n = nums.size();
        int start = 0, end = n - 1, mid;
        while (start + 1 < end) {
            mid = start + (end - start) / 2;
            if (nums[mid] < nums[mid + 1]) {
                start = mid;
                continue;
            }
            //nums[mid] > nums[mid + 1]
            end = mid;
        }
        if (nums[start] > nums[end]) {
            return nums[start];
        }
        return nums[end];
    }
};

```

👍 获赞 1    💬 添加评论



九章用户AI9PYF

更新于 6/13/2020, 8:17:10 AM

```

public class Solution { /** @param nums: a mountain sequence which increase firstly and then decrease @return: then mountain top */ public int
mountainSequence(int[] nums) { // write your code here if(nums == null || nums.length == 0){ return -1; } int start = 0; int end = nums.length - 1; while(start + 1 <
end){ int mid = start + (end - start)/2; if(nums[mid] < nums[mid + 1]){ start = mid; }else{ end = mid; } } return Math.max(nums[start], nums[end]); } }

```

👍 获赞 0    💬 添加评论



九章用户90XVBR

更新于 6/9/2020, 7:04:27 AM

参考java版答案写出来的

```

/**
 * 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: a mountain sequence which increase firstly and then decrease
    @return: then mountain top
    """
    def mountainSequence(self, nums):
        # write your code here
        if len(nums) == 0:
            return -1
        start, end = 0, len(nums) - 1
        target = nums[-1]
        while start + 1 < end:
            mid = start + (end - start) / 2
            if nums[mid] < nums[mid+1]:
                start = mid
            else:
                end = mid
        return max(nums[start], nums[end])

```

👍 获赞 0

💬 1 条评论



九章用户51HVT8

更新于 6/9/2020, 7:04:24 AM

比较简单的解法, 可以使用一个for loop, 一个max作为标签, 进入循环就好, 时间复杂度就是O(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:
    /**
     * @param nums: a mountain sequence which increase firstly and then decrease
     * @return: then mountain top
     */
    int mountainSequence(vector<int> &nums) {
        // write your code here
        int max = 0;
        for(int i = 0; i < nums.size(); i++)
        {
            if(max < nums[i]) max = nums[i];
        }
        return max;
    }
};

```

👍 获赞 0    💬 添加评论

加载更多题解

## 进阶课程

视频+互动	直播+互动	直播+互动	互动课
<div>九章算法班 2021 版</div> <div>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....</div>	<div>系统架构设计 System Design 2021 版</div> <div>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...</div>	<div>九章算法面试高频题冲刺班</div> <div>每期更新 15% 题目，考前押题，一举拿下FLAG &amp; BATJ Offer</div>	<div>面向对象设计 OOD</div> <div>应届生及亚马逊面试必考，IT求职必备基础</div>