

数学、计算几何、位运算常见问题详解



扫描二维码关注微信/微博
获取最新面试题及权威解答

微信: [ninechapter](#)

知乎专栏: <http://zhuannlan.zhihu.com/jiuzhang>

微博: <http://www.weibo.com/ninechapter>

官网: www.jiuzhang.com

- 在网格图、矩阵图、棋盘上做多个方向扩展时，用_____数组会让程序写起来更方便（填空）
- 多源点单终点最短路（多源单终）一般可以转换成____源____终的最短路问题（填空）
- 我们一般增加一个_____使得多源点多终点最短路转换成单源点多终点最短路（填空）
- 第五节课中我们一共讲了几种**DFS**记录状态的方法？

- 矩阵上的问题 (2题)
- 数位分离&进制转换 (2题)
- 高精度运算 (4题)
- 快速点题 (9题)

矩阵上的问题

Rotate Image

<http://www.lintcode.com/zh-cn/problem/rotate-image/>

<http://www.jiuzhang.com/solutions/rotate-image/>

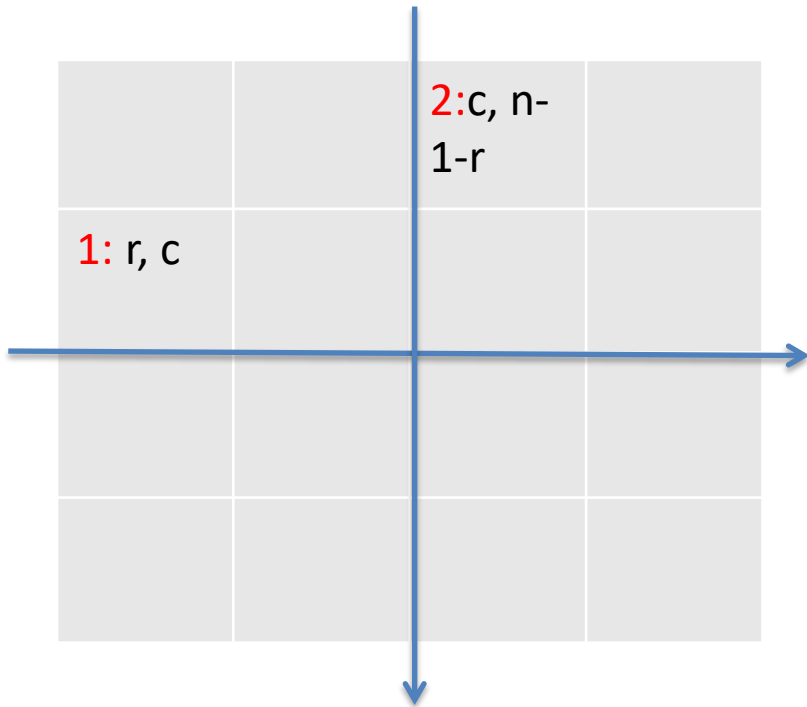
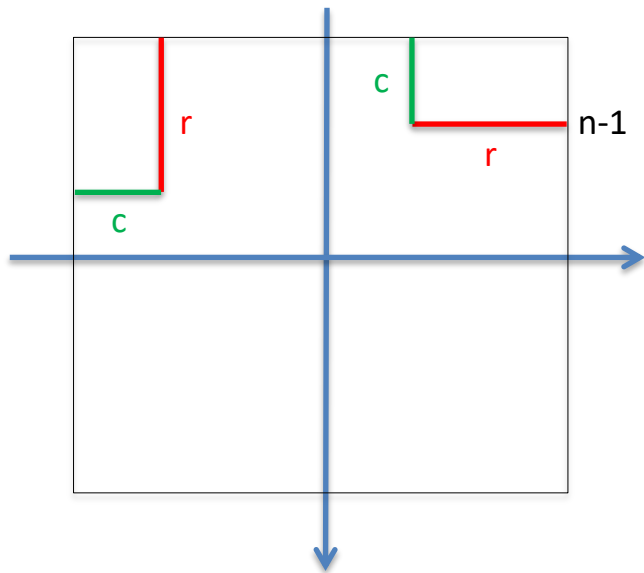
思路:

