

LintCode领扣题解 (/problem) / 两数之和 · Two Sum

## 两数之和 · Two Sum

中文

Adobe (/problem/?tags=adobe) Yelp (/problem/?tags=yelp) Bloomberg (/problem/?tags=bloomberg) 微软 (/problem/?tags=microsoft)  
掉盒子 (/problem/?tags=dropbox) 脸书 (/problem/?tags=facebook) 亚马逊 (/problem/?tags=amazon) 爱彼迎 (/problem/?tags=airbnb)  
领英 (/problem/?tags=linkedin) 雅虎 (/problem/?tags=yahoo) 苹果 (/problem/?tags=apple) 优步 (/problem/?tags=uber) 两根指针 (/problem/?tags=two-pointers)  
数组 (/problem/?tags=array) hash table (/problem/?tags=hash-table) sort (/problem/?tags=sort)

### 描述

给一个整数数组, 找到两个数使得他们的和等于一个给定的数 *target*。

你需要实现的函数 twoSum 需要返回这两个数的下标, 并且第一个下标小于第二个下标。注意这里下标的范围是 0 到  $n-1$ 。

❗ 你可以假设只有一组答案。

### 样例

Example1:  
给出 numbers = [2, 7, 11, 15], target = 9, 返回 [0, 1].  
Example2:  
给出 numbers = [15, 2, 7, 11], target = 9, 返回 [1, 2].

### 挑战

给自己加点挑战

- $O(n)$  空间复杂度,  $O(n \log n)$  时间复杂度,
- $O(n)$  空间复杂度,  $O(n)$  时间复杂度,

在线评测地址: <https://www.lintcode.com/problem/two-sum/> (<https://www.lintcode.com/problem/two-sum/>)

收起题目描述 ^

语言类型 ALL (37) python (14) java (9) cpp (8) javascript (4) golang (2)

上传题解



令狐冲

更新于 12/21/2020, 2:22:26 AM

Given an array of integers, find two numbers such that they add up to a specific target number.

The function twoSum should return indices of the two numbers such that they add up to the target, where index1 must be less than index2. Please note that your returned answers (both index1 and index2) are not zero-based.

You may assume that each input would have exactly one solution.

Input: numbers={2, 7, 11, 15}, target=9 Output: index1=1, index2=2

```
/**
 * 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 numbers : An array of Integer
     * @param target : target = numbers[index1] + numbers[index2]
     * @return : [index1 + 1, index2 + 1] (index1 < index2)
     * numbers=[2, 7, 11, 15], target=9
     * return [1, 2]
     */
    public int[] twoSum(int[] numbers, int target) {
        //用一个hashmap来记录, key记录target-numbers[i]的值, value记录numbers[i]的i的值, 如果碰到一个
        //numbers[j]在hashmap中存在, 那么说明前面的某个numbers[i]和numbers[j]的和为target, i和j即为答案
        HashMap<Integer,Integer> map = new HashMap<>();

        for (int i = 0; i < numbers.length; i++) {
            if (map.get(numbers[i]) != null) {
                int[] result = {map.get(numbers[i]), i};
                return result;
            }
            map.put(target - numbers[i], i);
        }

        int[] result = {};
        return result;
    }
}
```

👍 获赞 7

💬 4 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 11/27/2020, 2:39:29 AM

Given an array of integers, find two numbers such that they add up to a specific target number.

The function twoSum should return indices of the two numbers such that they add up to the target, where index1 must be less than index2. Please note that your returned answers (both index1 and index2) are not zero-based.

You may assume that each input would have exactly one solution.

