

VO:

coding 1: 密码 地里原题

coding 2: expression 地里原题

Design: 问了一些比较吃经验的 troubleshooting, 没让设计系统很神奇

BQ: 基础 BQ 问题, 遇到过啥挑战, 最有成就的一件事是啥

coding1: expression, 建议自己练习一下文件读取, 面试的时候是要求这个的, 我用的 java scanner

coding2: decoder

design: inventory system, 聊了一下 race condition, 如何 scale

BQ: 常规题

coding -- password(read file) how to optimize the space, only read the content needed

coding -- expression(dfs)

sd -- inventory system

bq -- most complexed end-to-end project

coding 1: shoppers delievey 两问, 其中第一问是 exactly 蠡口仪器零一, 第二问稍微复杂一点, 看这个应该就可以

coding 2: data pivot, 整体不难, 就是用 java 需要注意, 它那个平台真的很烂, 不能手动 import date 相关的 package, 搞得我花了好长时间 debug 差点没写完, 后来还好发现直接比较 string 的顺序就能 pass, 不然手动写一个比较日期的算法会很麻烦

SD: design catalog, 地理有好多参考, 设计一个 self join 的表就行了, 主要 focus 在 API 和 db schema

BQ: 常规 BQ

他家的题是不是从来不会变, 我去年差不多这个时候面了一次挂了, 今年全部是一模一样的题。我都惊了。

VO: expression eval, parse password matrix, design inventory system, BQ

VO1: Pivot table。这题推荐用 Python 来做应该会方便很多。楼主用 Java 写起来很难受。。。

VO2: Average wait time

VO3: BQ

VO4: System Design - Category page

四场面 (2coding 1 BQ 1 system design) 安排在了同一天, 发挥不好, 没好好看地里面经, 发现是 coding 和 system design 都是原题

pivot table (3 小问, 做出来 2 道半)

第一问输入是 `List[List[str]]` 然后 `List[i].length==5` (就是每行), 相当于总共五列, (细节我记不清了 面完没有趁热打铁记下来), 有 `date`, `region`, `sales` 什么的。

然后`List[i]`就是一个“订单”

但是 tricky 的点是每个列指标不固定，就比如这个测试数据`List[i][1]`是“sales”但是下个测试数据的`List[i][2]`是“sales”

总共是三问，

第一问送分的，就是问总共表里 sales 综合

第二问是给一个`column name`，然后根据这个来 filter 一些 sales。比如说我指定“region”，然后根据 region 生成一个 table。比如“Canada”对应的原表中的数据有很多条，输出的时候应该把这些都合总了。

第三问是个排序，给一个 date，需要 filter 在此之后（包括该 date）的订单。然后表里面也会有`cost`和`sales`，需要计算一个 profit，并且是根据第二问的基础上也会制定一个`column_name`。这一些条件统计出来一个 profit 最大的项，再根据特定的输出格式输出（就是字符串`f"The most xxx is xxx"`）。

average time （2 问，第一小问是排序 第二问是二分加 最小堆/优先队列）

BQ 常规的问题，就是问一些较为基础的（比如你曾经翻过的错是什么）

system design -> catalogs (面试官在中期比较喜欢打断我说话，也是一个经验吧)

题意就是一个货物信息的 display，然后货物有各种标签 可以属于不同的类，然后题目就是设计一个 web page 来 display 这些货物信息。

我当时问了一些问题，比如用户分布是否是全球、是否有频繁的 update webpage 还是说就静态地展示。但是我的考官考察的要点是数据库，scheme 是怎么样的。

coding1:

pivot table，地里老题，值得注意的是需要进行一些格式转化。第一问是很简单的根据不同的 country aggregate total sale。follow up 就是在第一问的基础上加一些附加条件和进行一个简单的计算（比如 profit）做完会问你复杂度和 production 中如何优化

coding2:

average waiting time。参考 www.1point3acres.com

同样的也会稳复杂度和优化。

Iz 前两个 coding 都剩了不少时间，奖励自己 20 分钟尬聊

system design:

catalog system。很 typical 的 system design。前半个小时 apidesign, schema design。后半个小时讨论整体的 structure, 会比较详细的问每个 structure 中的每个 component 的细节, 比如 cache 中 key value pair 怎么设计, load balancer 具体会干嘛 blah blah

bq:

Iz 的 bar raiser round 因为面试官时间冲突 rechedule 了一次, 最后是一位 cbc 大叔面的, 整体都很 nice, 我觉得赶着亚麻的 Ip 练习就好

本来以为周四 hc 开会就会有消息, 但估计他们急着放假没空发结果, 希望下周能有好消息, 攒 rp, 攒 rp

首先就是建议不要一天面完全部轮次, 最好是分两天, 中间也留几天复习。楼主是在第一天面试结束后去补习的后续轮次内容。提前求米, 米不够了, 有问题我会尽量回答, 谢谢大家。

