

Alice 分享面经，请大家积极讨论，留下想法，解题思路，每一道题下面有绿色思路，请大家follow 格式  
这个群的目的是大家共同进步，希望大家顺利面过

### 1. 3.28 -

<https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=617243&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3089%5D%5Bvalue%5D%5B3%5D%3D3%26searchoption%5B3089%5D%5Btype%5D%3Dcheckbox%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311%26orderby%3Ddateline>

time line

2月初 recruiter 联系， 2月底电面， 3月底onsite

谷歌Virtual五轮，早上10点开始， 每轮45分钟， 午餐休息一小时

1. coding : 类似LC843 猜词语， 越少次数能猜出来secret word得分越高， 问怎么样能deterministic的generate出来下一个要去猜的词语， 需根据系统的返回值调整

思路

2. coding : 设计一个飞机降落的系统， 在N分钟内只有一个降落的request可以true。 进来的request有ID 和timestamp， 问什么样的数据结构能实现该功能？

brute force : 用一个list, 保持sorted, 每次进来新的request, Binary Search找到离他最近的两个timestamp, 比较是否都大于K time=  $O(n \log n)$

Follow up : 优化时间 建一个BST, 每次新request进来, 边search位置, 边比较值是不是within K, time=  $O(\log n)$

思路

3. Design : 设计一个API接受用户request, 返回response。从头到尾都不知道这个API要干什么。定义request, response和 service class

要求能实现用Request找到Response, 实现 takeRequest ()

request进来太多, 如何handle

callback根据response的status, call downstream service

思路

4. coding : 给一个sorted 数组[-3,4,1,2,3,4,5], 求出平方值第K大的index BruteForce: 见一个 wrapper存index和值, 求出所有平方值, 然后sort得到第K大  
要求优化时间到O (n) , 两个指针左右走, 从第一个正数开始, 左右比较, 谁小移谁

5. BQ 轮, 问了快10个问题

2. 3/28

<https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=617689&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3089%5D%5Bvalue%5D%5B3%5D%3D3%26searchoption%5B3089%5D%5Btype%5D%3Dcheckbox%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311%26orderby%3Ddateline>

第一轮:

题目背景:

你是一家游戏公司员工 你们公司有一个游戏 每个玩家有自己的id 和每次得分

你想统计一下 周末两天 玩家的得分情况 请写一个function

input是 两个list of tuple 分别是周六和周日的得分情况 tuple里面是(id, 当日总得分)

output是 一个list of tuple 是每个id周末两天的得分总和

input

A\_sat = [(1, 100), (2, 99), (3, 87)]

A\_sun = [(2, 79), (1, 97), (3, 83)]

sort A\_sat according to id

output

Dict to save A\_sat

follow up1:

如果input的两个list 是sorted

output也要保持sorted

怎么做

follow up2:

如果input的list里是每个人每次的得分 (id有重复) 怎么做

A\_sat = [(1, 100), (2, 99), (3, 87)]

follow up3:

如果公司里不同人需要不同的output 比如我需要 周末两天的得分前三名 怎么做

heap with size 3

follow up4:

如果想要这个function output一个stream 怎么做 (就比如 call一次 output 第一名 再call 就是第二名 etc)

Dict to save A\_sat, heap to save the sum of each id

这一轮比较简单 但是面试官 问了很多python基础知识

比如:

最开始的问题 有没有办法空间优化 (不行因为tuple是immutable)

我写了 for l1, l2 in list1 + list2: 面试官问那个list1 + list2 是什么意思

其他的不记得了 都是中间穿插问的

第三轮:

题目背景:

我有一群电脑(node) 是一个tree的结构 每个connection有个传送速

node的定义是

node = {node.id, node.speed}

tree的结构是:

treenode = {node, list of node}

问给一个文件size 最短多久可以从root电脑传到所有电脑上

follow up:

如果是graph (有环) 怎么做

multicast

<https://leetcode.com/problems/network-delay-time/>

第四轮:

题目背景:

写一个function input是一个句子(string) 这个function 要一直读 直到有similar的词出现  
两个单词similar的定义是:

如果单词1 除了最后一个字母之外 和单词2是一样的 那这两个单词是similar的

ex:

worm

word

follow up:

similar增加一个定义

如果单词1 去掉一个字母 或者 加上一个字母 和单词2 是一样的 那这两个单词是similar的

ex:

word

words

-----我是面试分割线-----

第五轮:

面试官给了一个function header 让我完成function

function(error: list[list[int, int]], t: int, min\_rate: int, len: int) -> boolean

return true if there exists an interval with length len that includes time t and the minimum error rate of the interval is min\_rate, otherwise false

[[[1,3], [4, 6], [7, 9]], [[1,3], [4, 6], [7, 9]]]

follow up:

function(error: list[list[int, int]], threshold: int) -> boolean

return true if there exists an interval where length of interval \* minimum error rate of the interval >= threshold, otherwise false

### 3. 3/28 <https://www.1point3acres.com/bbs/thread-617887-1-1.html>

1.warm up 是给一个字符串只有1, 0, count substring only contains 0.。比如 10011000 就是0, 0, 00, 0, 0, 0, 00, 00, 000 返回9

follow up是 给一个matrix count submatrix only contains 0.

