

# 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](http://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（3题）
- 高精度运算（4题）
- 快速幂（1题）

# 矩阵上的问题

## Search a 2D Matrix II

<http://www.lintcode.com/zh-cn/problem/search-a-2d-matrix-ii/>

<http://www.jiuzhang.com/solutions/search-a-2d-matrix-ii/>

思路一：

- 假设不考虑矩阵从上到下，从左到右排序的特性
- 把矩阵的所有元素扫一遍，一个一个的找，看target出现了多少次
- 时间复杂度？
  - $O(n*m)$

思路二：

- 把大小关系画出来看看有什么特点？
  - L字型的大小顺序，顺着L字型从上往下依次变大
- 仔细观察，哪里可以作为突破口？
- 左下角这个元素非常特殊(记为x)
  - 如果 $\text{target} < x$  最后的一行还有继续比较的意义吗？
  - 如果 $\text{target} > x$  最左的一列还有继续比较的意义吗？
  - 如果 $\text{target} = x$ 最后一行 & 最左的一列还有继续比较的意义吗？

[1, 3, 5, 7],

[2, 4, 7, 8],

[3, 5, 9, 10]

- 所以算法是:

- $\text{target} < x$  砍掉最后一行
- $\text{target} > x$  砍掉最左一列
- $\text{target} = x$  计数+1, 砍掉最后一行&砍掉最左一列

[1, 3, 5, 7],

[2, 4, 7, 8],

- 时间复杂度 $O(n+m)$

- 盯住哪个元素会成为左下角元素 $x$ ,  $x$ 每次向上 or 向右 or 右上走一步
- 在矩阵中最多只能走 $n+m$ 步

[3, 5, 9, 10]



- Company Tags: Amazon Google

考点:

- 通过给出矩阵的特点发现规律
- 二分法思想的借鉴

能力维度:

- 4. 逻辑思维/算法优化能力
- 6. 算法分析（时间/空间复杂度）

## Rotate Image

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

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

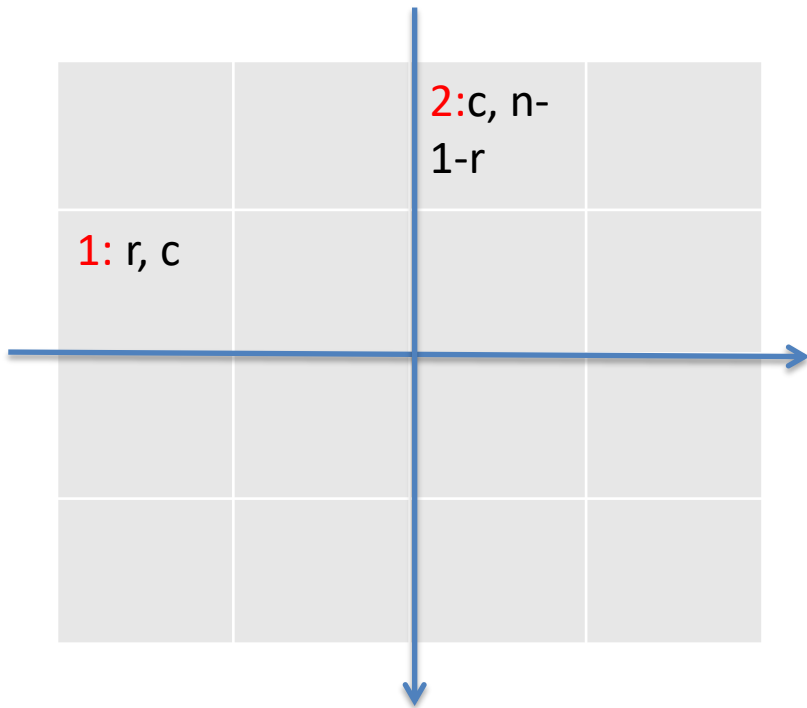
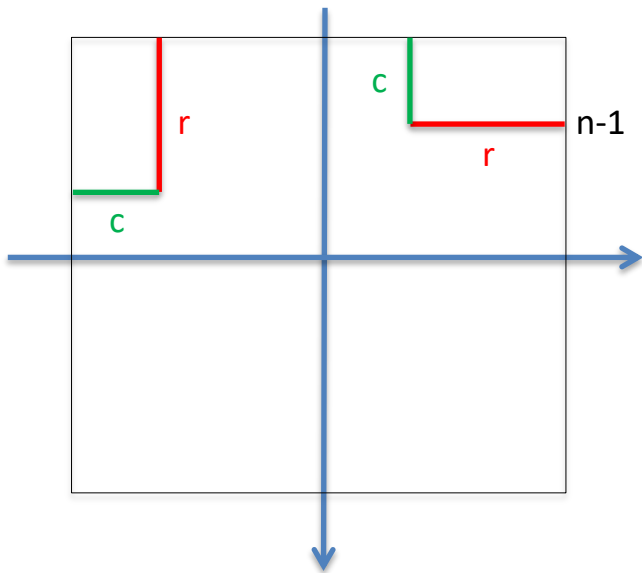
思路:

