

Phone:

- LC1011
- Merge two sorted lists. 时间复杂度, 空间复杂度。怎么优化。。。
- 2. 一些java方面的基本问题, multi-threading, class vs interface等。
- 第一题是一个cache。说得很玄乎, 然而实现起来并不难。follow up问了如果rank经常变怎么办。 <https://bre.is/kBbkxdaM>
然后开始coding, 写一个cache, 要求给定一定size, 如果cache size不够, 就把最小的挤出去。
- 用pq做的,
- `the sum of the contiguous subsequence with maximum sum.`
- LC20
- LC671
- 第一题 给一个number 查这个数字是不是个perfect square. return boolean. ex: 4 = true, 9 = true, 8 = false
- 写好后问multithread的问题
- process 和 thread 的区别。
- 244
- 339
- 364
- 一个整形矩阵, 行列都是排好序的。用最快的时间找出所有数字的中值。LC378
- LC170
- LC88
- LC380
- LC68
- Java Garbage Collector 是怎么work的
- Longest Palindromic Substring 的变形, 求最长的palindrome的长度
- design blocking queue, 实现enqueue 和 dequeue, 实在没练过多线程的题, 这题回答的很惨
- 2. Factor Combinations, 用的dfs 解决的
- 1. insertion sort
- 2. binary tree print by level order (follow up, print full tree)
- 2. 3个基本知识问题: what's thread vs process, what's write back cache vs write through cache, what's stack vs heap.
- 1. given a binary search tree and a value, find k numbers in the bst that are closest to the given value.
- LC56
- LC38
- process vs thread, virtual memory, paging, paging fault (这两个感觉不太好解释, 没解释清楚地样子), final vs finally vs finalized, inter-process communication
- 1. different between process and thread
- 2. Java - different between final, finally, finalized
- 第一道是找树的最大长度。
- max stack LC716
要求两种方法做, 第一种两个stack口述了一遍
然后要求优化时间复杂度, 拿treemap做
- rotate linkedlist k times LC61

- heap vs stack process vs thread
- LC339
- LC364
- LC156
- 另外给一段程序问C++运算符的overloading相关,
-

什么是transaction 解释ACID特征

什么是java中的exception 比较throw exception和return error code的优劣

什么是paging和page fault

算法是实现HashMap follow-up如果是multi-threaded该怎么处理

- LC150 还要写出给的两个exception : FormatNotValidException, ArithmeticException
- LC339
- 第一轮 刷题网 而物流 paint house
- 然后问了email spam filter design
- 1.第一题很简单, 就是看BST, 二分查找树是不是valid。
- 第一题一个board game, 给出限制, 问输入是不是合法的。基本无难度, 感觉代码写的快就行。
- 第二题类似LRU Cache但是Eviction条件不同, Put的时候自带权重, 要求Evict权重最低的。HashMap+PriorityQueue搞定。然后分析Multi-thread的情况, 分析哪里可能是瓶颈, 提出可以用ConcurrentHashMap + Lock on Queue。
- 第一题蠡口刘酒, 第二题蠡口散散, 两题的返回类型都是boolean
两道题都口头过了一遍给出的test case,第二题两个常规followups:1. 函数会被多次调用, 如何优化避免读input list多次 2.input list存的数memory装不下怎么办
- 然后出了一道lc原题, 就是给一群朋友的follow关系 map, 比如:map[0][1] = true, 就说明0在follow 1, 如果map[56][12]= true, 就说明56在follow 12
-
- 然后找出一个leader, leader的条件就是他没有follow 任何人, 但是其他所有人都follow他。
-
- 然后做的太快, 那人就随便找了些follow up问了问。
- 比如没有leader, 这个好办
- 比如leader有多个, 我这时候就想到这种情况的话要么有k个leader, 要么0个.. 栗子: 如果1,2,3个人, 2, 3没有follow 任何人, 所以要么他们是leader, 要么没有leader, 因为如果3不是, 但是他又没有follow 2, 那么2也不是. 讨论了下, 然后那人说leader之间不需要follow
- 先问了process和thread的区别, 问的很细, 就光聊这个聊了有半个多小时
- LC43
- LC380
- 给string1,string2,问string2是不是由string1移动构成的, 应该是leetcode原题 Lc87
- 给你一个array,看里面的数能构成一个三角形吗, 返回t/f
- 1. 热身 : pow(a, b)
- 2. Rankable cache.
- 虚拟内存, 进程和线程, 数据库事务和ACID, java的checked/unchecked exception, final/finally/finalize的不同
- LC243, followup是244。
- 2. 四留, followup是多线程怎么处理同时避免重复。
- 散流撕, 四流, 四期
- NestedClass(static和non-static)怎么instantiate,什么是paging和page fault

