

模拟算法和字符串处理技巧



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://zhuanlan.zhihu.com/jiuzhang>

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

官网: www.jiuzhang.com

- 高中信息学竞赛，湖北省队第一名，冬令营金牌，5年信息学竞赛教练
 - 本科保送浙大计院
 - 哈佛大学交流一年
 - 计算机转行金融（Quant）
-
- 业余爱好：音乐剧，浙大灵韵音乐剧社创始人，常春藤盟校春晚导演

- ZOOM 的用法
- 老师/助教回答提问方式
- 课程Ladder: <http://www.lintcode.com/zh-cn/ladder/14/>
- 九章QA: <http://www.jiuzhang.com/qa/>
- 错过第一节课不要紧，可以上下期课的第一节课（第一节课均免费）
- 答疑QQ群: 238430471

- 以下哪种情况可以直接使用数组作为hash (多选)
 - a. '1' b. 1 c. "123" d. 'A' e. "db"
- 拿到动态规划的题目，一般我们的第一个直觉是什么？
 - 考虑_____一步 （填空，两个字）

- 电面经典模拟题 (2题)
- 系统操作模拟题 (3题)
- 罗马数字转换题 (2题)
- 简单智力类问题 (1题)

电面经典模拟题

Sliding Window Average from Data Stream

<http://www.lintcode.com/zh-cn/problem/sliding-window-average-from-data-stream/>

<http://www.jiuzhang.com/solutions/sliding-window-average-from-data-stream/>

Sliding Window Average from Data Stream

Example:

- size=3
- $m.next(1) = 1$
- $m.next(10) = (1 + 10) / 2$
- $m.next(3) = (1 + 10 + 3) / 3$
- $m.next(5) = (10 + 3 + 5) / 3$

思路:

- 我们先来一个最简单的做法
 - 来一个数就存数组
 - for 循环最近size 个数求和取平均返回
- 时间复杂度是多少呢?
 - 每次 $O(\text{size})$
- 怎样优化算法——如何快速求和?
 - 前缀和数组

- 什么是前缀和数组？
 - 下表中a是原始数组
 - 定义s是a的前缀和数组， $s[i] = a[1] + a[2] + a[3] \dots + a[i]$

idx	0	1	2	3	4	5	6	7	8	9
a		3	4	11	1	-2	3	2	1	5
s	0	3	7	18	19	17	20	22	23	28

- 方便快速求a数组中某一段的和
 - $a[k] + a[k + 1] + \dots + a[j] = s[j] - s[k - 1]$ 时间复杂度 $O(1)$
- 怎样快求s数组？
 - $s[i] = s[i - 1] + a[i]$ 时间复杂度 $O(n)$

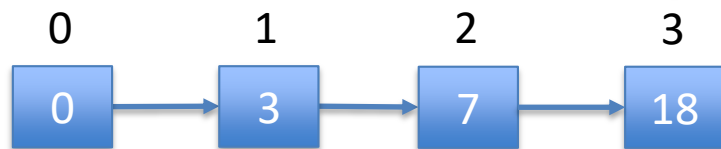
Sliding Window Average from Data Stream

- 如何节省储存空间呢？（2种方法）

1. 链表保存sum

2. 数组滚动

Sum链表:



插入第3个数 11
算出 $\text{sum}[3]=18$
最近3个数的和
 $\text{sum}[3]-\text{sum}[0]=18$

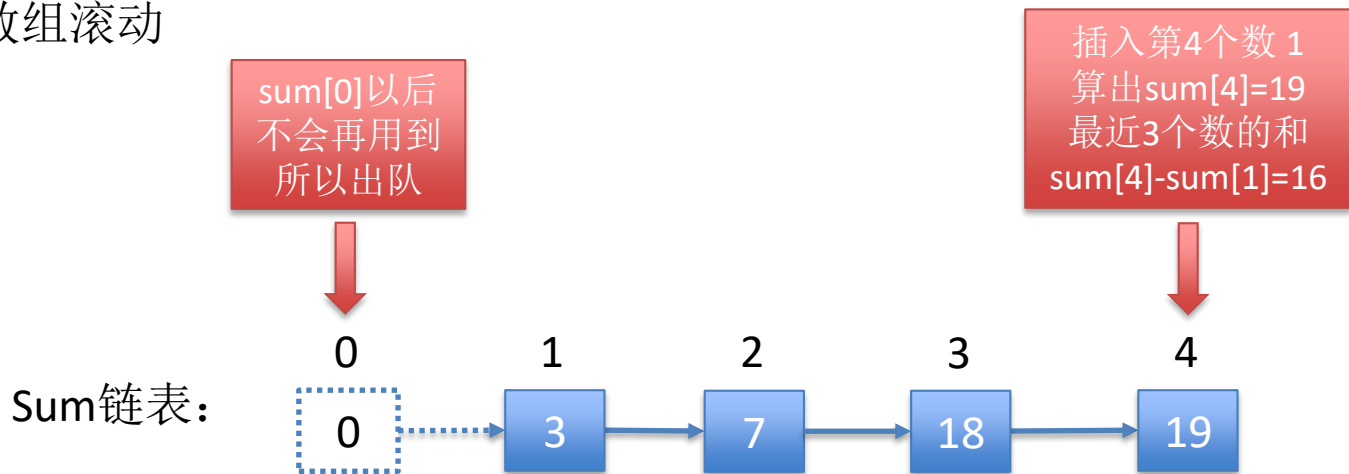


Sliding Window Average from Data Stream

- 如何节省储存空间呢？（2种方法）

1. 链表保存sum

2. 数组滚动



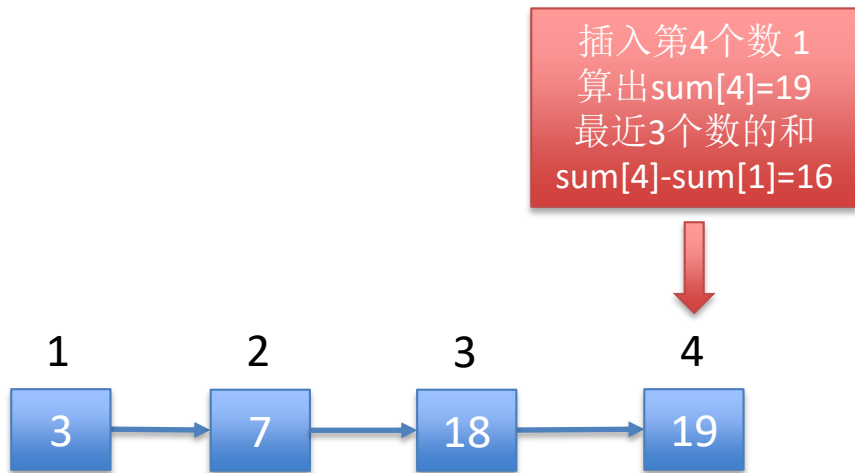
Sliding Window Average from Data Stream

- 如何节省储存空间呢？（2种方法）

1. 链表保存sum

2. 数组滚动

Sum链表:



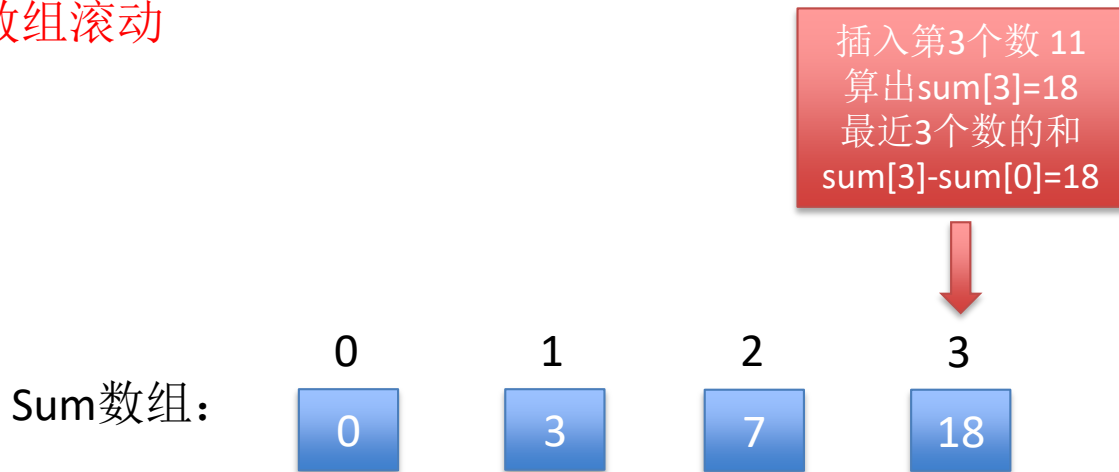
空间复杂度: $O(\text{size})$

Sliding Window Average from Data Stream

- 如何节省储存空间呢？（2种方法）

1. 链表保存sum

2. 数组滚动

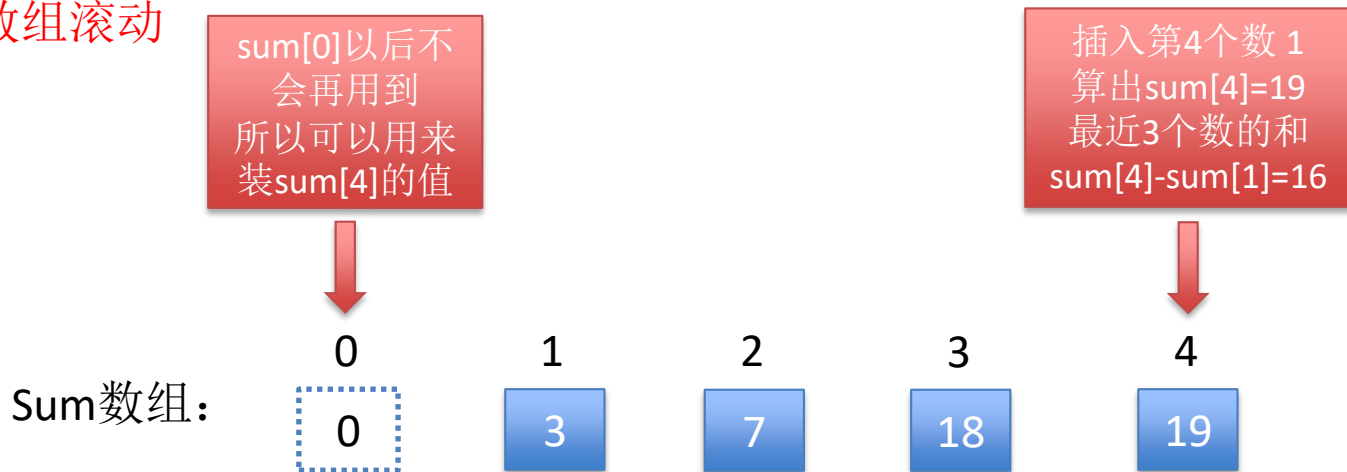


Sliding Window Average from Data Stream

- 如何节省储存空间呢？（2种方法）

1. 链表保存sum

2. 数组滚动

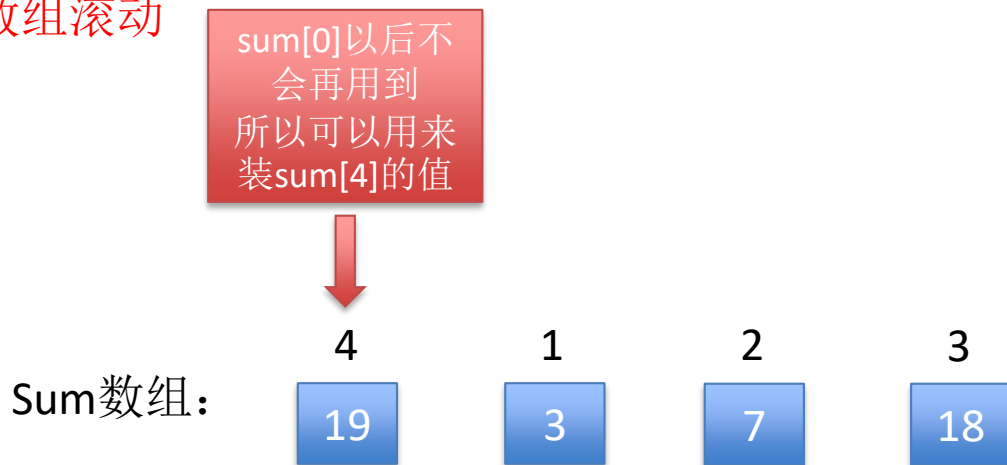


Sliding Window Average from Data Stream

- 如何节省储存空间呢？（2种方法）

1. 链表保存sum

2. 数组滚动

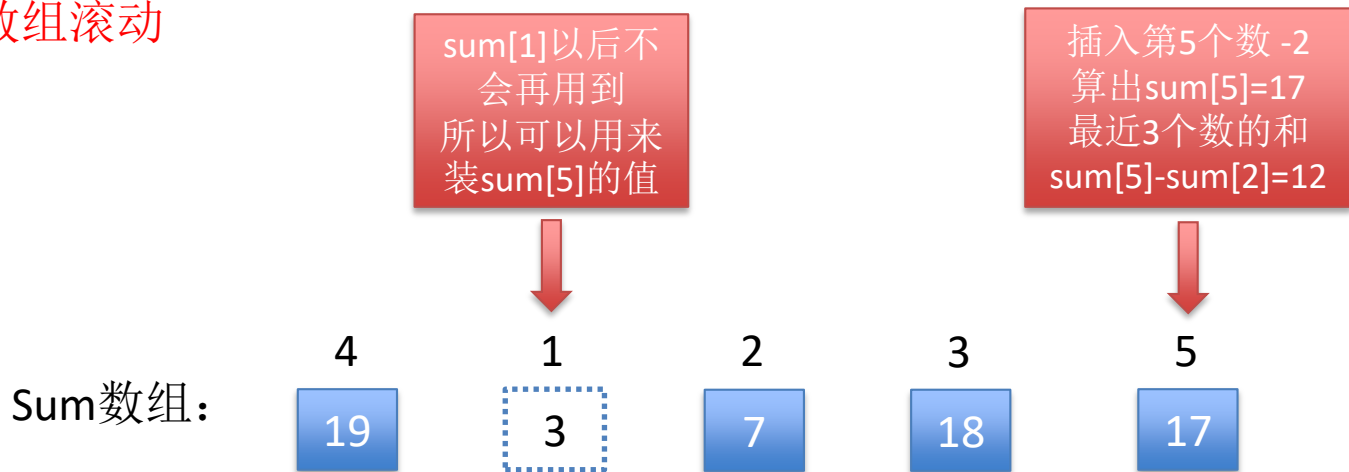


Sliding Window Average from Data Stream

- 如何节省储存空间呢？（2种方法）

1. 链表保存sum

2. 数组滚动

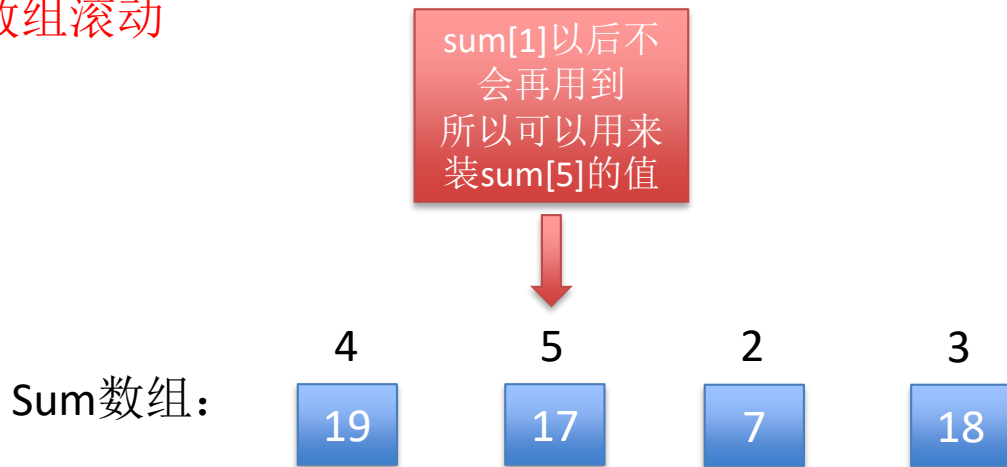


Sliding Window Average from Data Stream

- 如何节省储存空间呢？（2种方法）

1. 链表保存sum

2. 数组滚动

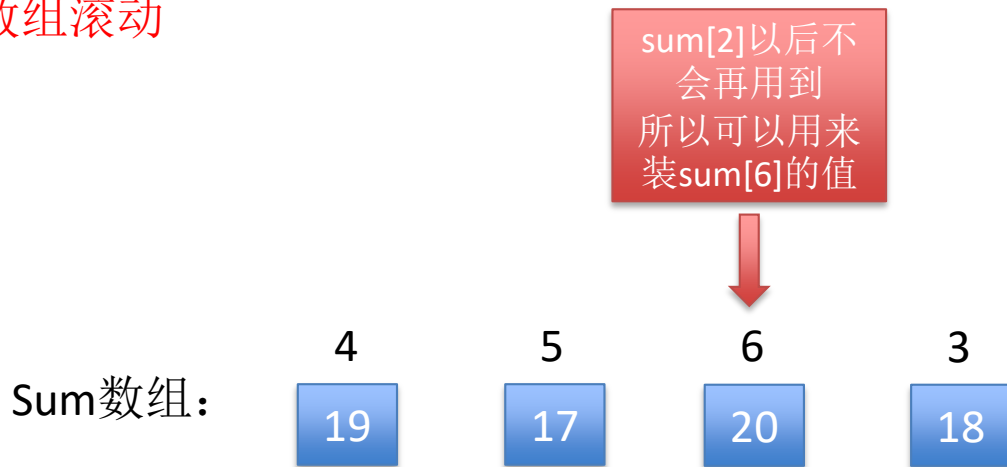