- 直观的印象:
 - 图像中任何一个点, 连转4次一定会转回来
 - 所以一点包括它自己一共有4个相关的点
 - 一次旋转就是 $1 \rightarrow 2$ $2 \rightarrow 3$ $3 \rightarrow 4$ $4 \rightarrow 1$

Rotate Image

- 假设矩阵n行n列，第1个点在r行c列 (r, c)，怎样推出旋转后的对应点？

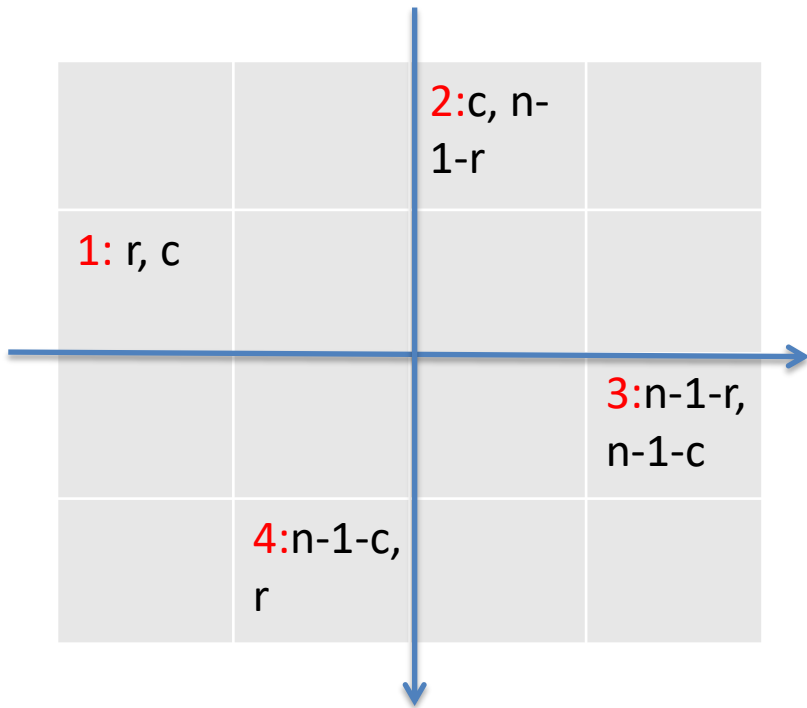