- 前5分钟自我介绍，然后问了什么是Write-back和Write-through Cache，follow up是哪一个比较快。
- 接下来是coding，[刷题网380](#)的小变形，getRandom改成了removeRandom。小哥几乎全程沉默，无论是解释算法和设计还是边写边解释代码都没有什么回应。
- 写完之后发现才过了不到半小时，小哥说时间多人工尝试几个test case，我觉得如果不是他们给的coderpad关掉了运行功能的话肯定会要求我现场跑通。
- 走完test case之后时间也还没到40分钟，于是小哥follow up了一个伞霸妖，不仅要解释方法也要完整写出来。好在是高频题，没花太多时间写完了。
- 上来就问process, thread的区别，要求用stack, queue解释。面试进一步问malloc() function的原理，又问了thread之间怎么communicate...lz没有准备太多系统底层的東西，于是答得不好。希望有人能给解释一下或者留个link? (回去翻了下system的书也还是没搞明白)。
- coding是给一个data source (很大，query慢)，给一个rank() return rankable item的score 要求设计一个bestScoreCache。大致是这样，不是很难。但是follow up问到了读cache时 racing condition的问题于是又继续follow up system里lock啊之类的操作，问的很细。。。lz汗如雨下。
- "find lowest common ancestor with parent pointer"
我的解法 如先前討論過的 非常簡單 就先把其中一個 node靠parent指針移動到root 然後所有的node都存在set裡面 接下來另一個node也往root的方向移動 只要node在set裡面 就return 這個node
但是老印就堅持下面的方法比較好
<https://gist.github.com/skxie/7283262>
但我覺得這個方法雖然省了空間複雜度 但是時間複雜度更高不是嗎??
- 1. 给一个output是0或1的随机数生成器，0的概率是p，1的概率是1-p
求写一个生成0，1概率是50/50的函数
- 2. 给一个范围，比如2-19，写一个函数，要求返回的随机数在范围里并且每个数概率一样
- 第一轮：利口原题，落叶子，给了dfs解。serialize/deserialize 树，说了bfs解，说这样需要添加null节点，不好，问怎么优化，脑子僵住了没想到，要了提示，说traversal。于是给了inorder/preorder遍历解法，利口原题。
- 1. 利口215。
- 2. 蠡口244。
- 基础知识：thread和process的区别，heap和stack，比较TCP和UDP
- 最后算法题，单词最短距离，给两个单词，和一个单词list，找出这两个单词的最短距离，当这两个单词相同的时候怎么做
- LC原题 flowerbed 1042
- best rank cache，类似于LRU
- 346 ES5 vs ES6 以及 Sql vs no sql
- 一道是括号匹配题
- 第二道是求两个词的距离的题
- 1. Given a sorted array of unique elements R, which are letters of the English alphabet, and an input character x. The elements in R are sorted with the least element appearing first. Find the minimum r in R such that r > x. If there is no r > x, find the first element of the array (wrap around).
- e.g. R = ['c', 'f', 'j', 'p', 'v']
if x equals:
'a' => return 'c'
'c' => return 'f'
'z' => return 'c' (wrap around case)

- 2. 李冠 三伊酒
- 1. 给定二叉树，判断这棵二叉树是否对称 LC101
- 2. 还是二叉树，要么2个子节点要么没有。父节点的值不大于子节点。返回节点值构成的集合中的第k小的值。LC671
- 3. 给出一个字符串，找出第一个只出现一次的字符
- 第一题，subset，分析时空复杂度
- 第二题，一颗全是正整数的二叉树，求根节点到叶子节点的和最大的路径
- 2. 甩了我一个投硬币问题，问我一个fair的coin，规则是连续不停扔硬币，投N次中出现K个head的Maximum likelihood estimation(我当场就呆了，完全忘记了，没准备还给当年老师了)，他问我MLE大概是什么definition，还希望我写在白板写公式推导，当时我就觉得我要挂了，在这边我卡了15-20分钟，支支吾吾的在小哥不断提示下也还是不停卡壳，最后几乎是他帮我写出来的
- 3. coding也是跟数学相关，让写一个函数，入参是四个：一个sorted list of int X(有可能有负数)，一个int A，一个int B，一个int C，function内部就是算 $Y = A * X^2 + BX + C$ 返回sorted Y
- 当然最简单的办法是算好Y再sorted，小哥当然说要更快的办法，然后给我画了一个曲线，提醒我先求极值的线（就是这个曲线是在X=?的时候最低值，然后X从这个点递增or递减Y都递增），就是对 $Y = A * X^2 + BX + C$ 对X求导。。。让我可以考虑先把公式写出来巴拉巴拉。。。我最后也是卡了很久磕磕巴巴写出来了
- 一开始是一些常规BQ，然后是两道java foundation：虚拟内存是啥，描述一下Java内存模型中的stack和heap。coding是刷题网156，follow up是输入是错误的如何处理可能的edge case。
- 1.int hasIntegerSquareRoot(int n). 当时脑子当机竟然想不出logn的方法（binary search即可），用了linear。。（还问了一下edge case，input超大怎么办）。当然小哥也没让我一直想，想了literally 30sec，小哥说没事我们看下一题。。
- 2.nestedList Sum. reversed. 我以为reverse应该是follow up结果直接来reverse，没啥好说的高频面镜。先用array做了。然后口述了leetcode光头哥大神给的weighted += unweighted的答案小哥并没有很认，除了用array存，他还认两边pass. 第一遍先求depth。还问了不用stack咋办。我说queue啊或者不用queue的bfs。他说可以用递归吗我说可以啊 take 3 param as same as stack。。
- 第一轮是一个swe，hr没有给meeting code，导致interviewer耽误了15分钟。之后又聊了15分钟之前的项目。开始算法题之前简单问了一下基本系统知识，memory，storage之类的。题目是很常见的计算一个operation list的结果，[2, 3, +] = 5这种，讨论了一下各种语言handle error不一样的地方，还有什么情况可能overflow，因为时间限制，加了一个如何handle unary operation。
- micro services好处，怎么migrate到micro services，API版本控制等等。
- Implement a hash table. 需要跟面试官clarify并且讨论怎么处理collision，然后写code。这题讲真不容易一次写对。。。我还忘记了re-hash。不知道结果如何。
- Q2:
- Print a tree and fill it to be complete tree, for example
- 10
- 1,2
- -, -, 3
- -, -, -, -, 6
- 387, 739
- 1. 244
- 2. 369