#### 2. redacted document

给两个file, 一个是原始输入A 超级大, 内存放不下的那种

第二个B里面包含要去重的word (一个word 一行, 内存放的下)

然后输出file C

eg : A : like a cat work hard pretty

B:

a  
work,  
pretty

那么C输出就应该是：like cat hard

follow up讨论各种corner case 比如有标点符号怎么处理， 比如一个word 换行了怎么处理之类的。。

很多case 就要和小哥不断交流

换行就是

XXXXXXXX wo-

rk das

这种情况wo-rk也要去掉

3. LC 就死溜 变种， pop数组给出， pushed 就是1 到 n的顺序 先用了time  $O(N)$ , space  $O(N)$  的解法做了

然后follow up 是space 变成 $O(1)$ ， 不考虑time。。。

小姐姐人超好不断提示。。。感觉就差给答案了😁。。。

就是 如果pop里面存在顺序是 a, b, c 的三个数子  $a > b \ \&\& \ a > c \ \&\& \ b < c$  那就一定是invalid的 比如 3 1 2

4. warm up： 给一个字符串其中如果重复字符连续出现 3+ 次那么就是stretch letter了 把多余的letter 去掉。。这个string 就是一个stretch word

eg : helllllloo -> helloo

第二题是给一个stretch word 和一个indictionary() function， 判断这个stretch letter是不是可以从 dictionary 中某一个word 生成 dictionary 里的单词一定不是stretch的

本质就是求stretch word的所有possibility 然后调用indictionary

eg helllllloo -> hello, helo, heloo, helloo

backtracking。。。楼主用了recursive 的写法。。不过小哥哥想要iterative的写法。。但是最后讨论完 time, space complex 一样就没写了。。时间也到了

5.第一题：geeksforgeeks.org/form-minimum-number-from-given-sequence/

第二题：evaluate division, 给的初始文件很大， 然后内存有限只存的下所有出现过的letter， 要求 query 的time 为 $O(1)$ 没时间写代码了。只讲了思路， 就是预处理union find

比如文件里  $A/B = 2 \ B/C = 3 \ D/E = 5 \rightarrow$  那么内存里存 B, 2A | C, 6A | E, 5D 这样

#### 4. 3/25

<https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=614424&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3089%5D%5Bvalue%5D%5B3%5D%3D3%26searchoption%5B3089%5D%5Btype%5D%3Dcheckbox%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311%26orderby%3Ddateline>

一共面了5轮，上午三轮coding，下午一轮coding，一轮behavior。每轮45分钟，轮和轮中间没有给休息时间。中午给了一个小时吃饭。

**Coding的四道题如下。**每轮都是在gdoc上写的，先讲思路，然后写代码，come up test cases，然后手动simulate代码执行在test case上，确认正确性。

1. Print expressions with special syntax representing repetitive substrings. E.g. 'abc(d)<2>e' --> 'abcdde', 'a(bcd)<2>e' --> 'abcbcdde', 'a(b(c)<3>d)<2>e' --> 'abcccdabcccdde'. 括号中的部分是重复的string, <>中的数字是重复多少次。

解法: 用recursion或者stack来展开重复的部分。每次遇到'('的时候叫这个recursion function.  
time  $O(n)$ ,  $n$  是output string的length  
space  $O(n^2)$ , worst case 是recursion在每一个字母都发生。

Follow-up: Does you code handle <0> situation? How to handle ill-formed input string?  
扫的时候确认下，每一个'(' 一定有一个')'，以及<>中间的可以convert成non-negative integer, 否则报错。

2. 给个string, 和两个字母，两个字母的最小距离。字母可以重复。E.g. 'GOOGLE', 'G', 'E', --> 2, 最后一个G和最后一个E 差2位。'GEOGLE', 'G', 'E' --> 1, 第一个G和第一个E差一位。

解法: 用两个pointer, 查询当前的字符时候是给定的两个字母。如果是而且两个pointer的字符不一样，update min\_distance. 每次挪动字符的时候只挪动前面的那个。  
time  $O(n)$ , 两个pointer扫了input string一边  
space  $O(1)$

Follow-up: 同样的问题放到了2D matrix上。算最小的Euclidean distance.

```
G 0 0 G
0 0 0 0
0 E 0 0
0 0 E 0
```

--> 3, 左上角G和第三行的E的距离为3。

3. design and implement a Versioned Key-Value Store. Should support the following operation:

```
kv = KVStore()
kv.set('a', 0)
kv.set('b', 1)
kv.get('a') --> 0
version_id = kv.version()
kv.set('a', 2)
kv.get('a') --> 2
kv.get_version('a', version_id) --> 0
```

解法: KVStore里用hashtable来存key --> VersionedItem, VersionItem里用另一个hashtable来存version --> val. KVStore里用一个variable来表示current\_version, 每次叫version(), increment current\_version variable.