- 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m列，第1个点在r行c列 (r, c)，怎样推出旋转后的对应点？

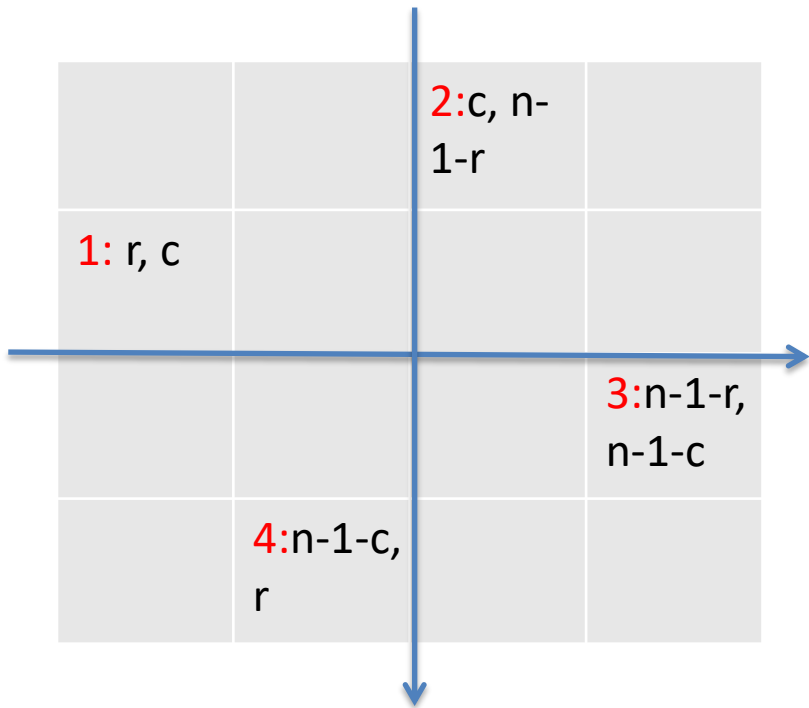
旋转:  $(r, c) \rightarrow (c, n - 1 - r)$