- 第一题：merge两个排好序的array。follow up：如果是merge一千个sorted array怎么办？分析时间复杂度
- 第二题：给你一个log file，从中提取出所有的ip地址。log file里内容五花八门，需要从中提取出所有valid的ip地址。follow up主要考察怎么handle corner case，比如下面这些不是valid的：12.03.04.56, 12.34.56.78.90, 12.345.67.89
- 第三题：给你一个int array，返回所有只出现过一次的数字。follow up：返回数字的顺序应该跟数字在input array里出现的顺序相同
- 第一道题 最近距离 没啥好说的 写完了就完了 bug free
- 两道蠢口1. 1612. 散酒. 都是原题
- 1. 244，考虑可能字符串不在初始化系列里，考虑相等。
- 2.341 follow up 加一个 remove 操作
<https://steventang626.com/2019/01/18/NestedIterator/>
- . TCP UDP 和 use case
- LC605 flowerbed
- LC671 second min number in tree
- Flip locker for n students, each move for i steps. 一个一位数组表示locker开还是关，最开始全部关闭，然后模拟打开、关闭
- 215. Kth Largest Element in an Array
- 339. Nested List Weight Sum
- 364. Nested List Weight Sum II follow up 加了一个weight list，这个是LC没有的
- LeetCode 53. Maximum Subarray
- LeetCode 152. Maximum Product Subarray
- LC50 Pow
- merge two sorted list，任何list都行，array都可以
- LC173 BSTIterator
- 讲完之后他问如果突然有很多人开始用你的service，你会怎么scale（一秒变[system design](#)）。我就给他大概说了一下怎么scale web-service 和service server 以及怎么sharding database. 没有非常细，因为我心里一直觉得这只是电面。终于20多分钟过去了，开始让我coding了。他说他准备了两道题，还好都是简单题，大概不到五分钟一道秒了。第一题是判断两个hashmap, 其中一个是不是另外一个的subset. 第二题，遍历一个set 不能用for loop。第二题我用的while loop + python里面set的pop方法。面试官并不知道set有这个方法，我就给他解释了这个方法是怎么work的。并且google 给他看了一下确实有。做完后大概30多分钟，我感觉应该到提问环节了，可是并没有。这时候面试官给了我一道sql题，问我如果有很多users 和user groups需要怎么给他们建立table，我告诉他这是一个典型的many-to-many的relationship。需要三个table，一个user table 一个group table, 一个(user, group) table去连接他们。面试官表示他懂了，不需要我写代码了。但一切并没完，面试官意犹未尽的问了我http，让我说一下我对http的理解。这个我最近刚看过，关于request response 里面的fields是干什么的基本都还记得。所以我并不觉得会有什么問題。
- 2. 利口 一丝留
- 输出数组中不重复数字[1,2,3,3,2,5,4] => [1,5,4]
- 2. Merge k sorted arrays
- 1. 亿霖散
- 2. 类似雾气变种
- 脸：
- oculus tech lead:
- 2. 溢流易
- 1. 问我java里面Arrays.sort 是怎么实现的？我说是quicksort. 他说好，你把quicksort写出来。