唯一tricky的地方是get\_version的时候, 有可能这个key, 没有这个version (为了space, 我们没存), 那么做一个binary search来找最大的小于等于version的版本返回。

4. Given a budget (W), buy ads. The costs and rewards of ads on each day are given by array (rewards and costs). If you bought ads on day t, you cannot buy ads on t+1. Find the max reward under budget.

解法: dynamic programming.  $\text{max\_reward}[\text{date}][\text{budget}] = \max(\text{max\_reward}[\text{date}-1][\text{budget}], \text{rewards}[\text{date}] + \text{max\_reward}[\text{date}-2][\text{budget}-\text{costs}[\text{date}]])$

Follow-up: return the dates of ads that achieved the best reward.

### Behavioral

聊聊天, 问问怎么handle conflict, 描述一个没有方法meet deadline的situation和你的解决方案。

### 5. 3/25

<https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=615956&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3089%5D%5Bvalue%5D%5B3%5D%3D3%26searchoption%5B3089%5D%5Btype%5D%3Dcheckbox%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311%26orderby%3Ddateline>

1. 国人小姐姐, 一维01数组, 找所有只含有0的subarray的数目, two pointer,  $O(n)+O(1)$ ; follow up 是将一维数组变成二维matrix, 求所有只含有0的submatrix的个数。我一开始给出的不是最优解, 基本思路是对于每一行, 先调用一维的方法算出个数加到最终的结果上, 然后与向上的每一行做或操作, 每次结果调用一维的方法, 就能得出以该行为结尾的只含有0的submatrix个数,  $O(n^3) + O(n)$ , 空间是做or的时候需要一个temp数组。在提示下, 做到最优解, 是 $O(n^2) + O(n)$ :

具体做法是, 在从上到下对每一行进行处理后, 用一个一维数组维持一个向上看有多少零的数组, 然后通过求histogram面积的办法(可用单调栈), 可以得出以currRow作为结尾的新增全0 submatrix的个数, 例子: 0 0 1

0 0 0

0 0 0

假设现在求第三行, 上来先对第三行调用第一问的函数, 求出仅是第三行增加了6个全0 submatrix, 然后向上看 可以得到 2, 2, 1, 这样的数组, 求每个index能生成的柱状图的最大面积, 第一个柱+2, 第二个柱是+4, 第三个柱是+3, 就是额外新增了9个。所以以第三行为结尾的全0 submatrix共有15个。时间有点久, 有不对的地方还望指正, 谢谢。

2. 国人小姐姐，一个一维数组，找出一个最长的subarray的长度，并且这个subarray的 $\max - \min < k$ 。用treemap+two pointer 解得， $O(n \log n) + O(n)$ ；感觉的小姐姐希望我用priorityQueue解，不过用priorityQueue的话，删除会使得时间复杂度变为 $O(n^2)$ 。后来想了下，最优解可以做到 $O(n) + O(n)$ ，用两个deque，类似利口二三九；第二题是实现一下binary heap。
3. ABC 小哥，给一个二维int数组，从左上到右下，只可以从数字低的往高的地方走，a. 是否存在左上到右下的路径，dfs b. 找出所有路径，把dfs改一下，全搜索 c. 如何优化，memo+dfs 类似利口 三二九
4. 国人小哥哥，利口四一原题， $O(n) + O(1)$ 。第二题，给一个string，如果里面有一些char是一样并且长度大于等于2，就输出这些range，follow up是给一个API isWordInDict()，给一个string，问哪些dictionary string可以通过给定的string得到，backtrack。
5. 有个拍卖会的背景，拍卖会上有1-k个参与者，每个人都assign了一个有1-k数字的牌子，为了防止有人举牌子的时候正反导致混淆，需要将一些会导致混淆的数字加上下划线，就是求哪些是confusing number，参考利口一零八八。corner case比较多，比如如果数字反过来不在1-k范围内就没必要了，数字例如100，反过来是001也是例外等等，一定要正反看起来一模一样才需要加下划线。

因为之前的5轮理应有一轮是bq，但是面试官搞错了，所以一个星期之后补面了一轮bq

所有面完后的接下来一周，hr就通知过了hc了，感谢！求NYC捞！

在准备期间，有几个好帖子，分享给大家：

<https://www.1point3acres.com/bbs ... 6orderby%3Ddateline>

<https://www.1point3acres.com/bbs ... read&tid=446944>

<https://www.1point3acres.com/bbs/thread-448608-1-1.html>

## 6. 3/25

<https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=616108&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3089%5D%5Bvalue%5D%5B3%5D%3D3%26searchoption%5B3089%5D%5Btype%5D%3Dcheckbox%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311%26orderby%3Ddateline>

题目如下

(1) 给你一堆字符串找最长的字符串。最长字符串是字母连续的。举例子: [a, ab, c, abc, aebc, m] 的最长子串应该是 [a,ab, abc, aebc].