- 方法一：推出对应的4个点的坐标

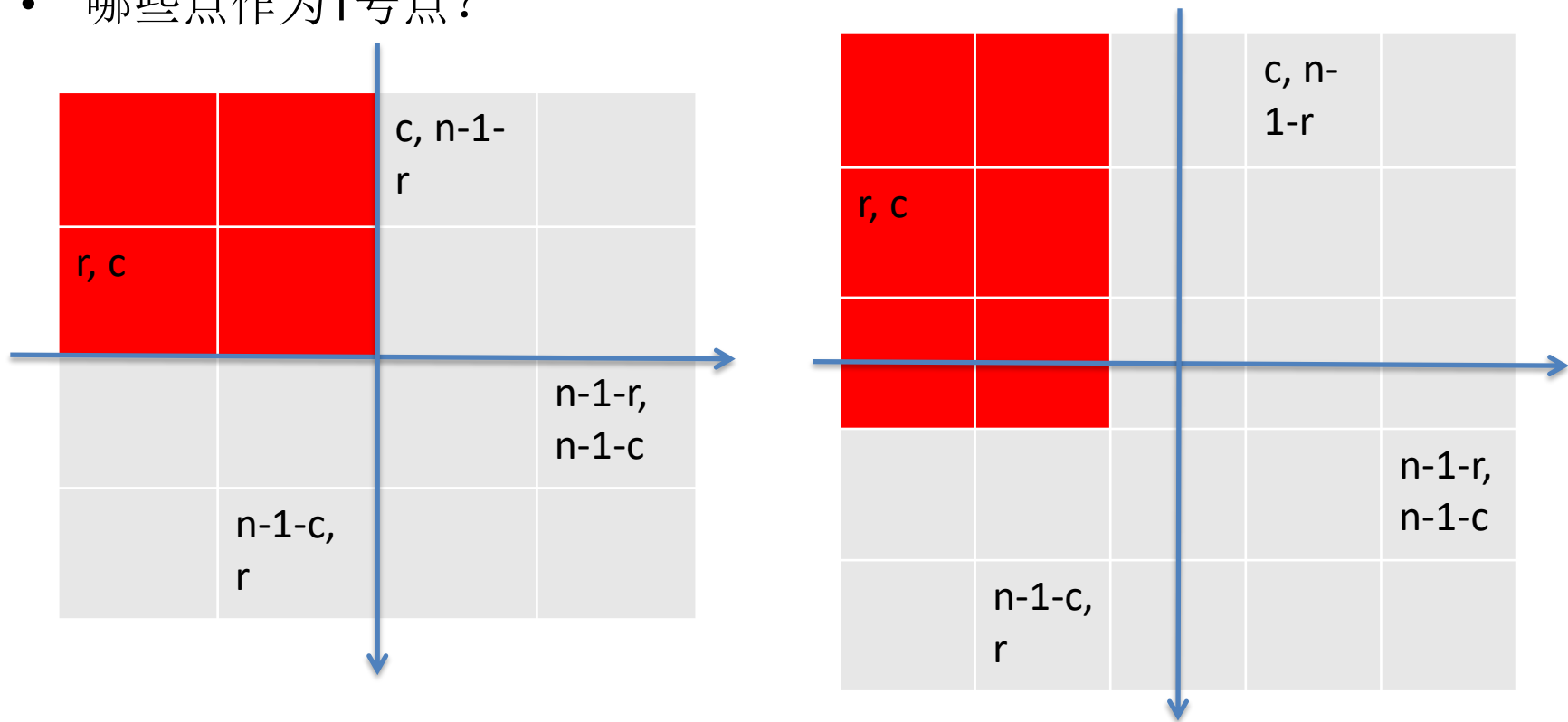
旋转:  $(r, c) \rightarrow (c, n - 1 - r)$

- 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?