- 2. 利特口的 把酒凌。附加信息是每个string都是小写字母，所以我就用 int[26]了。
- System Design <https://bre.is/LDB9wF9A>
- linux log parser输入是linux dmesg output
- 两步，第一步看那一分钟内log 最多 — 时间定长，每行只读到timestamp 然后往map 里面存下，记一个全局最大值，最后返回最大值的key 就好
- 第二步 要返回一个表，每分钟每个process 有多少条 log — regular expression + map 就好
- 1. 二叉树按层打印
- 2. 给一个字符串序列，找任意给定的两个字符串的最短距离。
- 一轮店面水果 利口原题二百四十三：单词之间的最短路径，加问如果api会被call很多遍怎么办
- Returns $\text{base}^{\text{exponent}}$, as the standard mathematical exponentiation function
- 1. 蠢口 柳舞 简单版不考虑e 因为看到之前的面经所以准备了不考虑e的版本 直接秒了
- 2. 蠢口 伞散酒 伞柳丝 都准备到了所以做得很快 reverse的版本我三种解都准备了 先跟他说2-pass的版本，要我写了。写完后问我怎么1-pass，我答牺牲空间优化时间，把每个level的integer存map里，一边记录depth。然后其实第三种方法是leetcode讨论里stephan大神的unweighted weighted那个，但是我还没来得及说，面试官就自己说了，说这个蛮难想的不过很巧妙。我马上把他狂吹一波。
- 利口244和364
- 求树的深度 很简单略过
- 381
- 432 All $O(1)$
- 题一：
- 给定一个 BST，将其转换为首尾相连的双向链表。
- * 自己定义一个Node的表示方法，和双向循环链表的表示方法
-
- 题二：
- 实现Pow函数， `double pow(double a, int b) {}`
- * 需要注意一下b为负数的情况
- 判断一个字符串是不是代表数字 不需要parse
- 利口 而丝丝
- 第一题 244 shortest word distance ii。我提了一种算法，hashmap，面试官提了一种，让我说了一下tradeoff。之后就是写代码。
- 代码里写了个typo，面试官最后帮改过来了。哭哭～
- 第二题 best retain cache。
- 1. 给个String，判断是不是个valid number。看着很简单，但edge case还挺多的。讨论交流后cover了一些基本的case，写完后面试官给了个test case，模拟run了遍没问题就进入下一题
- 2. 计算两个string的edit distance，Leetcode 72
- 后缀表达式
- 问题二，给了一个两数合的接口。一个方法是加数字，一个方法是测试数字。测试数字就是问给的数字能不能用已经加的两数合组成。写完，走了例子，可以。问能不能把地图换成集合，我说可以，因为大哥你说没有重复。
- 跟进一，三数合怎么做
- 跟进二，改一下接口，传进来一个k，k数合怎么处理，给了方案，表示可以，问了复杂度，尼玛，脑子抽水，测试方案的时候用的三数合，所以直接给报了一个平方，发现不对，在老哥要我问问题的时候改正了，我说早上连着两轮脑子有懵，老哥说好的，不要担心。
- 跟进一，二都不用写代码。跟进一，稍微写了一下伪代码。跟进二就聊怎么处理，复杂度

- 1. 一个数组里面都是正整数，问能否把里面的元素分成K个group使每个group的总和相同
- 2. 写一个数据结构支持 add(key), remove(key), removeRandomKey() 并且每个操作都是 O(1)
- 蠡口儿凌凌，稍微有变化不是找island是找山谷，花了点时间讨论如何define山谷。。
- Node 有parent指针，找两个Node lowest common ancestor;
- 实现一个interface 支持两个method addNumber(int from, int to) 会create一个Interval; getTotalLength: return current cover length 重复的只能算一次; 有点像range module.
- 277
- 第一题热身，输出一个数组中所有只出现过一次的数字。用字典统计每个数组次数，遍历字典即可。
- 第二题是刷题网站尔厮寺，答了用字典统计每个字母的所有位置，然后把要找的两个单词的所有位置遍历求最短距离，小哥问可不可以优化一下，答了用双指针，然后让写代码，问用字典存储有什么benefit，答了如果很多query速度会快。问了一下时间空间复杂度，然后结束。中间有几次蓝牙耳机disconnected，于是敲完代码解释的时候直接手握手机交流，建议大家电面的时候尽量用有线耳机避免尴尬。
- 2. 写一个带权重的cache，给一个从数据库拿数据的api，要求如果拿到的数据权重比cache里高，evict那个低的。之后考察如果多线程怎么加锁。我就说读写锁，结果他说要按key加锁。我告诉他按key加锁是错的，他说那我们就跳过这个把...
- <https://www.1point3acres.com/bbs/thread-106498-1-1.html>
- 贰斯吴，只是问短距离，请他CLARIFY其他条件后，是这个，然后FOLLOW UP问为什么这么做，好处和坏处，然后是耀其领，也是要CALRIFY条件，FOLLOWUP问为什么这么做
- 1. 利口斯鑒
- 2. 给一个integer数组，找出其中有没有三个elements可以组成三角形。返回任意一组就可以。
- 3. singleton, thread singleton
- 2. 基础知识：heap 和 stack 区别；thread 和 process 区别
- 3. 利特口的 散散就 + 那个cache题目
- <https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=516800&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D6%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311%26orderby%3Ddateline>
- follow up : 多线程怎么办

OnSite

- 第一题是利口 气溜
- 第二题是利口 气伞领
- 如果开发一个auto complete功能，api request应该发哪些数据给server；
- 1. technical communication, 描述过去一个项目
- 2. 蠡口刘久巴
- 3. 午饭。小印小哥特别积极，一直夸linkedin福利好
- 4. 白人senior manager，斯坦福毕业工作了十几年了，挂在这轮。他太擅长聊天，我聊着聊着太comfortable了。我这次找工作第一次onsite，还没有熟悉bq的套路。他问了我工作中最看重什么，我说最看重impact 和help others。多嘴说了一句，以后估计也当不了principal，觉得自己没这天赋，但是还是想干好自己的事情。这句话被quote了，lack of career ambition and growth mindset。
- 5. 岛屿最大面积变种，要求按大小返回岛屿坐标。
- 蠡口儿乞儿。最优解前序后序 stack