做法建图 + topological sort。从最小入度出发，一定是最长的字串。这道题还有几个我猜测的follow up 比如这个input 特别长，建立不了图怎么办？或者input 太长，内存读不下怎么办。

这时候你要上elastic search 或者 node4j。牺牲时间换空间

(2) 三哥要我实现一个 程序，使得一个program在A时间内只能发生B次。其实这就是个rate limiting。正确的做法应该是bucket token。完美的实现应该是谷歌GUAVA。



follow up : 所有的function都要能够最终被call。怎么做？你可以实现一个fifo queue(不建议)。你也可以给一个call back wait time。我推荐后者，反正谷歌这个开源的包怎么实现的，你就怎么做这个题目。。。

一模一样的题目，前后一起面。一个是白人小哥，一个是三哥。一个hire，另一个是no hire。可见面试的bias之大。最后还一个三哥，他用了两个trick挂我。首先，引导我走向错误的一条道路，让我去实现的滑动窗口算法。这个算法有很多问题，我开始也说了，但是他还是要我实现。然后很多follow up在这个算法里也解决的不好。我现在懂了，如果面试官有时间要你问问题，你最好选择别问，选择继续做题(你如果开始就100%写对了，就绕过这个情况)。因为你问的问题其实无关痛痒。答出够多的follow up比问问题要好。前面1, 2个面试官都问我，要不要问问题，我说不用。你们继续问follow up。可是三哥就是要你问问题。当然有好三哥也有坏三哥，我们不能一棍子打死

## 7. <https://www.1point3acres.com/bbs/thread-611967-1-1.html>

1. 一个数组初始为空，不停的有数字append进来，对于每个新加入的数字，返回前面有多少比这个数字小; follow up: 如果数组是个sliding window怎么搞 (time stamp)

2. 实现user api : setTimer(time, callback), 内部有一个hardwareTimer和api internalTimerHandler 时间到会调用，但是hardwareTimer只会trigger最后一次set的时间，怎么实现user api

我写的挺清楚吧，就是给你两个api : internalTimerHandler, setHardwardTimer, 用这两个api去实现user api : setTimer, cancelTimer。问题就是内部的hardware timer只会trigger最新设置的那个时间。比如 : setHardwareTimer(10), setHardwareTlmer(7)只会在时间7 trigger internalTimerHandler。

思路就是hashmap记录所有的用户callback，然后去找下一个需要trigger的timer。算法不难就是很多可以优化的地方，还有corner case

3. design : key-value storage

4. bq

5. count the unique webpages inside the website given: 没有任何followup, 面试官想看到的是document down what is unique (给出所有例子), what is inside website; 再写code, 再写 testcase。感觉考察的是软件开发规范，不是算法

## 8. <https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=609590&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3089%5D%5Bvalue%5D%5B3%5D%3D3%26searchoption%5B3089%5D%5Btype%5D%3Dcheckbox%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311%26orderby%3Ddateline> 帖子里很多讨论

leadership的题非常怪，和其他家的bq差别很大。是给你一个假定的场景问你会怎么做，往往是一个进退两难的场景。。。

Coding:

1. you have a list of current meetings and one meeting room with start and end timestamp when a new meeting coming in, check if it can be scheduled. Current meetings will not conflict with each other.

Example:

Current: [[0,1],[3,4],[7,9]] New:[2,3]

return Yes

答：过一遍，一个一个check

Follow up 1:

假设current meeting不变，但是有stream of new meetings要check怎么办

答：sort ->binary search

Follow up 2:

instead of one meeting room, there are four.

答：先把所有和new meeting conflict的拿出来，combine with new meeting，然后用计算最大需要几个房间。这个计算可以用LC 253

Follow up 3:

four meeting room 加上stream of new meetings

答：把满足四个meeting room的meeting压缩成等价于满足一个meeting room的meeting list, 再sort, 再bs

follow3做法中压缩的意思是:比如有四个已知会议 [(0,5),(1,6),(2,7),(3,8)] 且有四个会议室可以使用可以压缩成一个已知会议[(3,5)] 且只有一个会议室

上面两种情况对new meeting加入的返回结果是等价的

算法和LC253基本一样

把每个meeting的起始时间和结束时间放到一个list里面sort by time

loop这个list遇到一个时间是起始时间count+1,遇到结束时间count-1

就会得到时间和对应时间需要的meeting room的列表类似[(t1,0), (t2,1), (t3,2), (t4,1)]

然后把所有大于等于（其实只有等于）4的interval弄出来

这些interval每一个视作一个meeting，这样的meeting用一个meeting room去和新meeting match得到的结果和用原始数据去match是等价的

2. 给一个array of integer not sorted

return 一个array满足：

所有奇数index的数要大于等于该数左右两边的数

Example : [1,2,3]

return: [1,3,2] or [2,3,1] 返回一个即可

答：一个一个append, 不满足条件的话倒数第一和倒数第二调换一下就行了

关于第二题：stream of data是说有很多new meeting要check 但是check完不用把new meeting加进已有的会议列表

3. 用户有一堆GPS point with timestamp like [lat1,lng1,ts1],[lat2,lng2,ts2]