旋转: $(r, c) \rightarrow (c, n - 1 - r)$



- 方法一：推出对应的4个点的坐标

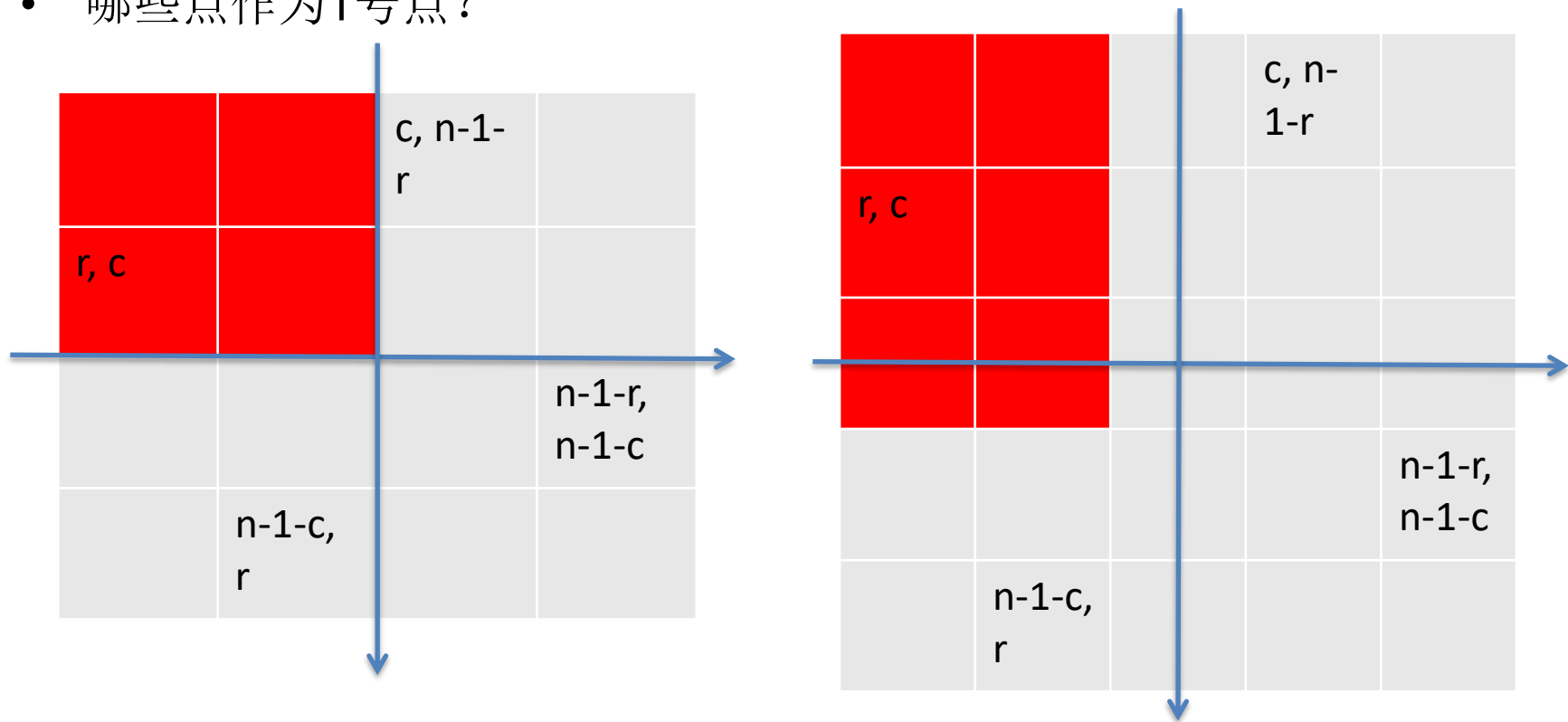
旋转: $(r, c) \rightarrow (c, n - 1 - r)$

- 1 (r, c)
- 2 $(c, n - 1 - r)$
- 3 $(n - 1 - r, n - 1 - c)$
- 4 $(n - 1 - c, r)$



Rotate Image

- 哪些点作为1号点?