- 6. [system design](#)。tiny url and top k. 国人大哥放水
- 1. Design delayed task scheduler (似乎是他家经典[system design](#)面经)
- 2. Design short url system
- 3. Coding
- (1) Number of islands:
 - 2d matrix filled with 0 or 1. How many islands?
 - follow up: Return their position index sorted by their area.
- (2) Max points on a line
 - given many 2d points, draw a straight line, return max points on the line. (math problem)
- 4. Coding
 - Replace all substrings in a string with a new substring
- 5. Lunch
- 6. Behavior with director
- 7. Design
 - (forgot) about Java multi thread and concurrency. 领英特色
- <https://bre.is/pSu9pkZZ>
- 1. hm行为面试
- 2. 介绍自己项目
- 3. [系统设计](#)top k问题
- 4. 二拜, 一二期, 一二楼
- 5. 忘记具体题目了应该和二拜差不多, 也是可以用bfs/dfs/union find做的题目。两个排序好的链表求并集/交集, follow up常数空间 (应该是, 有点忘记了), follow up大数据map reduce具体怎么做
- 第一轮讲自己一个最感兴趣的项目, 自己准备了个全程自己设计自己做的项目, 所以讲了大概50分钟, 问的很细, 从api 到database。
- 第二轮, coding, 给你一个string 里面包含一个括号【】 括号里面有逗号分开的数字, 你把这个数字分开返回, 这个主要考察的是找到所有的corner cases, 然后有个followup就是数字里面有‘-’表示的是range, 然后要把这个range的数字全部返回。
- 第三轮, 吃饭
- 第四轮, 设计一个系统能够monitor 一群host 和host上面的程序, 然后还有设计一个警报系统, (个人感觉应该好好研究一下一些monitor系统比如datadog)
- 第五轮, 经理问各种bq, 以前最喜欢的项目, 对组里最大的贡献, 为什么想来linedin, 以后想要做的方向, 为什么想换工作。如果你的project run late怎么办。
- 第六轮, dphard 给你一个数列, 和一个k 返回这个数列能不能分成k个数列, 每个数列的和相等, 利口原题六百9思巴。有一点不同是数列里面的子可以为负数。开始的时候尝试各种不同的方法到最后时间不够了也没弄出来, turnout这个题没有任何优化办法只能暴力解决, 试所有的combination。所以教训是做题如果找不到最优解, 一定不要花太多时间想, 先把worksolution写上 (即使是bruceforce) 不要等到没有时间了才写

-
- LC187
 - coding1 : 利口 尔临临 变种。
 - coding2 : 尔伞溜
 - coding3 : 伞气伞变种, 求和变求乘积。还有一道想不起来了。。。
 - 2. 二七二, 六九。其中六九先是返回int, 后来要返回double, 这里卡了一下, 经过提示才做出来, 以为这里挂了, 结果不是

- 3. 六八。这是一道hard题，我准备的挺好的，写起来就挺费时间的，花了很长时间跟面试官沟通，最后面试官也说你的算法没问题，但是hr反馈的时候就说这一轮表现不好，其他的也没有strong yes所以不能downgrade。。
- 第五轮：算法。考了邀思旧。
- 3.Coding：蠡口三思，蠡口二四四
- 4.Coding：蠡口壹佰，蠡口其二
- 找树叶，sliding window minimum, factor combination,
- 第一轮面了两道题，一道是非常常见的MaxStack，我用的是BST和bi-directional linked list，另外一道跟GC相关，也是设计一个数据结构，要你实现能够
- 1) reserveMemory(Location l, size n)
- 2) write(Location l, byte[] b)
- 3) get(Location l)
- 4) free(Location l)
- 没有时间写具体code了就讲了一下思路
- 第二轮算法是BST, List<Integer> findKClosestElementInBST(int target, int k)
- 第三轮 力扣散吧领
- 第四轮 力扣斯期
- Coding 1: 美国小姐姐，打印一个很长的染色体序列重复出现长度为10的所有子序列，写了一个sliding window+hashset的解，小姐姐觉得可以把set拿去优化空间复杂度到O(1)，给了点提示：由于染色体只有四个字母所以我们可以建立字母到数字的一个mapping，然后用一个boolean的visited数组在对应数字下标赋值为true,记录已经出现的染色体序列。由于这里花了一些时间第二题就说了思路简单写了一下。第二题是实现addInterval(int start, int end)和getTotalCoverage(),蠡口merge interval的变种
- Coding 2: 台湾小哥+小印小哥，小印小哥出了一个判断两个linked list是否相交的高频题作为warm up，台湾小哥第二题出了一道打印所有palindrome subsequence的题目，我写了一个recursive的标准解，然后台湾小哥问能不能优化，打印所有subsequence的话貌似没有overlap subproblem不能加memo来解决，台湾小哥觉得可能有更好的解法，小印小哥觉得没有更好的解法了，于是接下去我和台湾小哥就开始brain storm各种可能性看能不能优化，虽然最后没有得出一个可行的优化方案，但台湾小哥表示我之前写的解法已经good enough了
- 李口 期役流
- 李口 斯山而
- round4：Coding 1 三道题：find closest value in BST, find k closest values in BST, find k smallest pairs
- round6：Coding 2 两道题：max stack, number of different palindromic subsequences
- 2. coding：利口气衣儿
- 3. coding：利口斯伞儿
- 1. 离口二流要，这轮主要是讨论很多edge case和error checking，有些input不一定是valid的。
- 最后考了一道lowest ancestor，普通二叉树和bst，只讨论了思路。
- 2. 第一题，给一个数组和一个target，问有多少个subset可以相加得到这个target，第二题利口要是久
- 第四个是behavior的结果对面一个sr manager直接变成了另外一个system design meeting最后一个问题卡住了就说multi thread 的process 如果其中一个thread failed 然后怎么保证data consistency and in order。我本来要用kafka的例子结果他让我在rdbms 里面怎么解决怎么implement 我就懵了一下子没想出来
- 问题是一个数组 求非相邻的数的最大和 例如
- [3,4,5,6,7] -> 3 + 5 + 7