Sliding Window Average from Data Stream

- 如何节省储存空间呢？（2种方法）

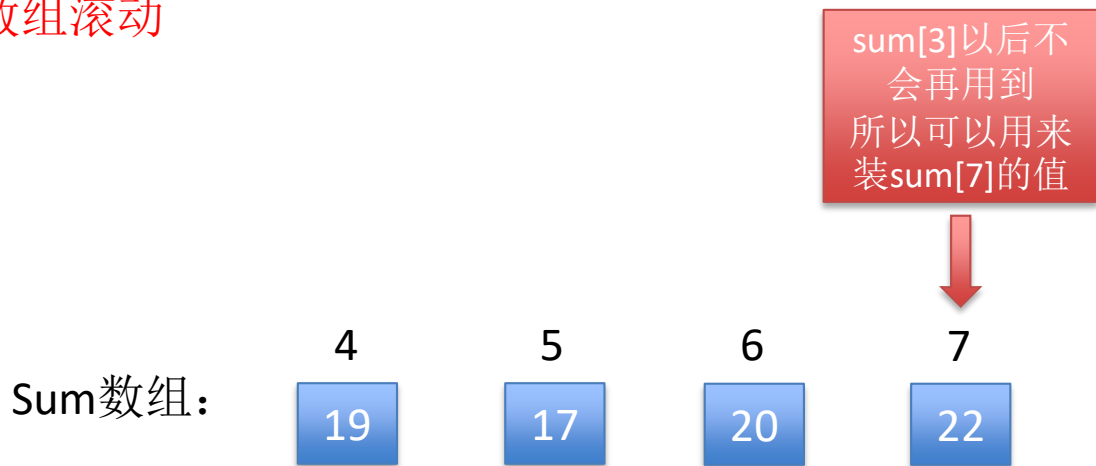
1. 链表保存sum

2. 数组滚动



Sliding Window Average from Data Stream

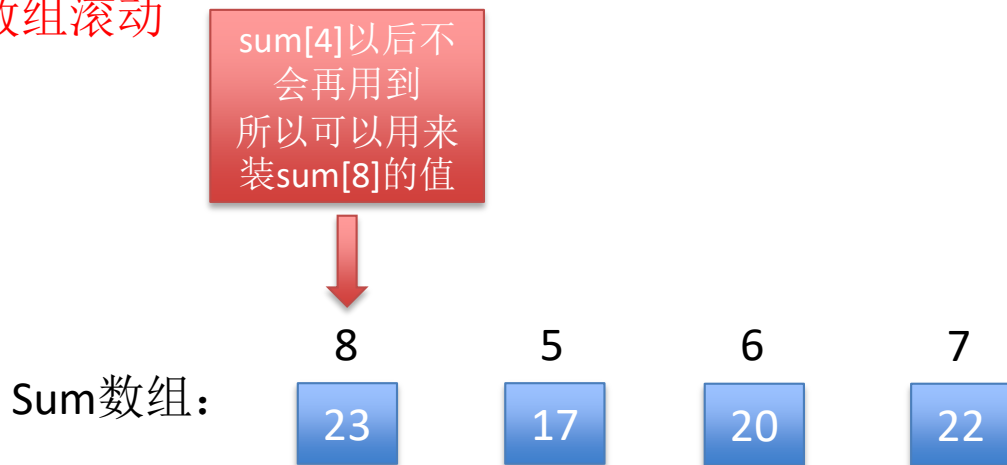
- 如何节省储存空间呢？（2种方法）
 1. 链表保存sum
 2. 数组滚动



- 如何节省储存空间呢？（2种方法）

1. 链表保存sum

2. 数组滚动

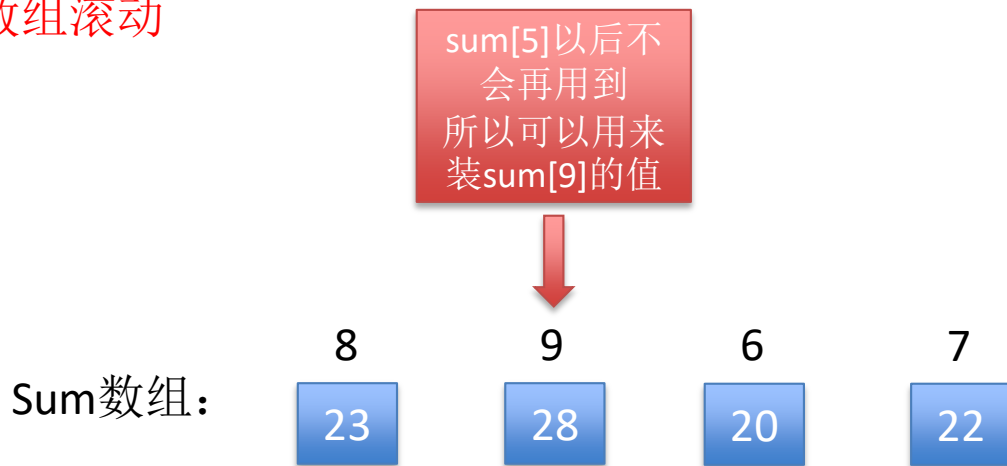


Sliding Window Average from Data Stream

- 如何节省储存空间呢？（2种方法）

1. 链表保存sum

2. 数组滚动



Sliding Window Average from Data Stream

- 如何节省储存空间呢？（2种方法）

1. 链表保存sum

2. 数组滚动

逻辑位置	0	1	2	3	4	5	6	7	8	9
实际位置	0	1	2	3	0	1	2	3	0	1

- 怎样得到实际位置？
 - 逻辑位置取mod，这一题mod (size+1)

空间复杂度：O(size)

- Company Tags: Google

考点:

- 能否想到前缀和优化
- 能否进一步想到空间优化
- 15分钟bug free 写出来