其中有一些是真实值，问（那些不是真实值的点每一个）和（真实值所连成的线）的距离是多少  
没啥算法，直接过一遍就行了，这道题的edge case很多

Follow up：

这个follow up和题没啥关系，开放式问题，问怎么判断一个GPS point是不是准确的

4. 说有一种最近美国很流行的棋牌游戏中的一种：

每一张牌有以下几个属性：1. 颜色 2. 形状 3. 数字 4. 填充物类型

每个属性都有三种，比如颜色有红黄蓝 形状有方圆三等等

玩家有三张牌，当这三张牌四个属性的，每个属性都相同或者都不同时，为胜利  
写一个判断三张牌是否胜利的function。。。

Follow up：

每个属性不止三种（不知道每个属性max有几种），玩家不止三张牌的情况

无力吐槽的一道题，这么复杂的题干，但是解法不需要算法。。。

就是for loop过一遍，每个属性出现的都放到一个set里

原题是size(set()) = 1 or 3

follow up就是size(set()) = 1 or k

k 是玩家手牌数

我觉得只用到for loop的这种都不算算法题吧...

9. <https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=609442&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3089%5D%5Bvalue%5D%5B3%5D%3D3%26searchoption%5B3089%5D%5Btype%5D%3Dcheckbox%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311%26orderby%3Ddateline>

没有电面 直接onsite 面的5

第一轮：coding "It is 'very good'". 要求tokenize成["this", "is", "very good"]. 就是单引号里面的不变 单引号外面的按space分开

followup：给你1000台机器 一个file有petabytes大 只有一行 按照上面的规则tokenize

第二轮：花了一会弄明白了规则 有这么一副牌 每个牌有4个attribute，分别是shape, color, number, shade,

定义一个set是3张牌，这三张牌的每个attribute要么全部一样 要么全部不一样

1: 实现is\_set(c1, c2, c3)

2: 实现find\_set([list of cards]) #只有有任何三张组成set就可以返回true

第三轮：coding

1: 怎么unit test一个函数 compute(op1, op2, operator) 这三个parameter都是string形式的 compute是perform加减乘除的函数0-

2: LC meeting room II

第四轮：behavior 吹牛

第五轮：设计 message queue

10. <https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=609384&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%255B3089%255D%255Bvalue%255D%255B3%255D%3D3%26searchoption%255B3089%255D%255Btype%255D%3Dcheckbox%26searchoption%255B3046%255D%255Bvalue%255D%3D1%26searchoption%255B3046%255D%255Btype%255D%3Dradio%26orderby%3Ddateline&page=1>

刚刚收到消息hc挂了，说有2轮code不够clean，有明显的bug。有两轮很好，有一轮题目没解好。

1. bq, 遇到认识的中国小哥，人很好，我运气也好，给我水果了

2. 一个三哥，给我任意一个string，让我设计encode和decode的方法。string可以是任意字符，要支持nested encode/decode，懵逼，不会，要了很多hint，work close to the solution，最后还是他说出答案了，这一轮很不好。

3. 白人小哥，给两个种子数子，比如1 and 2,  $1*2 = 2$ , string就变成 122, 每次拿末尾的两个digits, 比如22, 就变成1224, 一直下去，直到循环。返回前面非循环的数字的长度，这也很简单，把每一次做计算的两个digits当作pair存hashmap里，每次做运算时就拿出来比较一下，遇到一样的就循环了。算秒了吧，code写了，对方让我优化，我问优化时间还是空间，开始优化空间，比如不用每次存string，也不需要get string.size(), 只需要拿一个数字来记录长度就好。讨论到这里时间到了。最后feedback说，code不够clean，狗在google doc上写code，一边想solution一边写，看来还是要多注意细节，不能让对方抓到缺点。

4. 印度三哥，人很阳光，给了我很好的feedback。说给一堆task的log，有task name，开始和结束时间，模拟单线程运行，我说hashmap可以做，不难，描述了解法。对方说，换个数据结构，我说stack也可以，对方说好，拿stack。我速速写了，然后他follow up，改变了一点task时间的计算方式，也不难，我描述了一下方式，他优化了一下我的解法。结束了。

5. 白人小哥。模拟读network package。给一些network package包，没有顺序，只有offset, size, const char\*. 设计一个read(int size) api，每次都读给定已有包的package内容。我想了想用priority queue，定义了package的object，再弄一个全局的read的数字来track读了多少了，按照offset来排序priority queue，每次都从top() element来读。我的解法木有问题，代码我觉得也没问题，可能就是没有写priority\_queue的comparator，比较code没有让运行，有些syntax的细节不够好，但当时要理解题意然后写出答案还是有些慌张的。还有，用cpp面试真的不 friendly。

几天后hr说送hc了，说overall不错，有两轮很好，有三轮code不够好，有点bug。挂了。有点伤心，连挂了几场了，心态有点不稳，而且视频面真的很多信息都体现不出来，对面试结果肯定是有影响的，毕竟大家都是人，人和人多维信息的互动还是很重要的。

11. <https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=607991&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3089%5D%5Bvalue%5D%5B3%5D%3D3%26searchoption%5B3089%5D%5Btype%5D%3Dcheckbox%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311%26orderby%3Ddateline>

Node {

set<Ports> neighbour;

int val;

}

void send(port, message);

void onRec(port, message);

Each node has a set of ports, and the ports connected other nodes. each node can send the message to the port that connected with other nodes which these node will receive the message.

send method

Problem: Implement void onRec(port, message) to find the minimum value among the network. You can send any message.

