

LintCode领扣题解 (/problem) / 两数和-小于或等于目标值 · Two Sum - Less than or equal to target

两数和-小于或等于目标值 · Two Sum - Less than or equal to target

中文

LintCode 版权所有 (/problem/?tags=lintcode-copyright)

两根指针 (/problem/?tags=two-pointers)

sort (/problem/?tags=sort)

描述

给定一个整数数组，找出这个数组中有多少对的和是小于或等于目标值。返回对数。

样例

例1:

```
输入: nums = [2, 7, 11, 15], target = 24.
输出: 5.
解释:
2 + 7 < 24
2 + 11 < 24
2 + 15 < 24
7 + 11 < 24
7 + 15 < 24
```

例2:

```
输入: nums = [1], target = 1.
输出: 0.
```

在线评测地址: <https://www.lintcode.com/problem/two-sum-less-than-or-equal-to-target/> (<https://www.lintcode.com/problem/two-sum-less-than-or-equal-to-target/>)

收起题目描述 ^

语言类型

ALL (16)

python (7)

java (6)

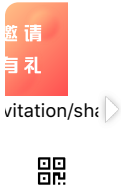
cpp (3)

上传题解

令狐冲

更新于 7/28/2020, 5:33:25 PM

排序后双指针即可。



```
/**
 * 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 an array of integer
     * @param target an integer
     * @return an integer
     */
    public int twoSum5(int[] nums, int target) {
        // Write your code here
        if (nums == null || nums.length < 2)
            return 0;

        Arrays.sort(nums);
        int cnt = 0;
        int left = 0, right = nums.length - 1;
        while (left < right) {
            int v = nums[left] + nums[right];
            if (v > target) {
                right--;
            } else {
                cnt += right - left;
                left++;
            }
        }
        return cnt;
    }
}
```

👍 获赞 3

💬 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



DDBear

更新于 7/28/2020, 5:33:10 PM

算法

算法一

暴力 N^2 枚举

算法二 二分查找

二分知识

如果序列有序, 则可以用一种更有效率的查找方法来查找序列中的记录, 这就是折半查找法, 又称为二分搜索。

折半查找的基本思想: 减少一半的查找序列的长度, 分而治之地进行关键字的查找。他的查找过程是: 先确定待查找记录的所在的范围, 然后逐渐缩小查找的范围, 直至找到该记录为止 (也可能查找失败)。

在最简单的形式中, 二分查找对具有指定左索引和右索引的连续序列进行操作。这就是所谓的查找空间。二分查找维护查找空间的左、右和中间指示符, 并比较查找目标或将查找条件应用于集合的中间值; 如果条件不满足或值不相等, 则清除目标不可能存在的那一半, 并在剩下的一半上继续查找, 直到成功为止。如果查以空的一半结束, 则无法满足条件, 并且无法找到目标。

算法思路

算法二在算法一的基础上进行改进

枚举一个数 $nums[i]$, 找有多少个 j $nums[i] + nums[j] \leq target$

可以先对 $nums$ 排序 然后使用二分查找的方式快速查找

算法三 双指针

利用双指针 $start$, end 求解

对于每个 $start$, 找到最大的 end , 使得 $A[start] + B[end]$ 小于等于 x

这样, 因为数组是单调的, 所以 $A[start] + B[end + 1] \dots A[start] + B[n - 1]$ 都大于 x

所以对于 $start$, 一共有 $end + 1$ 对 $a + b$ 的和小于等于 x 。

$start$ 移动到下一个位置, 再去求新的 end

因为数组元素都是单调的, 所以若 $A[start] + B[j]$ 大于 x , 则 $A[start + 1] + B[j]$ 一定也大于 x , 所以, 在 $start$ 移动到下一个位置时候, 我们不需要重新从开始位置求一遍 end , 而是在上一次求得 end_{last} 基础上继续减少就可以了。

证明如下:

$$A[oldstart] + B[oldend] \leq x$$

$$oldstart < newstart$$

$$\text{所以 } A[oldstart] + B[oldend + 1] > x$$

$$\text{推出 } A[newstart] + B[oldend + 1] > x$$

$$\text{所以 } newend \leq oldend$$

复杂度分析

- 空间复杂度
 - 三个算法都不需要多开辟空间, 因此空间复杂度为 $O(1)$ 。
- 时间复杂度
 - 算法一 暴力枚举 $O(N^2)$
 - 算法二 枚举 i , 用二分查找加速 j 的数量 时间复杂度 $O(N \log N)$
 - 算法三 双指针, $start$ 最多增加 n 次 end 最多减少 n 次 所以时间复杂度 $O(N)$

代码

算法二

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:
    /**
     * @param nums: an array of integer
     * @param target: an integer
     * @return: an integer
     */
    int twoSum5(vector<int> &nums, int target) {
        // write your code here