- 电面好题，筛选基本代码能力的面试者

能力维度:

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

◆ 小技巧总结:

- 如何快速求和? 前缀和数组(dummy 0)
- 如何节省储存空间呢? 链表/滚动
- 写滚动的技巧 先写程序最后加滚动

- 前缀和数组应用:
 - <http://www.lintcode.com/zh-cn/problem/interval-sum/>

- 滚动的应用:
 - DP 背包问题 **backpack** 系列
 - BFS 中的循环队列

Edit Distance II

<http://www.lintcode.com/zh-cn/problem/edit-distance-ii/>

<http://www.jiuzhang.com/solutions/edit-distance-ii/>

Example:

- s="abcd" t="abce" output: true
- s="abcd" t="abcef" output: false
- 两种one edit:
 - 1.增加/减少一位
 - 2.更改一位

思路:

- 想一想特殊情况 柿子要找软的捏
- 两字符串长度相差太大时? 两个字符串长度一样?
- 三种情况:
 1. 两个字符串长度差 >1 直接false
 2. 两个字符串长度差 $=0$ 只能有一位不同
 3. 两个字符串长度差 $=1$ 只能插入/删除一个字符
- 边界情况, 特殊输入? 时间复杂度: $O(n)$

- Company Tags: Facebook

考点:

- 很懵逼的情况下怎样突破 -> 找特殊情况
- 分类讨论问题的情况

能力维度：

1. 理解问题
4. 逻辑思维/算法优化能力
5. 细节处理（**corner case**）

- 题意简单，面试官不用过多解释
- 题目难度不高，有基础的面试者可以在短时间内bug free实现
 - 要求：练到10-15分钟内 bug free写出来

系统操作模拟题

Read Characters From File - multiple calls

<http://www.lintcode.com/zh-cn/problem/read-characters-from-file-multiple-calls/>

<http://www.jiuzhang.com/solutions/read-characters-from-file-multiple-calls/>

Read Characters From File - multiple calls

Example : 1 2 3 4 5 6 7 8 9

- 每次读的个数 6 1 4
- 第一次: 1 2 3 4 5 6
- 第二次: 7
- 第三次: 8 9

难点:

- 如果只读3个, 剩下的一个怎么处理? (读多的怎么留着给下次用?)
- Read4 这个函数只读了2个怎么办? (读到末尾时, 没有读全4个)

Read Characters From File - multiple calls

思路:

- 计算机中硬盘是怎么读数据的?