- 给出 red OR blue AND NOT green 就要返回包含red和blue但是不包含green的list of file
- 3. Coding. Text Justification
- 5. Coding. 二叉树找到和是target的路径
- 第一题刷题网160，后续是带环。我说可以先找环然后把环去掉正常做，最后再接回来。然后不让改输入，用flag解出来了。
- 第二题刷题网256，后续是265。之后又详细讨论了265不同解法的时间和空间复杂度。之后问除了dp还能怎么解，我说用dfs + pruning，之后又说了下dfs，pruning和dp之间的关系。
- 给一组数找出结果等于target的子集的个数 看似简单的背包 但是并没有说只是非负数 跪在这轮
- 实现两个api 添加interval get没被cover的interval
- number of island + followup岛的颜色不同怎么办
- A. falling leaves, 感觉是多年前的老题 好像leetcode有
- B. 两指针是否相交 要求无额外空间使用，这破题目代码写的让人感到很无趣，四个快慢指针
- A. 给很多interval 判断总共cove多少位置
- B. 直方图接雨水的老题
- 第二轮：coding， 奇尔
- 中午吃饭
- 第四轮：behavior轮，现在的manager怎么样，怎么样才是一个好的manager，challenging project，最有成就感的事。
- 第五轮：coding， 1)伞溜刘， 2)而摆， followup，根据island大小排序
- return all palindrome subsequence.
- 1. 刷题网 起要溜
- 2. 刷题网 伞溜溜
- 两轮coding，散溜溜、而流无，其要留讨论了两种解法
- LC272
- 利口156和利口46
- LC336, 256
- 判断两颗二叉树是否相同。热身之后，进入正题，蠡口么寺酒。
- 第一题：蠡口贰零零；第二题：蠡口伞把寺；第三题：蠡口齐么留
- 5. 利口柳散柳变种
- 6. 利口柳久变种，参数和结果是double
-

System Design

- **tinyURL**
 - 感觉由于这题已经被问烂了所以具体实现问得比较多，从选择几位char作为tinyURL开始就开始刨根问底，然后画模块图这些面试官都在走流程然后具体化到数据流是如何走的，在回答noSQL的table design上忘了从key value store出发然后就被面试官抓住问了很久。对于L家的system design由于题目大概率就是那几题，大家一定准备过程中尽量细化，不能只从high level考虑设计。
 - design考的是shortern url，然后跟面试官说看过了这道题，面试官就没有太纠结在为什么这样design，而是问了很多细节的问题，比如bottleneck在哪里。为什么选用key-value的database。key-value的database的read path和write path长什么样。为什么ring-architecture能够实现high availability和high scalability

答完了还有时间就又问了一些multi data center的问题。还有如果在做shorten URL的同时又想有做一些aggregation，比如globally的top 1000 hit url，要怎么做。最后reach agreement的一个solution就是有一个region-local的db做这个region的统计，然后隔一段时间take a snapshot然后有个service用这些snapshots统计出global的statistics