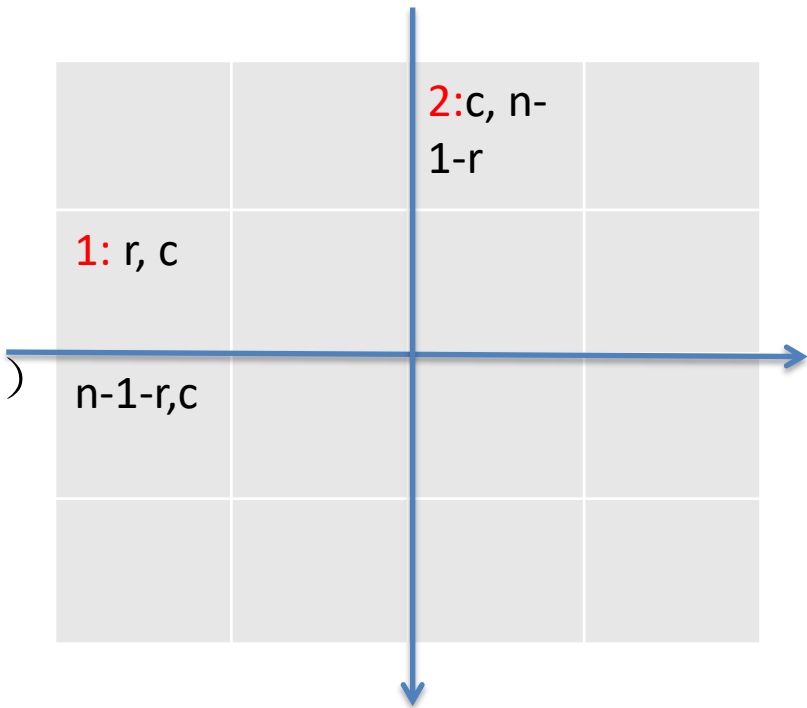
Rotate Image

- 方法二：仔细看看旋转公式

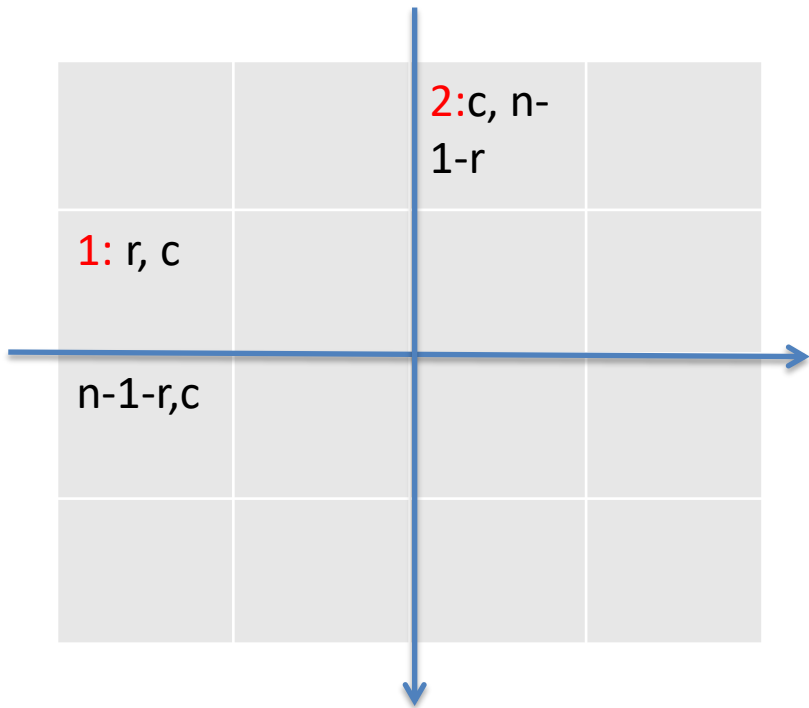
旋转: $(r, c) \rightarrow (c, n - 1 - r)$

上下翻转: $(r, c) \rightarrow (n - 1 - r, c)$

旋转 = 上下翻转后 + 交换 x, y (转置)



- 方法二：仔细看看旋转公式
- 旋转方法总结：
 - 顺时针90：先上下，再对角线
 - 逆时针90：先左右，再对角线



Rotate Image

- Company Tags: Amazon

考点:

- 旋转公式的计算

能力维度：

5. 细节处理（corner case）

Sparse Matrix Multiplication

<http://www.lintcode.com/zh-cn/problem/sparse-matrix-multiplication/>

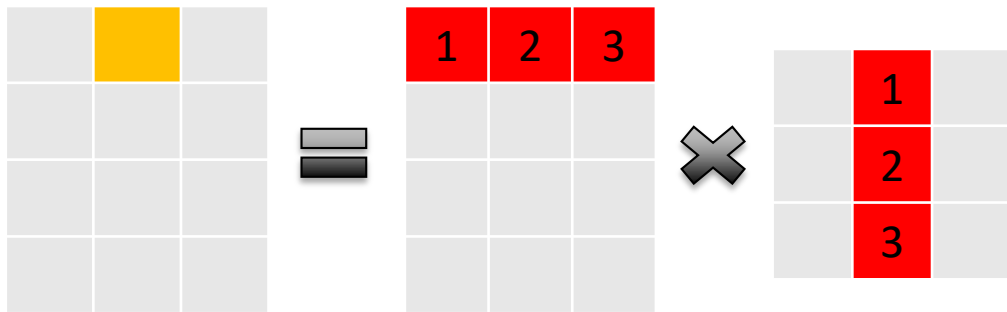
<http://www.jiuzhang.com/solutions/sparse-matrix-multiplication/>

Sparse Matrix Multiplication

思路:

- 矩阵乘法是怎么样子的？

$$\begin{aligned} - C &= A * B \\ - [n, m] & \quad [n, t] \quad [t, m] \\ - C[i][j] &= A[i][k] * B[k][j] \quad (1 \leq k \leq t) \end{aligned}$$



- 常规实现时间复杂度？
 - $O(nmt) \approx O(n^3)$
 - 见代码
- 如果是稀疏（很多0）的时候怎么办？以什么样的循环顺序？
 - $n \ t \ m$
 - 见代码
- 进一步还可以怎样优化？