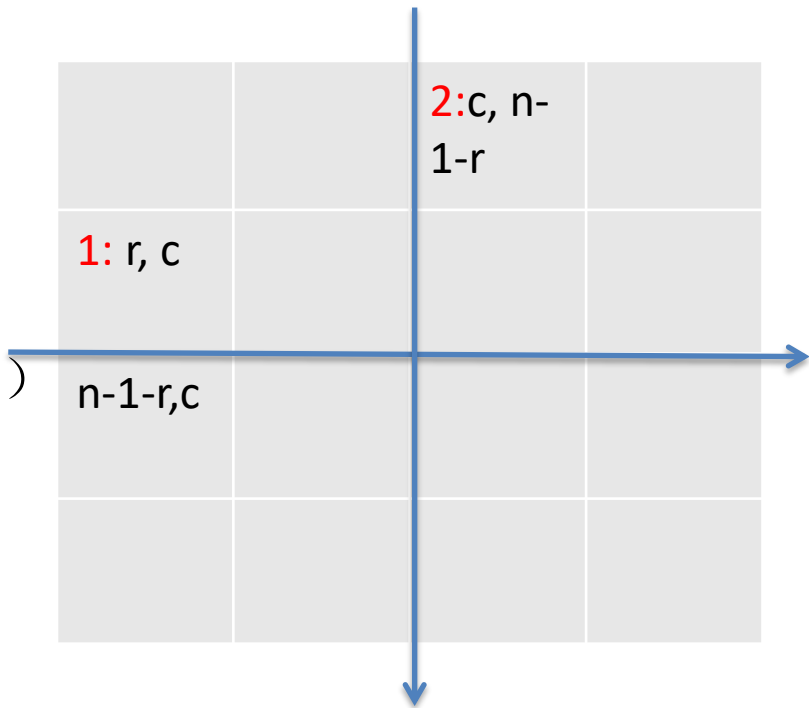
# Rotate Image

- 方法二：仔细看看旋转公式

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

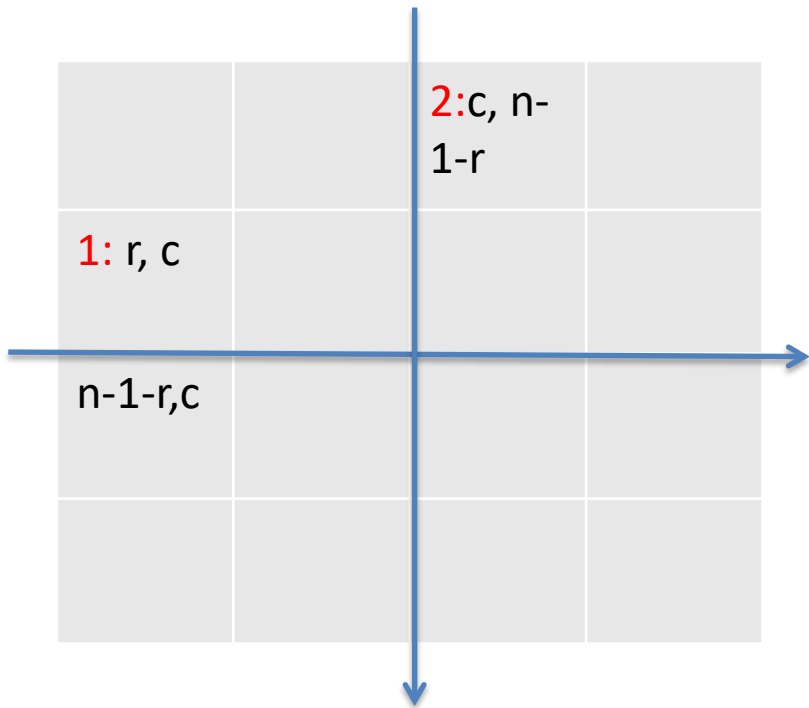
上下翻转:  $(r, c) \rightarrow (n - 1 - r, c)$

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



- 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/>

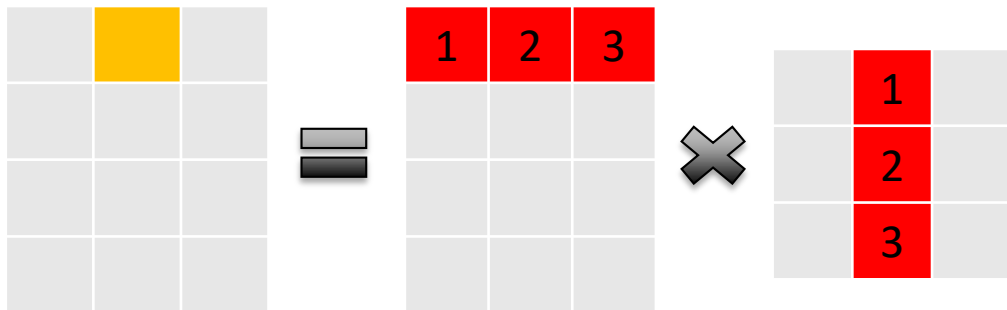
<http://www.jiuzhang.com/solutions/sparse-matrix-multiplication/>

# Sparse Matrix Multiplication

思路:

- 矩阵乘法是怎么样子的?