- 设计tinyurl 问使用sequential id有可能被猜出来mapping会crowd数据库 没太理解问题的意思 随便答了一个加一层hash来避免猜出来mapping 在双master和master slave的优缺点上问了半天 感觉就是烙印吃饱了撑的刁难人
- 而且对于每个url被访问了多少次，能够输出过去24小时之内的总访问量以及给定一个时间范围的总访问量；先画设计图讲了大概的思路，然后面试官会针对设计的各种部分提问，比如pre-generator怎么保证新的url不和已经被用过的重复，url expire的政策怎么设计，统计访问量在有多个server的时候怎么实现，需不需要cache，如果某个server down了怎么保证访问量的数据没有丢失等等
- 主体部分常规设计：NoSQL + hashing + offline key generation server + caching. 说了db design，算了QPS数据，db sharding
- 有趣的是最后有个followup，如果要收集telemetry data，analyse client request/geo location这些东西，我说可以用data streaming queue - kafka (跪舔linkedin家的当家产品)
- click stats，就是统计每个short url被read多少次。反正把自己能说的都说了，也不知道感觉怎么样。
 - 1. 怎么生成短地址
 - 2. 怎么存储 - 主要是分析Sql和Nosql的优劣
 - 3. Redirect
 - 4. 如何统计被访问最多的地址
- design1 : time delay scheduler
 - 6. 延迟任务规划器，写了完整的代码，讨论了conditional variable原理，lock，何时会抛出interrupted exception，executor的用法，threading pool的好处
 - 其实这题跟concurrency关系不大，我读过面经，一直在往condition variable那边靠，像blocking queue那种设计，wait() + notify()
 - 但聊到最后，发现heap+hashmap (<timestamp, taskid>) 就可以了...还聊了一会heap的原理，heapify up/down, $O(\log(n))$ add/peak，聊偏了...
 - 参考java的写法。跟印度大哥半天解释不清楚，就直接把code写上去了，又写了一个client调用的code，印度大哥大姐很满意。感觉这轮更像算法轮。
- desing Calender
 - Google Calender
 - 设计一个日历系统
 - 这是整个面试的亮点，感觉面试官希望我快速介绍基本功能，设计中间层cover大多数功能，根据自己假设的accessing pattern设计storage, service, logging, monitor, analytic的实现，简直是大脑不能停的45分钟。
 - 前看了一些资料有点概念，答得还可以，自认为该cover的都cover了，就是怎么建表更快查询其他人schedule面试官提出异议，
 - 给了个方案，就是再建一个表，只存meeting id和用户id，然后说这样join少。我也不太懂，平时不摆弄sql和table这些，点头称是，讨论完这个时间到了
 -

- design2 : 给了一个API可以获得某个人的直接联系人, 实现几个API : 获得某人的1st 2nd 3rd degree联系人
- monitoring system.
 - 设计一个monitoring 系统, 需要实时监测System Level (cpu usage, network usage ...)和 app level (exception, error...) 的 matrix. 然后要可以发notification给owner
 - 关键字 hundreds of datacenters, hundreds of services in each datacenter, multiple type of events may occur in each service. 要求, collect all the events for each service, user can monitor it in real time, if some event exceed the threshold, corresponding engineers will get a notification. 典型的streaming processing的 architecture. message queue(kafka)+workers(storm)+database+cache+notification service(response queue). 需要注意的点就是 1 collect events 是用pull还是push, notification是用pull还是push 聊清楚trade off. 2最新的数据process完 写进db的同时 可以直接放在cache里面, 因为很可能被读. 3要注意engineer 在线和不在线的情况. 4 哪种event 需要通知哪个engineer 可以是一个pub sub的架构也可以存成一个static table or key value. 这一轮面试官40分钟问完, 问了很多细节的问题, 因为我比较熟, 所以面试官非常满意. 还剩20分钟 说我表现得非常好, 他一边给我写feedback 一边让我问问题。
 - 设计monitoring system监控100个server, 需要从display到存储到采集数据整个stack都涉及到
 - 设计distributed Logging System
 - 设计一个诊断系统, 类似地理说的Kafka加上aggregator的设计方式。
- 1. 系统设计 : 超高频top k / DATADOG
 - 5. top k article, 5分钟, 1小时, 一天的, 问了怎么传输, 怎么存储计算, 有几个模块等等, 这轮感觉都答上了。
 - DS&Algorithm : 类似top k的设计, 需要return 5min, 1hour, 24hour之内的数据整合。这里问的不是top k, 而是mean - 股票交易系统, 经常有price update (incoming message), GetAvg会get过去不同时间区间内的average price. 需要O(1)
重点是深入rolling window这个ds : 怎么删去expire的数据, 如果靠incoming message trigger expired data deprecation, 一直没有incoming message怎么办。假设这个fixed sized window可以fit进memory
 - 跟面经不太一样...没有很熟bucket, 卒
- key value store,
 - 从读写低延迟, 可靠性, 拓展性出发, 按照level db的思路答了。
 - 支持基于单个KEY的insert/update/delete/fetch 操作, 基本上照着RocksDB/Couchbase的实现来聊的
 - high level聊了consistent hashing, 以及如何加减virtual node。single machine聊了memcache的一些知识点, 比如lru, slab allocation。
- top k exceptions,
 - 前半部分一直被国人大哥challenge, 国人妹子后来引导我, 磕磕绊绊过去了, 后面聊得还行。个人感觉系统设计完全看面试官, 每个面试官的风格都不一样。比如我问time bucket按多长时间算, 他说up to you, 我说那就按1min算, 他问为什么要按1min算。沃尼玛, 别的面试官都是直接说按1min来算的好吧, 我说因为用户需要看5分钟1小时的数据, 1min不多不少。面试官说我的解释太模糊, 需要有数据支撑。。。还问我为啥用redis, 我说这里的数据schema redis可以满足, 并且是内存型数据库所以速度快, 同时还可以持久化到硬盘上。面试官说这里我们不需