Input: numbers={2, 7, 11, 15}, target=9 Output: index1=1, index2=2

```
/**
 * 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(object):
    def twoSum(self, nums, target):
        #hash用于建立数值到下标的映射
        hash = {}
        #循环nums数值, 并添加映射
        for i in range(len(nums)):
            if target - nums[i] in hash:
                return [hash[target - nums[i]], i]
            hash[nums[i]] = i
        #无解的情况
        return [-1, -1]
```

👍 获赞 6

💬 2 条评论



令狐冲

更新于 11/27/2020, 2:58:37 AM

使用双指针算法。时间复杂度  $O(n \log n)$  额外空间复杂度  $O(n)$

```
/**
 * 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
 */
public class Solution {
    class Pair implements Comparable<Pair> {
        int number, index;

        public Pair(int number, int index) {
            this.number = number;
            this.index = index;
        }

        public int compareTo(Pair other) {
            return number - other.number;
        }
    }
    /**
     * @param numbers: An array of Integer
     * @param target: target = numbers[index1] + numbers[index2]
     * @return: [index1, index2] (index1 < index2)
     */
    public int[] twoSum(int[] numbers, int target) {
        int[] result = {-1, -1};

        if (numbers == null) {
            return result;
        }

        Pair[] pairs = getSortedPairs(numbers);

        int left = 0, right = pairs.length - 1;
        while (left < right) {
            if (pairs[left].number + pairs[right].number < target) {
                left++;
            } else if (pairs[left].number + pairs[right].number > target) {
                right--;
            } else {
                result[0] = Math.min(pairs[left].index, pairs[right].index);
                result[1] = Math.max(pairs[left].index, pairs[right].index);
                return result;
            }
        }

        return result;
    }

    private Pair[] getSortedPairs(int[] numbers) {
        Pair[] pairs = new Pair[numbers.length];
        for (int i = 0; i < numbers.length; i++) {
            pairs[i] = new Pair(numbers[i], i);
        }
        Arrays.sort(pairs);

        return pairs;
    }
}
```

👍 获赞 5

💬 2 条评论



令狐冲

更新于 11/24/2020, 1:46:08 AM

使用双指针算法, 时间复杂度 ( $n\log n$ ), 空间复杂度  $O(n)$ 

```
/**
 * 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:
    """
    @param numbers: An array of Integer
    @param target: target = numbers[index1] + numbers[index2]
    @return: [index1, index2] (index1 < index2)
    """
    def twoSum(self, numbers, target):
        if not numbers:
            return [-1, -1]

        # transform numbers to a sorted array with index
        nums = [
            (number, index)
            for index, number in enumerate(numbers)
        ]
        nums = sorted(nums)

        left, right = 0, len(nums) - 1
        while left < right:
            if nums[left][0] + nums[right][0] > target:
                right -= 1
            elif nums[left][0] + nums[right][0] < target:
                left += 1
            else:
                return sorted([nums[left][1], nums[right][1]])

        return [-1, -1]
```

👍 获赞 4

💬 添加评论



令狐冲

更新于 6/9/2020, 7:03:51 AM

Given an array of integers, find two numbers such that they add up to a specific target number.

The function twoSum should return indices of the two numbers such that they add up to the target, where index1 must be less than index2. Please note that your returned answers (both index1 and index2) are not zero-based.

You may assume that each input would have exactly one solution.

Input: numbers={2, 7, 11, 15}, target=9 Output: index1=1, index2=2

```
/**
 * 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:
    vector<int> twoSum(vector<int>& nums, int target) {
        // hash[i]表示nums中数值为i的下标
        unordered_map<int, int> hash;
        vector<int> result;

        // 一边循环每个数, 一边加入hash表。
        for (int i = 0; i < nums.size(); i++) {
            if (hash.find(target - nums[i]) != hash.end()) {
                // target - nums[i]的下标更小, 放在前面
                result.push_back(hash[target - nums[i]]);
                result.push_back(i);
                return result;
            }
            hash[nums[i]] = i;
        }

        // 无解的情况
        result.push_back(-1);
        result.push_back(-1);
        return result;
    }
};
```