Question 2

Given a list of tasks

T1(10,5) mean start at 10 and last 5

T2(25,15) means start at 25, and last 15

The system can only schedule at most one task at a time.T(18,7) write a algorithm to decide if the task can be schedule

Follow up: the system can schedule at most k task at one time

Question 3:

cit.next()

cit.stop()

Each time call cit.next() return a index pair which indicate building a new tower,

Start from (0,0), a man can only go on tower. find the minimum towers need to build to reach the end destination.

01234  
0STTT  
1 T  
2 TT  
3 T  
4TTTE

Question 4:

A string called extension string with one character repeated more than 3 times.

For example: helloooo is extension string because o last 4 times.

hellloooo is extension string because l and o last 3 and 4 times respectively.

input: string w

output: return all extension index of the string.

For example: hellloooo return [[2,4][5,8]]

Follow up:

Given method:

bool isDictionaryWord(w)

Implement: isExtensionDictionaryWord(w)

when if isDictionaryWord return true;

when w can be collapse to a dictionary word, return true;

otherwise return false

for example: hello is dictionaryWord, call method isDictionaryWord return true;

hellloooo is not dictionaryWord, but it can collapse to hello which is dictionaryWord.

Question 5:

behaviour: Lots of leadership questions.

**补充内容 (2020-3-12 12:35):**

第一题是distributed system problem, send, onRec() 都是 asynchronous DFS,BFS 解决不了。

12. <https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=607484&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3089%5D%5Bvalue%5D%5B3%5D%3D3%26searchoption%5B3089%5D%5Btype%5D%3Dcheckbox%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311%26orderby%3Ddateline>

今年一月底面的，没有电面直接onsite了，问的很简单，之后才知道面的是L3

1. 在一个有多个root的树上判断两个点是不是有共同的祖先节点。

比如：

```
      1       2
     3 4    5  6
    7 8 9 10 12 13 14
```

7 和 12 没有共同祖先，10 和所有点都有共同祖先

2. 写一个iterator class，就是有next和hasnext的那种

3. 在一个文件系统中，给一个路径，删除路径下的所有文件和目录，可以用一些api比如listFileInDir，deleteFile，deleteEmptyDir这些

4. DP题，一个商人在A，B两市做生意，每天在两市的收入可以是a1，a2。。an，b1，b2。。bn。在两市之间travel的cost是T。问n天内最大收入。

5. LP。一堆乱七八糟的问题。有一道题说是如果你安排个活动，发起了投票，但是结果出来之后组里有几个人不同意还是不愿意去，你怎么办。面试官提醒我应该当机立断一些，他们不愿意去就算了。

13. <https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=606725&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3089%5D%5Bvalue%5D%5B3%5D%3D3%26searchoption%5B3089%5D%5Btype%5D%3Dcheckbox%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311%26orderby%3Ddateline>

没有电面，直接昂赛特。5轮，4轮技术，1轮behavior。

1. 半设计半coding。设计一个metrics service，只要说出大概架构就好，然后实现其中一个小component，不难。

2. decode 字符串，3[a]2[b]3[c] -> aaabcccbccc. 用栈就行。followup: encode 要求最优解。aaabbbbaaabb 要encode成 2[3[a]3]，而不是3[a]33[a]3。followup挣扎了半天也没做出来。。。。

3. 外国小姐姐，面之前以为会是最简单的一轮，结果是最难得一轮。。题目是给一个矩阵，每个元素都是 / 或者 \，可以看成是木板。比如一个3\*3的矩阵张这样

[C++] 纯文本查看 复制代码

?

```
0  /  /  /
1  \  \  /
0  \  \  \
2
0
3
```

然后从第一行某个位置扔球，让你输出球最后从哪里滚下来。比如从[0, 1]扔，最终会从[2, 2]滚下来。好像也没什么好方法，只能一行行模拟。followup是，如果把某一位置的\变成/，或者/变成\，判断球落下来的位置会不会变。挺有意思的一道题。

4. 外国小哥哥，题想不起来了。。（哎呀，脑筋），但是是sliding window的变种。只是右指针挪动的时候，变量/cost/result/whatever 不是加1而是加  $2^{(\text{right} - \text{left} + 1)}$ 。想了一会做出来了。

5. Behavior question。当时狗家刚刚兴起behavior，还不是很成熟，面我的是个中国小哥，问题也比较简单，聊得还算愉快。

3周后给offer SDE 2. 但是相面 Senior，和recruiter聊了之后又加面两轮。本来告诉我是1轮coding，1轮design，结果是2轮design。。。。

1. 很传统的设计题，统计过去1小时的top 10 query。

2. 一个online订票系统。

感觉两道题都不难，但是都会考察你想的全不全，corner case怎么处理，并发性什么的。由于lz当时design真的不太擅长，最后加面失败！！这里建议也有加面需求的童鞋，虽然recruiter告诉我面不好是有可能撤回之前offer的，但是可能性微乎其微，只要你不和面试官打起来，一般都不会撤回。所以能加面还是要加面，毕竟生一个level差不少钱呢。