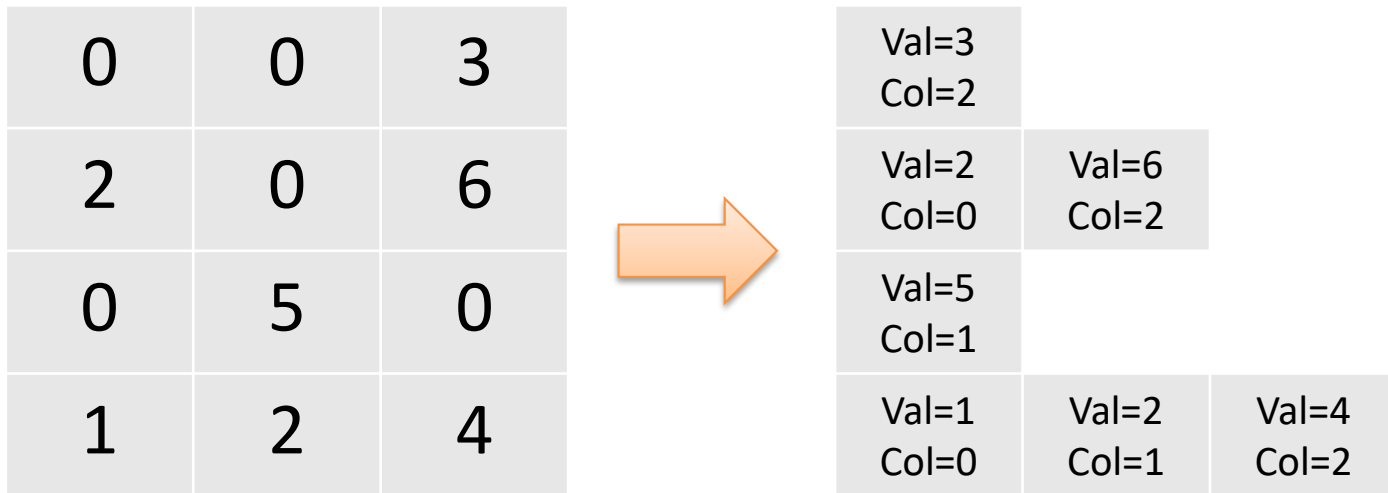
$$\begin{aligned} - C &= A * B \\ - [n, m] & \quad [n, t] \quad [t, m] \\ - C[i][j] &= A[i][k] * B[k][j] \quad (1 \leq k \leq t) \end{aligned}$$



- 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

- 去掉B的一行中为0 的元素



- 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

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

# 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

## Big Integer Addition

<http://www.lintcode.com/zh-cn/problem/big-integer-addition/>

<http://www.jiuzhang.com/solutions/big-integer-addition/>



## 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/>

## Big Integer multiplication

<http://www.lintcode.com/zh-cn/problem/big-integer-multiplication/>

<http://www.jiuzhang.com/solutions/big-integer-multiplication/>

# 快速幂

## 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:

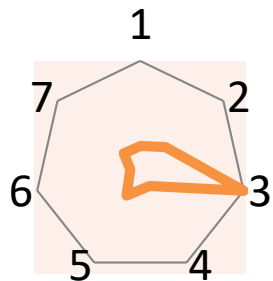
- 3. 基础数据结构/算法
- 5. 细节处理 (corner case)



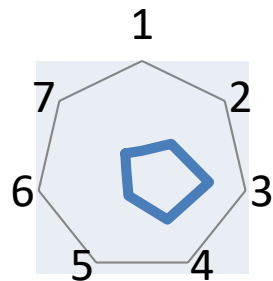
- Search a 2D Matrix II
- Rotate Image
- Sparse Matrix Multiplication
  
- Big Integer Addition
- Add Binary
- Add Two Numbers
- Big Integer multiplication
  - 一次性进位
- Pow(x, n)

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

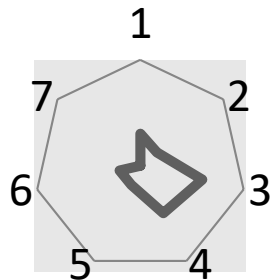
LinkedIn



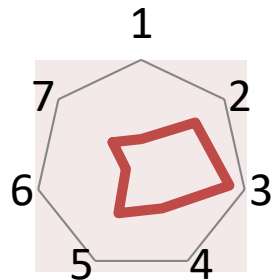
Facebook



Amazon



Google





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](http://www.jiuzhang.com)