👍 获赞 2

💬 1 条评论



白助教

更新于 6/9/2020, 7:03:50 AM

双指针方法, 时间复杂度 $O(n \lg n)$

```
/**
 * 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
 */
public class Solution {
    /**
     * @param numbers: An array of Integer
     * @param target: target = numbers[index1] + numbers[index2]
     * @return: [index1 + 1, index2 + 1] (index1 < index2)
     */
    class Pair {
        Integer value;
        Integer index;

        Pair(Integer value, Integer index) {
            this.value = value;
            this.index = index;
        }
        Integer getValue() {
            return this.value;
        }
    }

    class ValueComparator implements Comparator<Pair> {

        @Override
        public int compare(Pair o1, Pair o2) {
            return o1.getValue().compareTo(o2.getValue());
        }
    }

    public int[] twoSum(int[] numbers, int target) {
        // write your code here
        //用一个pair数组记录每个numbers[i]的值和它的位置i, 防止排序后不知道该元素的位置
        Pair[] number = new Pair[numbers.length];
        for(int i=0;i<numbers.length;i++) {
            number[i] = new Pair(numbers[i], i);
        }
        //排序后使用双指针
        Arrays.sort(number, new ValueComparator());
        int L=0, R = numbers.length-1;
        while(L<R) {
            if( number[L].getValue() + number[R].getValue() == target) {
                int t1 = number[L].index;
                int t2 = number[R].index;
                int[] result = {Math.min(t1,t2), Math.max(t1,t2)};
                return result;
            }
            if( number[L].getValue() + number[R].getValue() < target) {
                L++;
            } else {
                R--;
            }
        }
        int[] res = {};
        return res;
    }
}
```

👍 获赞 2

💬 7 条评论



布偶什么的就是布偶

更新于 10/20/2020, 6:26:56 AM

一种python解法

```
/**
 * 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 numbers: An array of Integer
    @param target: target = numbers[index1] + numbers[index2]
    @return: [index1 + 1, index2 + 1] (index1 < index2)
    """
    def twoSum(self, numbers, target):
        for i, a in enumerate(numbers):
            for j, b in enumerate(numbers[i + 1 - len(numbers):]):
                if a + b == target:
                    return [i, j + i + 1]
        return [-1, -1]
```

👍 获赞 1

💬 6 条评论



九章-小原

更新于 8/19/2020, 11:33:22 PM

## 算法：hashmap

用map记录所有当前查找过的 `num[i]`，存下 `target-num[i]`

如果存在 `num[j]` 在map中说明存在一对 `i,j` 的使得 `num[i], num[j]` 和为 `target`,

即找到了答案

## 复杂度分析

- 时间复杂度  $O(n)$ 
  - $n$  为数组的大小
- 空间复杂度  $O(n)$ 
  - $n$  为数组的大小

## 算法：双指针

先对数组拷贝

然后对数组排序，在排序后的数组中利用双指针从左右向中间寻找

- 如果 `numbers[i] + numbers[j] == target` 说明找到答案
- 如果 `numbers[i] + numbers[j] < target` 说明当前和比答案小。左指针右移
- 如果 `numbers[i] + numbers[j] > target` 说明当前和比答案大。右指针左移

然后在拷贝数组中找到对应 `numbers[i]` 和 `numbers[j]` 的下标，对这两个下标排个序



# 复杂度分析

- 时间复杂度  $O(n\log n)$ 
  - $n$ 为数组的大小
- 空间复杂度  $O(n)$ 
  - $n$ 为数组的大小

java

c++

python

---

(/) 课程 (course/) 刷题课 (premium-course/) 1对1私教 (1on1/) 免费课 (Seminar) 刷题题解 (ow/sh) 成功案例 (更多...) (/accounts/profile/)

邀请  
与礼  
vitation/shz



👍 获赞 0      💬 1 条评论



李助教

更新于 6/9/2020, 7:04:27 AM

测试题题解 本题十分简单, 考察vector的枚举操作, 本参考程序使用了最基础算法, 时间复杂度为 $O(n^2)$

```
/**
 * 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 numbers: An array of Integer
     * @param target: target = numbers[index1] + numbers[index2]
     * @return: [index1 + 1, index2 + 1] (index1 < index2)
     */
    vector<int> twoSum(vector<int> &numbers, int target) {
        // write your code here
        //暴力枚举每一个i和j, 如果加起来和为target即为答案
        int l=numbers.size();
        int i,j;
        int flag=0;//flag作为找到答案后跳出的一个标记用变量
        for(i=0;i<l;++i)
        {
            for(j=i+1;j<l;++j)
            {
                if(numbers[i]+numbers[j]==target)
                {
                    flag=1;
                    break;
                }
            }
            if(flag)
                break;
        }
        vector<int> ans;
        ans.push_back(i);
        ans.push_back(j);
        return ans;
    }
};
```