要速度快啊，我说对于5分钟1小时的数据我这种方案需要online，近乎实时的，所以需要速度快。然后面试官还问为什么要用redis，其他的key value型数据库为什么不行，我说只要能满足这种场景的key value型数据库都可以用，只是redis也符合，所以我就说了redis，没说其他的key value型数据库不能用。我觉得面试官有些点就是在跟我抬杠。。。而且我面的不是senior，还是同胞，用得着这样吗。。。既然都up to me，那我说啥就是啥呗

- topk但是在描述问题的时候包装铺垫了很多内容，其次和之前面经不太一样的是，不仅要过去一小时一天还要求可以调取过去一年的data，这样只用redis就不够了，得上SQL。我提出用redis cluster cache + SQLbackup也是可以实现的，计算了一下redis memory使用量也在合理范围内，但还是感觉不是他想要的。
- k is configurable
- <https://www.jiuzhang.com/qa/219/>
- 一天里LinkedIn member转发最多的content，（URL），如何设计。楼主用的sqs + aggregation service(write) + reading services三层做的。中间讨论了很多数据的size和design的tradeoff，聊的还是挺顺的。
- 设计一个metric系统，包括怎么收集，aggregation，存储，查询，dashboard，alert
- 我靠。跟地理的面筋相差很大，好么！地里都说是搞什么loosy counting 和各种approximation. 然而，面试官跟我说，
- 一个cluster有很多server，每个server会时不时发出一些exception. 需要我搭建一个设计系统，它要提供一个api：getTopExceptions(int K, Long t1, Long t2).
- =>整个cluster在[t1, t2]区间里top K的exception的信息，包括stack trace什么的。
- 这里t1, t2都是以秒为单位，而且t2-t1可以是好几年那么长。这个K呢？我再三向他确认，他说可以是10, 100, 甚至是100000. 我就跟他说，
- 这个api是不是太flexible了。一般不是过去5分钟，1小时，一天，一个月么。而且谁会看top 100000的exception呢？他说我们不能更改要求，必须按照要求来。。。。
- You have a lot of servers and each is producing thousands of logs per second. You have unlimited resources. Design a system that will aggregate all the logs from all the systems, for a given window of 24 hours. And need to return the top N exceptions.
- 聊performance和scalability BD
 - 也是一个大年纪华人问的是给一个10个B的table data table 一个user table 然后怎么做join return 比如说top hot sale product。考虑sharding 考虑index 考虑partition data aggregation就差不多了
- event system,
 - 怎么scale 然后怎么实现realtime analysis 基本上就是linkedin他们自己产品 了解下都没问题 我其实都回答上来了 只是后来对面说时间还有多少
 - meeting invite的系统。
- 一开始是给你一堆file，让你返回包含指定内容的list of file name。

一开始我有点想偏想用tries去搜索，后来想想不用，直接做hashmap就好了。之后问题就偏到了分布式文件系统上面了，如何储存，如何cache，如何更新，如何维护。小哥在不断的加要求，我感觉也不是在为难我，是真的在考察我。最后小哥表示我回答了他想要的，他目前就是干这个，detail能问一天，orz。

- Document Search

6. system design 就是这位国人大哥了 document index. 我说用map reduce 来build index, 然后可以存在trie里。他不大想用trie, 问我trie和hashmap有啥区别, 我说trie节省空间, 他表情不同意的样子。后来决定index决定存db里。问了如何scale, 我说可以partition data。他说如何sharding, 我说可以consistent hashing。他说具体怎么做, 我要正要画个圈给他讲consistent hashing, 他说不用算法, 我们是high level system design, 我就给他画了web server, load balancer, db的图 (according to recruiter, 他说他需要提醒我需要从high level design的方向思考)。中间提到了不同db的优缺点, 如果搜集top k的关键字, 我讲了memcached和redis的优缺点, 和如何按bucket存频率。

- 设计领英0,1,2,3朋友圈的具体算法。

- 实现就是双向bfs, 2,3度用双指针。最后还可以继续用hashset来优化时间复杂度。印度小哥积极引导, 我也一直在顺着他说, 最后总算是都写完了。
- 设计二级好友三级好友

- Spell Checker

- Malicious

- 设计一个全球范围内的blacklist service, 就是有很多恶意ip会发来ddos攻击, 你要设计一个blacklist的服务, 能够ban掉之前已经诊断为malicious ip发过来的请求。这里不要求你设计怎么样判断一个ip是否是恶意ip, 给了个isMalicious()的api signature。难点在于不同data center之间怎么sync数据, availability和consistency怎么取舍。哪里会有single point of failure, 然后怎么设计能解决。最后followup就是结合你的工作经验问这个服务上线之后你最想加一个什么功能, 不一定是functional的, 可以是logistics上的。面试官比较期待的答案是support和monitoring之类的。
- 三哥面试, 全球有好多个data center, 如何检测用户的异常登录 (例如今天还在亚洲明天就来美洲了), 如何防止DoS攻击, 不同center怎么样共享学习到的信息——例如黑名单什么的。

- 设计用户activity收集系统 这个系统需要回答一些analytical的问题 觉得本质就是OLAP系统

- 设计trending [linkedin](#) share post