- 本题类似内存从硬盘读数据
 - 输入字符数组: 硬盘
 - `read4`: 读硬盘一整块到缓冲区
 - `read(n)`: 内存实际需求
 - 需要我们做的: 实现一个缓冲区及相应功能

思路:

- 缓冲区用什么数据结构?
 - 队列（**FIFO**），因为要保持数据顺序不变
- 缓冲区队列的逻辑?
 - 队列为空时就进队（**read4**）
 - 队列不为空时就满足内存的请求，也就是出队

Read Characters From File - multiple calls

思路:

- Example : 1 2 3 4 5 6 7 8 9
- 每次读的个数: 6 1 4

请求	result
6	



队列为空

Read Characters From File - multiple calls

思路:

- Example : 1 2 3 4 5 6 7 8 9
- 每次读的个数: 6 1 4

请求	result
6	



Read Characters From File - multiple calls

思路:

- Example : 1 2 3 4 5 6 7 8 9
- 每次读的个数: 6 1 4

请求	result
6	1 2 3 4



队列非空: 出队

Read Characters From File - multiple calls

思路:

- Example : 1 2 3 4 5 6 7 8 9
- 每次读的个数: 6 1 4

请求	result
6	1 2 3 4

5	6	7	8
---	---	---	---

队列为空: 进队

Read Characters From File - multiple calls

思路:

- Example : 1 2 3 4 5 6 7 8 9
- 每次读的个数: 6 1 4

请求	result
6	1 2 3 4 5 6

		7	8
--	--	---	---

队列非空: 出队

Read Characters From File - multiple calls

思路:

- Example : 1 2 3 4 5 6 7 8 9
- 每次读的个数: 6 1 4

请求	result
6	1 2 3 4 5 6
1	



队列非空

Read Characters From File - multiple calls

思路:

- Example : 1 2 3 4 5 6 7 8 9
- 每次读的个数: 6 1 4



请求	result
6	1 2 3 4 5 6
1	7



Read Characters From File - multiple calls

思路:

- Example : 1 2 3 4 5 6 7 8 9
- 每次读的个数: 6 1 4

请求	result
6	1 2 3 4 5 6
1	7
4	



队列非空

Read Characters From File - multiple calls

思路:

- Example : 1 2 3 4 5 6 7 8 9
- 每次读的个数: 6 1 4

请求	result
6	1 2 3 4 5 6
1	7
4	8



Read Characters From File - multiple calls

思路:

- Example : 1 2 3 4 5 6 7 8 9
- 每次读的个数: 6 1 4

请求	result
6	1 2 3 4 5 6
1	7
4	8



队列为空: 进队

Read Characters From File - multiple calls

思路:

- Example : 1 2 3 4 5 6 7 8 9
- 每次读的个数: 6 1 4

请求	result
6	1 2 3 4 5 6
1	7
4	8 9



队列非空: 出队

Read Characters From File - multiple calls

思路:

- Example : 1 2 3 4 5 6 7 8 9
- 每次读的个数: 6 1 4

请求	result
6	1 2 3 4 5 6
1	7
4	8 9



队列为空: 进队

全部读完: 程序
退出

时间复杂度: $O(n)$

- Company Tags: Google Facebook

考点:

- 对队列这种数据结构的理解
- 对细节的处理，大家都会但不容易写对
- Onsite
- 25min

能力维度:

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

Strings Serialization

<http://www.lintcode.com/zh-cn/problem/strings-serialization/>

<http://www.jiuzhang.com/solutions/strings-serialization/>

思路:

- 简单的想法, 用'; ' (或者 '+' 等) 将字符串连起来
- 如果字符串中本身就有 ';' 呢? ';' 是连接符还是原有字符?
- 考虑\n \\ 这一类转义字符的原理
- 用 ':' 表示转义, 那么连接符就是': ; ' 表示 ':' 本身就是 ': : '

- abc def -> abc::def::
- ab:c def -> ab::c::def::
- ab::c def -> ab:::c::def::

- Company Tags: Google
- 类似题目:
 - <http://www.lintcode.com/zh-cn/problem/binary-tree-serialization/>

能力维度:

4. 逻辑思维/算法优化能力

休息5分钟

System Longest File Path

<http://www.lintcode.com/zh-cn/problem/system-longest-file-path/>

<http://www.jiuzhang.com/solutions/system-longest-file-path/>

思路:

- 有点小麻烦的纯模拟
- 技巧一：用`split('\n')` 将原串分割开，相当于一次读一行
- 技巧二：利用`'\t'`的个数来当前目录/文件 在第几层
- 技巧三：从上到下一行一行顺序读入，用类似栈操作，把前面几层的字符串长度都记下来

System Longest File Path

	Level	Length
dir	0	3
subdir1	1	7
file1.ext	2	9
subsubdir1	2	10
sdir2	1	5
ssdir2	2	6
f2.ext	3	6

Level	0	1	2	3	4
Length	3				

System Longest File Path

	Level	Length
dir	0	3
subdir1	1	7
file1.ext	2	9
subsubdir1	2	10
sdir2	1	5
ssdir2	2	6
f2.ext	3	6