        //先对数组排序
        sort(nums.begin(), nums.end());
        //数组长度
        int n = nums.size();

        int ans = 0;
        //对于每个i, 二分找到最大的j nums[i]+nums[j]<=target
        for(int i = 0; i < n - 1; i++) {
            //确定二分上下界
            int left = i;
            int right = n;
            int pos = i;
            while(left + 1 < right) {
                int mid = left + (right - left) / 2;
                //不大于target 可以提高下界
                if(nums[mid] + nums[i] <= target) {
                    left = mid;
                    pos = mid;
                }
                //缩小上界
                else {
                    right = mid;
                }
            }
            //统计答案
            ans += pos - i;
        }
        return ans;
    }
};
```

算法三

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:
    /**
     * @param nums: an array of integer
     * @param target: an integer
     * @return: an integer
     */
    int twoSum5(vector<int> &nums, int target) {
        // write your code here

        //先对数组排序
        sort(nums.begin(), nums.end());
        //数组长度
        int n = nums.size();
        //双指针上下界
        int start = 0, end = n - 1;
        //答案
        int ans = 0;
        while(start <= n - 1) {
            //找到最小的end nums[start]+nums[end]<=target
            while(end >= 0) {
                if(nums[start] + nums[end] > target) {
                    end -= 1;
                } else {
                    break;
                }
            }
            //只记录start<end的情况
            if(end > start) {
                ans += end - start;
            } else {
                break;
            }
            start += 1;
        }
        return ans;
    }
};
```

👍 获赞 1

💬 2 条评论



令狐冲

更新于 6/13/2020, 1:35:26 PM

排序后双指针即可。

```
/**
 * 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:
    # @param nums {int[]} an array of integer
    # @param target {int} an integer
    # @return {int} an integer
    def twoSum5(self, nums, target):
        # Write your code here
        l, r = 0, len(nums)-1
        cnt = 0
        nums.sort()
        while l < r:
            value = nums[l] + nums[r]
            if value > target:
                r -= 1
            else:
                cnt += r - l
                l += 1
        return cnt
```

👍 获赞 1

💬 2 条评论



令狐冲

更新于 6/9/2020, 7:04:28 AM

排序后双指针即可。

```
/**
 * 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 an array of integer
     * @param target an integer
     * @return an integer
     */
    int twoSum5(vector<int> &nums, int target) {
        // Write your code here
        int n = nums.size(), cnt = 0;
        sort(nums.begin(), nums.end());
        int l = 0, r = nums.size() - 1;
        while (l < r){
            if (nums[l] + nums[r] > target){
                r--;
            } else {
                cnt += r - l;
                l++;
            }
        }
        return cnt;
    }
};
```

👍 获赞 0

💬 添加评论

**jessiew**

更新于 10/11/2020, 9:56:00 PM

一种Binary Search的算法, 时间复杂度为 $O(n \log n)$, 比双指针worse case $O(n^2)$ 更快一点。算法: 1.outer loop $O(n)$: i从右往左遍历 2.nested loop $O(\log n)$: 在i+1到end中binary search从右到左第一个小于等于target - numsi ()的数end, 找不到返回-1 3.如果找到这个数, count += end - i

```
/**
 * 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 twoSum5(int[] nums, int target) {
        // Edge cases
        if (nums == null || nums.length < 2) return 0;

        // Two sum problem (count all duplicates)
        int count = 0;
        Arrays.sort(nums);

        // Binary search the first smaller or equal number from (nums.length - 1) to i + 1
        for (int i = nums.length - 2; i >= 0; i--) {
            int end = bsFirstSmallerOrEqual(nums, i + 1, target - nums[i]);
            if (end != -1) count += end - i;
        }

        return count;
    }

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

👍 获赞 3

💬 添加评论



九章用户F263EI

更新于 6/9/2020, 7:03:49 AM

最暴力的解法是遍历每一对数字, 若其和不超过target且之前没有查看过这对pair, 结果就+1, 时间复杂度 $O(n^2)$; 稍微优化一点是排序后双指针, 排序并不影响结果。排完序之后, 是对每一个元素 (设其index为right), 去遍历它的 每一个左边的元素 (设其index为left), 如果 $\text{numsleft}() + \text{numsright}() \leq \text{target}$, 结果就+1。这样避免了去查看right后面的数据。其实当right往左走时, left只需往右走, 不需再去查看left之前的数, 因为它们已经满足了 $\text{numsright}() + \text{numsleft}() \leq \text{target}$ 。


```
/**
 * 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
 */
class Solution:
    def twoSum5(self, nums, target):
        ans = 0
        nums.sort()
        left, right = 0, len(nums) - 1
        while left <= right:
            while left < right and nums[left] + nums[right] <= target:
                left += 1
            ans += left
            right -= 1
        ans += left * (left - 1) / 2
        return ans
```

👍 获赞 2

💬 2 条评论



九章用户BHJVP5

更新于 12/23/2020, 2:37:59 AM

双指针并没有考虑去重的问题, 剩下的是凑字了 凑字凑字凑字凑字 凑字凑字凑字