Sparse Matrix Multiplication

- 记录B的一行中非0元素的列数

0	0	3
2	0	6
0	5	0
1	2	4

2		
0	2	
1		
0	1	2

- 优化后的时间复杂度?
 - $O(nmt) \approx O(n^3)$
- 优化了什么东西?
 - 常数项时间

- Company Tags: LinkedIn Facebook

考点:

- 矩阵乘法实现时的优化

能力维度:

3. 基础数据结构/算法
4. 逻辑思维/算法优化能力

数位分离&进制转换

Integer to Roman

<http://www.lintcode.com/zh-cn/problem/integer-to-roman/>

<http://www.jiuzhang.com/solutions/integer-to-roman/>

Integer to Roman

- 1~9: {"I", "II", "III", "IV", "V", "VI", "VII", "VIII", "IX"};
- 10~90: {"X", "XX", "XXX", "XL", "L", "LX", "LXX", "LXXX", "XC"};
- 100~900: {"C", "CC", "CCC", "CD", "D", "DC", "DCC", "DCCC", "CM"};
- 1000~3000: {"M", "MM", "MMM"}.

Example:

- 421= CDXXI

思路:

- 数位分离之后直接转换
- 如何数位分离? %10 /10
- (扩展) 如何将一个数转成k进制? % k / k

◆ 小技巧总结:

- 如何数位分离? $\%10$ $/10$
- (扩展) 如何将一个数转成k进制? $\% k$ $/ k$

能力维度:

1. 理解问题
3. 基础数据结构/算法

Pow(x, n)

<http://www.lintcode.com/zh-cn/problem/powx-n/>

<http://www.jiuzhang.com/solutions/powx-n/>

思路:

- 普通求幂的时间复杂度 $O(n)$
- 怎样更快的求幂?

问: 7可以分解成哪几个数的和?

- $x^1 = x^{(1)} = x^1$
- $x^2 = x^{(10)} = x^2$
- $x^3 = x^{(11)} = x^2 * x^1$
- $x^4 = x^{(100)} = x^4$
- $x^5 = x^{(101)} = x^4 * x^1$
- $x^6 = x^{(110)} = x^4 * x^2$

- $x^1 = x^1$
- $(x^1)^2 = x^2$
- $(x^2)^2 = x^4$
- $(x^4)^2 = x^8$

十进制转二进制 %2 /2
时间复杂度 $O(\log n)$

Pow(x, n)

- Company Tags: LinkedIn Google Facebook

考点:

- 基础算法的理解与实现

能力维度:

- 3. 基础数据结构/算法
- 5. 细节处理 (corner case)

高精度运算

- 什么是高精度？
- 怎样表示高精度数？

8	7	6	5	4	3	2	1	0
				2	8	7	0	9
					6	8	9	6

- 高精度加法的实现？
 - 方法1：每次模拟进位（见代码）
 - 方法2：对应数位相加后一次性进位

8	7	6	5	4	3	2	1	0
				2	8	7	0	9
					6	8	9	6

- 高精度乘法的实现？
 - 数位相乘 $\text{ans}[i + j] += a[i] * b[j]$
 - 乘完后一次性进位（见代码）

8	7	6	5	4	3	2	1	0
				2	8	7	0	9
					6	8	9	6

Add Strings

<http://www.lintcode.com/problem/add-strings/>

<http://www.jiuzhang.com/solution/add-strings/>

Add Binary

<http://www.lintcode.com/zh-cn/problem/add-binary/>

<http://www.jiuzhang.com/solutions/add-binary/>

Add Two Numbers

<http://www.lintcode.com/zh-cn/problem/add-two-numbers/>

<http://www.jiuzhang.com/solutions/add-two-numbers/>

Multiply Strings

<http://www.lintcode.com/problem/multiply-strings/>

<http://www.jiuzhang.com/solution/multiply-strings/>

- Rotate Image
- Sparse Matrix Multiplication
- Integer to Roman
- Pow(x, n)
- Add Strings
- Add Binary
- Add Two Numbers
- Multiply Strings
 - 一次性进位