👍 获赞 0

💬 1 条评论



华助教

更新于 6/9/2020, 7:04:26 AM

```

/**
 * 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 numbers: An array of Integer
 * @param target: target = numbers[index1] + numbers[index2]
 * @return: [index1 + 1, index2 + 1] (index1 < index2)
 */
const twoSum = function (numbers, target) {
  var map = []
  var i;
  var res;
  for (i = 0; i < numbers.length; i++) {
    if (map.hasOwnProperty(numbers[i])) {
      res = [];
      res[0] = map[numbers[i]];
      res[1] = i;
      return res;
    }
    map[target - numbers[i]] = i;
  }
  res = [];
  return res;
}

```

👍 获赞 0

💬 1 条评论



九章用户ER9X1U

更新于 6/9/2020, 7:03:48 AM

Use Two pointers to solve this problem.

```

/**
 * 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 numbers: An array of Integer
   * @param target: target = numbers[index1] + numbers[index2]
   * @return: [index1 + 1, index2 + 1] (index1 < index2)
   */
  public int[] twoSum(int[] numbers, int target) {
    // write your code here
    int[] nums2 = Arrays.copyOf(numbers, numbers.length);
    Arrays.sort(nums2);
    int index1 = 0;
    int index2 = nums2.length - 1;
    int a = 0, b = 0;
    int i = 0;
    int[] result = new int[2];
    while(index1 < index2){

```

```
        if(nums2[index1] + nums2[index2] > target){
            index2--;
        }
        else if(nums2[index1] + nums2[index2] < target){
            index1++;
        }
        else{
            a = nums2[index1];
            b = nums2[index2];
            break;
        }
    }

    if(a!=b){
        for(int i = 0; i < numbers.length; i++){
            if(a == numbers[i]){
                result[0] = i;
            }
            if(b == numbers[i]){
                result[1] = i;
            }
        }
    }

    if(a == b){
        for(i = 0; i < numbers.length; i++){
            if(a == numbers[i]){
                result[0] = i;
                break;
            }
        }
        for(int j = i+1; j < numbers.length; j++){
            if(b == numbers[j]){
                result[1] = j;
            }
        }
    }
    Arrays.sort(result);
    return result;
}
}
```

👍 获赞 3

💬 2 条评论

**九章用户IM5DVY**

更新于 6/9/2020, 7:03:47 AM

要返回的是原数组的index而不是排序之后的数组

所以 argsort 一下原数组, 获取到排序后原数组的 index

然后正常排序 Two Pointers 做

“ 目测我这个版本最简单？

```
/**
 * 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 numbers: An array of Integer
    @param target: target = numbers[index1] + numbers[index2]
    @return: [index1, index2] (index1 < index2)
    """
    def twoSum(self, numbers, target):
        # write your code here
        if not numbers or len(numbers) < 2:
            return [-1, -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
 */
arg_nums = [x for x, y in sorted(enumerate(numbers), key = lambda x: x[1])]
numbers.sort()
left, right = 0, len(numbers) - 1
while left <= right:
    if numbers[left] + numbers[right] == target:
        return sorted([arg_nums[left], arg_nums[right]])
    elif numbers[left] + numbers[right] < target:
        left += 1
    else:
        right -= 1
return [-1, -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
 */
```

👍 获赞 3      💬 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

### 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

[首页 \(/?skip\\_redirect=true\)](#) | [联系我们 \(mailto:info@jiuzhang.com\)](mailto:info@jiuzhang.com) | [加入我们 \(/joinus\)](#)

Copyright © 2013-2020 九章算法 浙ICP备19045946号-1  
(<http://www.miibeian.gov.cn/>)

商务合作: [fukesu@jiuzhang.com \(mailto:fukesu@jiuzhang.com\)](mailto:fukesu@jiuzhang.com)

 (<http://weibo.com/ninechapter>)  知  
(<https://www.zhihu.com/people/crackinterview/>)

(/)