```
/**
 * 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
 */
public class Solution {
    /**
     * @param nums: an array of integer
     * @param target: an integer
     * @return: an integer
     */
    public int twoSum5(int[] nums, int target) {
        // write your code here
        if (nums == null || nums.length < 2) {
            return 0;
        }

        Arrays.sort(nums);
        int result = 0;

        for (int i = 0; i < nums.length - 1; i++) {
            int right = nums.length - 1;
            while (i < right) {
                if (nums[i] + nums[right] <= target) {
                    result++;
                }
                right--;
            }
        }

        return result;
    }
}
```

👍 获赞 1

💬 2 条评论

**Tin**

更新于 6/9/2020, 7:03:54 AM

两种解法。

第一种, 遍历左下标, 右下标用二分查找求出。和 Nepenthes 的解法类似, 但写在白板上更好看。bisect 的源程序要去看并能默写出来。时间复杂度是两个 $n\log N$, 排序一个, 遍历+二分一个。

第二种, 经典双指针。时间复杂度是一个 $n\log N$, 一个 N , 排序一个, 双指针遍历一个。用Java的人们竟然认为双指针复杂度是 $O(N^2)$, 万不可被误导。

```
/**
 * 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
 */
import bisect
class SolutionBisect:
    """
    @param nums: an array of integer
    @param target: an integer
    @return: an integer
    """
    def twoSum5(self, nums, target):

        nums.sort()

        totalPairs = 0

        for i, num in enumerate(nums):
            otherNum = target - num
            j = bisect.bisect(nums, otherNum, i)
            if j == i:
                break
            totalPairs += j - 1 - i

        return totalPairs

class SolutionTwoPointers:

    def twoSum5(self, nums, target):

        nums.sort()

        totalPairs = 0

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

        return totalPairs
```

👍 获赞 1

💬 添加评论

**九章用户U6HP8S**

更新于 6/9/2020, 7:03:54 AM

Time: O(nlogn) Space: O(1)

紫薯紫薯紫薯紫薯

```
/**
 * 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: an array of integer
     * @param target: an integer
     * @return: an integer
     */
    public int twoSum5(int[] nums, int target) {
        if (nums == null && nums.length <= 1) return 0;
        Arrays.sort(nums); // O(nlogn)
        int i = 0, j = nums.length - 1;
        int ans = 0;
        while (i < j) {
            int vi = nums[i], vj = nums[j];
            if (vi + vj <= target) {
                // if we know nums now is sorted and vi + vj <= target
                // then all k in range [i+1, j] has: vi + vk <= target
                ans += j - i;
                // now try to increase vi a little bit
                i++;
            } else {
                // we know current value of vj can't make vi + vj <= target
                // so it is safe for us to skip all duplicate values of vj
                while (i < j && nums[j] == vj) j--;
            }
        }
        return ans;
    }
}
```

👍 获赞 1

💬 添加评论

**The Art of Racing in the Rain**

更新于 6/9/2020, 7:03:52 AM


$\log(n)$ 时间复杂度的解法。先对数组进行sort, 然后一个指针从左往右, 另一个指针通过二分法找到从哪个数开始和另一个指针上的数的和超过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
 */
class Solution:
    """
    @param nums: an array of integer
    @param target: an integer
    @return: an integer
    """
    def twoSum5(self, nums, target):
        # write your code here

        if len(nums) <= 1:
            return 0

        nums.sort()
        count = 0
        for i in range(1, len(nums)):
            l, r = 0, i - 1
            while l + 1 < r:
                m = l + (r - l) // 2
                if nums[m] + nums[i] <= target:
                    l = m
                else:
                    r = m

            if nums[r] + nums[i] <= target:
                count += r + 1
            elif nums[l] + nums[i] <= target:
                count += r
        return count
```

 获赞 1 添加评论**Mo**

更新于 6/23/2020, 1:45:05 PM

```
/**
 * 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: an array of integer
    @param target: an integer
    @return: an integer
    """
    def twoSum5(self, nums, target):
        if not nums or len(nums) < 2:
            return 0

        nums.sort()
        result = 0
        i = 0
        j = len(nums) - 1
        while i < j:
            while i < j and nums[i] + nums[j] > target:
                j -= 1

            result += j - i
            i += 1

        return result
```

👍 获赞 0

💬 添加评论



辛同学

更新于 6/9/2020, 7:04:23 AM

双指针算法, 第二个循环做了重复计算的优化 理论上不会达到 $O(N^2)$;比较简洁。

```
/**
 * 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: an array of integer
     * @param target: an integer
     * @return: an integer
     */
    public int twoSum5(int[] nums, int target) {
        // write your code here
        if (nums == null || nums.length == 0) {
            return 0;
        }
        int count = 0;
        int i, j;
        for (i = 0; i < nums.length - 1; ++i) {
            for(j = i + 1; j < nums.length; ++j) {
                if(nums[i] + nums[j] <= target) {
                    count++;
                }
            }
        }
        return count;
    }
}
```

👍 获赞 0

💬 1 条评论

[加载更多题解](#)

进阶课程

直播+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

(/)