快速点题

- <http://lintcode.com/zh-cn/problem/trailing-zeros/>
- $11! = 39916800$, 返回 2
- 想想末尾的0由哪两个因子构成
- [Solution](#) Company Tags: Adobe

- <http://lintcode.com/zh-cn/problem/roman-to-integer/>
 - CDXXI = 421
 - 与Integer to Roman对应
 - 简单模拟题
 - [Solution](#)
- Company Tags: Google Amazon Facebook LinkedIn

- <http://lintcode.com/zh-cn/problem/sqrtx-ii/>
 - 给出 $n = 2$
 - 返回 1.41421356
 - 与Pow(x, n)对应
 - 二分基础练习题
 - [Solution](#)
- Company Tags: Facebook

Reverse Integer

- <http://lintcode.com/zh-cn/problem/reverse-integer/>
 - 给定 $x = 123$, 返回 321
 - 给定 $x = -123$, 返回 -321
 - 数位分离练习题
 - [Solution](#)
- Company Tags: Google Facebook

- <http://lintcode.com/zh-cn/problem/palindrome-number/>
- 11, 121, 1, 12321 是回文数
- 23, 32, 1232 不是回文数
- 数位分离练习题
- [Solution](#) Company Tags: Google Amazon

Add Digits

- <http://lintcode.com/zh-cn/problem/add-digits/>
- 给出 $\text{num} = 38$
- 相加的过程如下: $3 + 8 = 11$, $1 + 1 = 2$
- 数位分离练习题
- [Solution](#)

Company Tags: Amazon

- <http://lintcode.com/zh-cn/problem/happy-number/>

- 当 $n=19$ $1^2 + 9^2 = 82$ $8^2 + 2^2 = 68$
- $6^2 + 8^2 = 100$ $1^2 + 0^2 + 0^2 = 1$

- 如何判断无限循环？
- 数位分离练习题

- [Solution](#)

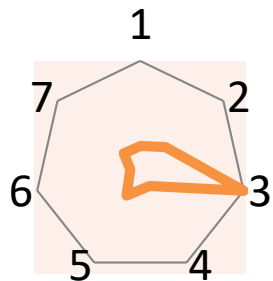
Company Tags: Apple Pinterest

- <http://lintcode.com/zh-cn/problem/binary-representation/>
- $n = "3.5"$, 返回 $"11.1"$
- $n = "3.72"$, 返回 $"ERROR"$
- 思考怎样将小数转换为二进制
- 进制转换练习题
- [Solution](#)

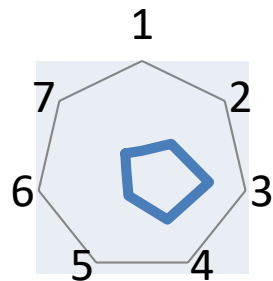
- <http://lintcode.com/zh-cn/problem/plus-one/>
 - [1,2,3] 表示 123, 返回 [1,2,4]
 - [9,9,9] 表示 999, 返回 [1,0,0,0]
 - 输入为数组的高精度加法
 - [Solution](#)
- Company Tags: Google

1. 理解问题
2. 代码基础功力
3. 基础数据结构/算法
4. 逻辑思维/算法优化能力
5. 细节处理（**corner case**）
6. 算法分析（时间/空间复杂度）
7. debug能力

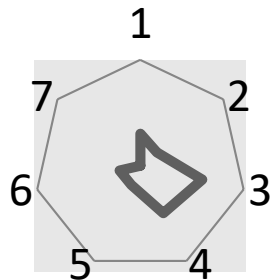
LinkedIn



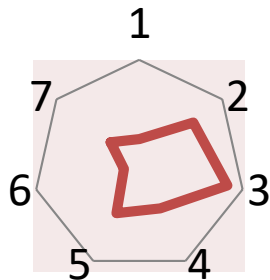
Facebook



Amazon



Google



- <http://www.jiuzhang.com/course/9/questionnaire>



扫描二维码关注微信/微博
获取最新面试题及权威解答

微信: [ninechapter](#)

知乎专栏: <http://zhuankan.zhihu.com/jiuzhang>

微博: <http://www.weibo.com/ninechapter>

官网: www.jiuzhang.com