14. <https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=606492&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3089%5D%5Bvalue%5D%5B3%5D%3D3%26searchoption%5B3089%5D%5Btype%5D%3Dcheckbox%26searchoption%5B3046%5D%5Bvalue%5D%3D1%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311%26orderby%3Ddateline>

2月底的狗家onsite，估计是最后一波onsite interview，之后都改成virtual的了把

onsite本来是4轮coding加一轮bq，但是面bq的那轮搞错了又面的coding，所以面了5轮，回家后video又面了bq。后面了解到5轮都会提交给HC，HC会注意所有的alert，然后review code 挑最好的4个。所以面5轮也未必比4轮好。

1-3轮 上午 4-5轮 下午

1. 两个字符串，里面有backspace，问结果一样否。蠢口easy 题，巴是是。没想到栽在这上面了。用stack做出来。followup是不用额外空间，这应该才是正题。东欧小哥一路提示，勉强写出来，run一个test case，发现return不对。时间到了，这轮感觉很不好。



2. 地里面筋题。同是阿三小姐姐出的，\ 和 / 组成的matrix，给小球进口col，问出口col。之前的地里见过，感觉很简单，临场居然不知道怎么做。小姐姐提示了两次，相当于直接给了做法。代码是写出来了。followup是matrix里面有flip怎么办，和地里的其他面筋一样，precompute。

3. 一组牌->array 正整数。两个人轮流拿牌，每人每轮可以且必须拿1， 2， 或 3 张牌。直到牌堆拿完为止。最后score是拿到牌的数值之和。问如何最大化score。DP解。

4. 图。一个文件从一台计算机传到所有其他所有计算机所需的时间。其实就是算最短路径。Dijkstra解。后来知道这轮本来是bq的。

5. 生成一个5\*5的 bingo board，第一列值在1-15之间， 第二列16-30，。。baord里不能有重复的数。followup是要生成很多board，并且每两个board之间不能有相同的列。也很简单，先生产很多不重复的第一列，然后第二列，。。。之后一个一个assemble就好。

和hr刚打完电话，感觉第一轮应该no hire往下，二三轮lean to hire，四五轮hire往上。还是热身比较慢，到后面状态慢慢变好。

总结：

1. 第一轮Easy tag题没写完拿个no hire还是很伤的。刷TAG不要随便放过easy题，很可能是warmup，做easy题应该要把所有可能解法都做一遍。

2. 第三轮，做过的题，没有仔细听面试官的提示，把代码写的更elegant一些

准备时间有限，已经尽力了。HR说close to hire，不知道是不是安慰。

别人总结的：

word distance exchange 2

```
vector<int> minDistance(string &word1, string &word2) { vector<int> ans;
int size1 = word1.size(); int size2 = word2.size();
if (size1 != size2) return ans;
unordered_map<char, int> map1;
unordered_map<char, unordered_map<char, int>> map2; int i;
for (i = 0; i < size1; i++) { if (word1[i] == word2[i])
continue;
if (map2.count(word1[i])) {
if (map2[word1[i]].count(word2[i])) {
return vector<int>({map2[word1[i]][word2[i]], i}); } else {
ans = {map2[word1[i]].begin()->second, i}; }
} else {
if (map1.count(word2[i])) {
```

```

ans = {map1[word2[i]], i}; }
}
map1[word1[i]] = i;
map2[word2[i]][word1[i]] = i; }
return ans; }
bike
def assignBikes(self, workers, bikes):
def dis(i, j):
return abs(workers[i][0] - bikes[j][0]) + abs(workers[i][1] -
bikes[j][1])
h = [[0, 0, 0]]

```

```

seen = set()
while True:
cost, i, taken = heapq.heappop(h)
if (i, taken) in seen: continue
seen.add((i, taken))
if i == len(workers):
return cost
for j in xrange(len(bikes)):
if taken & (1 << j) == 0:
heapq.heappush(h, [cost + dis(i, j), i + 1, taken | (1 <<
j)])

```

Find in Mountain Array

```

class Solution {
int findInMountainArray(int target, MountainArray A) {
int n = A.length(), l, r, m, peak = 0; // find index of peak
l = 0;
r = n - 1;
while (l < r) {
m = (l + r) / 2;
if (A.get(m) < A.get(m + 1))
l = peak = m + 1; else
r = m; }
// find target in the left of peak l = 0;
r = peak;
while (l <= r) {
m = (l + r) / 2;
if (A.get(m) < target)
l = m + 1;
else if (A.get(m) > target)
r = m - 1; else
return m; }
// find target in the right of peak l = peak;
r = n - 1;