Level	0	1	2	3	4
Length	3	7			




System Longest File Path

	Level	Length
dir	0	3
subdir1	1	7
file1.ext	2	9
subsubdir1	2	10
sdir2	1	5
ssdir2	2	6
f2.ext	3	6


	求和				
Level	0	1	2	3	4
Length	3	7	9		

System Longest File Path

	Level	Length
dir	0	3
subdir1	1	7
file1.ext	2	9
subsubdir1	2	10
sdir2	1	5
ssdir2	2	6
f2.ext	3	6



Level	0	1	2	3	4
Length	3	7	10		



System Longest File Path

	Level	Length
dir	0	3
subdir1	1	7
file1.ext	2	9
subsubdir1	2	10
sdir2	1	5
ssdir2	2	6
f2.ext	3	6



Level	0	1	2	3	4
Length	3	5	10		



System Longest File Path

	Level	Length
dir	0	3
subdir1	1	7
file1.ext	2	9
subsubdir1	2	10
sdir2	1	5
ssdir2	2	6
f2.ext	3	6



Level	0	1	2	3	4
Length	3	5	10		

无效值

System Longest File Path

	Level	Length
dir	0	3
subdir1	1	7
file1.ext	2	9
subsubdir1	2	10
sdir2	1	5
ssdir2	2	6
f2.ext	3	6



Level	0	1	2	3	4
Length	3	5	6		




System Longest File Path

	Level	Length
dir	0	3
subdir1	1	7
file1.ext	2	9
subsubdir1	2	10
sdir2	1	5
ssdir2	2	6
f2.ext	3	6



	求和				
Level	0	1	2	3	4
Length	3	5	6	6	



System Longest File Path

	Level	Length
dir	0	3
subdir1	1	7
file1.ext	2	9
subsubdir1	2	10
sdir2	1	5
ssdir2	2	6
f2.ext	3	6



Level	0	1	2	3	4
Length	3	5	6	6	
sum	3	8	14	20	



- Company Tags: Google

考点:

- 是否可以形象化的思考这个问题
- 是否可以熟练的处理字符串

能力维度：

1. 理解问题
2. 代码基础功力
5. 细节处理（corner case）
7. debug能力

- 面试题目很多都从平时工作中出来的

罗马数字转换题两道

Roman to Integer

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

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

- Example: CDXXI=421
- 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"}.

- 大家在做这题时遇到了什么问题？
 - 怎样区分IV 和VI？

思路：

- 从左往右加起来，比如XVII=10+5+1+1=17
- 那么像IV=4 IX=9 XL=40 XC=90 这样的怎么处理呢？
- 没有4 9 40 90 这种的，字母代表的数字从左往右是从大到小的
- 发现左边的如果小于右边的，就把左边的减去，比如CDXXI

- Company Tags: Facebook

能力维度:

1. 理解问题
2. 代码基础功力

Integer to Roman

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

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

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

简单智力类问题

Identify Celebrity

<http://www.lintcode.com/zh-cn/problem/identify-celebrity/>

<http://www.jiuzhang.com/solutions/identify-celebrity/>

- 名人就是所有人都认识他，他只认识他自己的那个（名人检验）

思路一：

- 一个简单的做法，对每个做一次名人检验，看是不是所有人都认识他，但他不认识所有人
- 时间复杂度 $O(n^2)$
- 怎样降低时间复杂度？ 冗余在哪里？

思路二：

- 我们询问一次的时候只利用答案为true的情况，如果为false呢？
- 一次询问knows(a, b): true a认识 b a一定不是名人
- false a不认识b b一定不是名人
- 所以一次询问就可以排除一个人，n-1询问后剩下一个人，再对这个做个名人检验就能确定他是否为名人
- 所以实现上就是从左到右扫一遍，每次都是保留下的人和新人做一次询问，最开始保留的人设为第1个人

- Company Tags: LinkedIn Facebook

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）

◆ 小技巧总结:

- 降时间复杂度 -> 找冗余
- 思维上双向: **true**时候, **false**的时候?

◆ Sliding Window Average from Data Stream

◆ 小技巧总结:

- 如何快速求和? 前缀和数组
- 如何节省储存空间呢? 链表/滚动
- 写滚动的技巧 先写程序最后加滚动

◆ Edit Distance II

◆ Read Characters From File - multiple calls

◆ Strings Serialization

◆ System Longest File Path

◆ Roman to Integer

◆ Integer to Roman

◆ 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? $\% k$ $/ k$

◆ Identify Celebrity

◆ 小技巧总结:

- 降时间复杂度 - > 找冗余
- 思维上双向: **true**时候, **false**的时候?



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