```

```

while (l <= r) {
    m = (l + r) / 2;
    if (A.get(m) > target)
        l = m + 1;
    else if (A.get(m) < target)

        r = m - 1; else
return m; }
return -1; }
} KMP
class Solution {
public:
    int strStr(string haystack, string needle) {
        int m = haystack.size(), n = needle.size();
        if (!n) {
        }
        return 0;
        vector<int> lps = kmpProcess(needle);
        for (int i= 0,j= 0;i<m;){
            if (haystack[i] == needle[j]) {
                i++, j++;
            }
            if (j == n) {
                return i - j;
            }
            if (i < m && haystack[i] != needle[j]) {
                j=j=lps[j- 1]:i++;
            }
        }
        private:
    }
    return -1;
    vector<int> kmpProcess(string needle) {
        int n = needle.size();
        vector<int> lps(n, 0);
        for (int i= 1,len= 0;i<n;){
            if (needle[i] == needle[len]) {
                lps[i++] = ++len;
            } else if (len) {
                len = lps[len - 1];
            } else {
                lps[i++] = 0;
            }
        }
        return lps;
    }

```

```

}
};

```

## Rectangel Sum MIN

1292. Maximum Side Length of a Square with Sum Less than or Equal to Threshold

```

public int maxSumSubmatrix(int[][] matrix, int k) {
    if (matrix == null || matrix.length == 0 || matrix[0].length == 0)
        return 0;
    int rows = matrix.length, cols = matrix[0].length;
    int[][] areas = new int[rows][cols];
    for (int r = 0; r < rows; r++) {
        for (int c = 0; c < cols; c++) {
            int area = matrix[r][c];
            if (r-1 >= 0)
                area += areas[r-1][c];
            if (c-1 >= 0)
                area += areas[r][c-1];
            if (r-1 >= 0 && c-1 >= 0)
            }
            area -= areas[r-1][c-1];
            areas[r][c] = area;
        }
    }
    int max = Integer.MIN_VALUE;
    for (int r1 = 0; r1 < rows; r1++) {
        for (int r2 = r1; r2 < rows; r2++) {
            TreeSet<Integer> tree = new TreeSet<>();
            tree.add(0); // padding
            for (int c = 0; c < cols; c++) {
                int area = areas[r2][c];
                if (r1-1 >= 0)
                    area -= areas[r1-1][c];
                Integer ceiling = tree.ceiling(area - k);
                if (ceiling != null)
                    max = Math.max(max, area - ceiling);
            }
        }
    }
    tree.add(area);
    return max;
}

```

## 587. Erect the Fence

straing num 5

```

class Solution(object):
    def isNStraightHand(self, hand, W):
        if len(hand) % W != 0: return False

```

```

count = collections.Counter(hand)
while count:
    m = min(count.keys())
    num = count[m]

    for k in range(m, m+W):
        v = count[k]
        if v < num: return False
        if v == num:
            del count[k]
        else:
            count[k] = v - num
    return True

Campus bike
class Solution {
public int[] assignBikes(int[][] workers, int[][] bikes) {
    int n = workers.length;
    Queue<int[]> q = new PriorityQueue<int[]>((a, b)->(a[0] == b[0] ? (a[1] == b[1] ? a[2] - b[2] : a[1] - b[1]) : a[0] - b[0]));
    int i = 0;
    for (int[] worker : workers) {
        int j = 0;
        for (int[] bike : bikes) {
            q.add(new int[]{Math.abs(bike[0] - worker[0]) + Math.abs(bike[1] - worker[1]), i, j++});
        }
        i++; }
    int[] res = new int[n];
    Arrays.fill(res, -1);
    Set<Integer> visited = new HashSet<>(); while (visited.size() < n) {
        int[] temp = q.poll();
        if (res[temp[1]] == -1 && !visited.contains(temp[2])) {
            res[temp[1]] = temp[2];
            visited.add(temp[2]); }
        }
    return res; }
}

Campus Bike2
def assignBikes(self, workers, bikes):
    bikes[j][1]
    def dis(i, j):
        return abs(workers[i][0] - bikes[j][0]) + abs(workers[i][1] -

h = [[0, 0, 0]]
seen = set()
while True:
    cost, i, taken = heapq.heappop(h)

```

```

if (i, taken) in seen: continue
seen.add((i, taken))
if i == len(workers):
    return cost
for j in xrange(len(bikes)):
    if taken & (1 << j) == 0:
        heapq.heappush(h, [cost + dis(i, j), i + 1, taken | (1 <<
j)])

```

#### 947. Most Stones Removed with Same Row or Column

```

class Solution {
    Map<Integer, Integer> f = new HashMap<>(); int islands = 0;
    public int removeStones(int[][] stones) { for (int i = 0; i < stones.length; ++i)
        union(stones[i][0], ~stones[i][1]);
    return stones.length - islands; }
    public int find(int x) {
        if (f.putIfAbsent(x, x) == null)
            islands++;
        if (x != f.get(x))
            f.put(x, find(f.get(x))); return f.get(x);
    }
    public void union(int x, int y) {
        x = find(x);
        y = find(y);
        if (x != y) {
            f.put(x, y);
            islands--; }
    }
}

```

#### 91. Decode Ways

#### 1096. Brace Expansion II 803. Bricks Falling When Hit

#### 727. Minimum Window Subsequence

#### count-complete-tree-nodes

#### 1145. Binary Tree Coloring Game 685. Redundant Connection II