BQ

没太多要说的, 楼主个人观点是题是问不完的, 尽量每题目自己多说一点, 大家毕竟好用, 通用的例子也不是取之不尽, 楼主有几个例子反复用了两次略显尴尬

Coding: Data Pivot Table

地里常见题, 需要注意的是 Column 的名字不能 hard code, 具体三问都是有关某列-某列, 做些比较, 或者找出前五名之类, 计算很简单, 用到 hash。要注意审题交流来确认输出格式, 最后转换 hash 到对应的格式。楼主练习的时候并没有转换格式, 现场写并不难。

题目做完三问补问一些问题, 比如现在的效率是多少, 是否有更高的效率, 每个步骤效率是多少, 每个步骤是否能提前结束, 比如找到了提前终止之类。

Coding: Avg Wait Time

也是经典题目，这里一定要注意如果用立寇原题妖气令妖练习了的话别说漏嘴，题目本身跟餐饮没有任何关系。其他注意的细节有输入的格式并不会提前按照时间排序，楼主的解法是直接先排序。第一问毕竟原题，很简单

第二问是如果在第一题基础上问如果有个目标时间，几个工人可以让你的平均等待时间达标。最快解法就是用到 `bs` 和 `heap`，建议假装思考，别看到题就写，解法很短，楼主半小时就写完了，奖励了自己半小时尬聊。

补问的问题与之前差不多，问某步骤或者某参数的意义是什么，某个步骤效率是什么，总体效率是什么，是否能更快，存储效率是多少，为什么你选择用二分

System Design: Catalog

题目本身不过多描述了，也是老题。不知道题目是否每次都完全一样，楼主的列表每次只需要回复某标签的全部直属子嗣，包括标签以及物品，不需要包含子嗣的子嗣，也就省去了优化数据库或者缓存标签族谱的必要。可以理解为用户每次点一个某标签的按钮，后续标签按钮与货物出现，再点一个，再细分。

其他要注意的点是，清晰的表达数据在库中的储存形式，`request` 的格式与 `response` 的格式。楼主还被着重问了用户刚打开网页的细节，网页需要在用户互动之前显示些商品那么商品从哪来，是否要 `request`，用户们刷新会如何之类，建议提前思考一下。

后面楼主时间充裕，写了管理员视角，也写了一些补充的东西，图文会被保存到后续审核，所以建议聊天的途中把重要的东西全部写在板子上

本来四轮都是原题：

1 password

2 计算公式 $f1=2$, $f2=f$

3 sd inventory system

4 bq

第一轮 code signal 出问题, compile 不了 code, helloworld 都打不出来。面试官主动给我 reschedule 了个加面。

加面出了个新题, 题目特别难理解, 读题读了好久。

Your task is to implement a function `solution(key)`, which takes a string `key` and returns an encoded 5×5 matrix.

The encoding happens in the following way: the initial letters in the matrix are the unique characters of the given string converted to upper case and in the order in which they appear, which then followed by the remaining letters of the English alphabet in order (except "J" is replaced by "I" to fit in the 5×5 matrix).

Example

For key = "Instacart", the output should be

```
[  
  ["I", "N", "S", "T", "A"],  
  ["C", "R", "B", "D", "E"],  
  ["F", "G", "H", "K", "L"],  
  ["M", "O", "P", "Q", "U"],  
  ["V", "W", "X", "Y", "Z"]  
]
```

算法 1: 经典地里原题, password(3 问), 这个最详细了: www.1point3acres.com. 最后 5 分钟闲聊

算法 2: 经典地里原题, 找 pattern. Expression 那道题 第二问出 bug 了, 修了好一会最后修好

系统设计: 老题 Payment Service (着重在 API). follow up: DB 的选择, 怎么 scale DB. 感觉我说什么都回馈 OK. 但 HR 后来反馈是这轮拿的 No. 这是原贴: www.1point3acres.com (题目在评论区里)

BQ: 中东面试官, 深挖项目. 问得很细(着重说挑战, trade-off), 还有少于 BQ. 解决矛盾.

都是面经题

password

计算公式 $f1=2, f2=f1$, 求 $f1$ 那道题

sd inventory system

hm bq 烙印面试官态度感觉很差, 也没有听我在说啥, 说过的东西反复问, 黑脸

1 - SD - Design inventory system (add stock, reserve stock, fulfill stock)

2- Coding - Password - www.1point3acres.com

3- Expression - www.1point3acres.com

4 - BQ (Tell me about your project)

VO:

矩阵解码那个

字符串匹配

bq: 大部分技术, 一些 bq

SD: inventory magement

写码

1. pivot table

input1. sales table:

prodId, sales, cost, state, timestamp

input2. pivotColumn

第一问:

output total sales of given column

其他 2 问也都类似比如让求 $\text{profit} = \text{total_sales} - \text{total_cost}$ for given col

2. average wait time

SD:

category system

主要 focus on getCategories API 和 data model, request format

VO 安排在一天的, 总共四轮四个小时, 到最后我头昏目眩的。本来我想稍微晚一点在面试, 给自己时间准备一下, 但 HR 感觉很赶, 说 hiring pipeline 有好几个 candidate。VO 第二天就给答复了。

VO:

Coding 1: 找 pattern。第二问需要 DP，没写完。

Coding 2: 经典 password。三问加一问用最优空间复杂度。 www.1point3acres.com

Bar Raiser: 主要是问了经验。如何带团队，做决定什么的。更像是聊天。

System Design: inventory system. focus 在 transaction, racing conditions. 还问了 security 和 production deployment best practice. 面试的人有点心不在焉，看脸色比我还累

2.VO

1. 算法: expression 那题

2. 算法: merge intervals 两连问

3. 系统设计: 国人面的, catalog 那道题. 面试官之前在亚嘛. 问得太细了, 还会时不时问一些很奇怪的问题, 说是 bonus 的问题. 但不关于 SD 的问题. (我表示回答不出来, 然后我反问它, 然后搞笑的是他说他也不知道答案. ~导致最后连图都没画就结束了

4. BQ: 国人, 问的很细很细, 聊得有点生硬.

onsite 4 轮

第一轮 data pivot -> 简单的数据处理 input 是 2d array 处理的时候使用 hashmap 存一下就好啦

第二轮 average cook time -> follow up 用 priority queue + binary search

第三轮 bq

1. pivot table

以下内容需要积分高于 188 您已经可以浏览

sum of a column

sum of a column based on country

sum of a column based on date

average wait time

以下内容需要积分高于 188 您已经可以浏览

similar to 立口 一七零一

greedy

Design a catalog system

以下内容需要积分高于 188 您已经可以浏览

how to handle a product belongs to multiple catalog

API design

database schema

面试挂掉. HR 说第一题和 BQ 答的一般.

BQ

Design an inventory system: 要支持 reserve stock 和 clear reservation

Coding: Password 3 问 地里原题

Coding: Expression 地里原题, 只做完 2 问

VO1: SD

Category tree, 这个地里我只看到过一个原题, 当时主要准备了 payment system 没准备这个。。个人感觉可能这轮没发挥好

VO2:

Merge Intervals, 也在地里看到原题

VO3:

Password 经典题

VO4: BQ

没啥特殊的

2coding + 1 bq + 1 sd

coding 1: data pivot

coding 2: avg cook time

都是老题

面试官 challenge 了第一题为什么要用 pandas, 不建议使用。

BQ 就是亚麻类型的 tell me about a time when: project tradeoff/ conflicts/ negative feedback / work with junior

SD 也是老题, 设计商店的 catalog 系统, 主要 focus 在 DB schema, 讲清楚如何 data access pattern 去实现前端 fetch 类目以及每个类目中商品信息。

follow up: 1 一个商品可能会属于多个 category 怎么解决 2 如何 scale 系统 3 如何确保系统在正常运行

data pivot 可以用 pivot_table

avg wait time

每一个 coding 都问了很多小问和 followup

sd 网页 一直问 api design 的细节 后面变成这个面试官教我写, 问我同不同意。。。 (他还迟到了)

vo4 bq 临时换了个烙印, 感觉他很想下班, 回答都是 cool

coding: 公式题

coding: 读文件解密码

SD: 设计 inventory manage 系统

BQ

第一轮: Behaviour question

深挖简历和过去经历

need e2e feature deliver to l4 level (that's why I got downleveled to l3)

第二轮：Code Question (something about intervals) I suggest review leetcode interview questions (merge interval etc)

第三轮：Data pivot

need to use heapq

duration, time order is received

orders = [[4, 1], [3, 2], [6, 5], [5, 100]]

customer_wait = 4+(5-2+3)+(8-5+6)+(5)

Q1 find the average time the customer need to wait for the order to be completed

can only do one order at a time

def solution(orders):

total_time = 0

start_time = orders[0][1]

for i in range(len(orders)):

if orders[i][1] < start_time:

wait_time = start_time - orders[i][1]

total_time += orders[i][0] + wait_time

start_time += orders[i][0]

elif orders[i][1] == start_time:

total_time += orders[i][0]

```
        start_time += orders[i][0]
    elif orders[i][1] > start_time:
        total_time += orders[i][0]
        start_time = orders[i][0] + orders[i][1]
    return total_time/len(orders)
```

Q2

how many orders do you have to do at a time such that avg wait <= k

avg time a customer have to wait if you have n shoppers (can do n orders at a time)

Q3

K is target average wait time, return num of shopper required to meet that, or 0 if it's not possible

def solution3(orders,target):

第四轮：System Design 设计一个类似于 product catalog

VO 4 轮（加上中间休息总共 5 小时）

1. BQ（就是问问以前做过的项目，楼主之前都是学校里做的 ML research 相关，比较缺乏实操经验，也不知道面试官感觉怎么样）

2. coding 一轮（2 道小问）：

和地里分享的一道，开会题，是一个意思

当时很快就写完了，然后还剩大半小时，就问时间/空间复杂度，然后问了 sorting 的各种 algorithm，后面就提前十几分钟结束了

3. System design

问如果做一个能显示各种 category (food, clothes..) 的页面, 然后点开每个 category 的话还能显示 sub- category, 应该怎么设计 database。感觉这里也答得一般

4. coding 二轮 (3 道小问):

给一个 table 然后各种算东西，总体不难，hashmap 存一下 value 就好了。但是一个有三道题，写出了前两道，到第三道的时候大体的思路都写出来了，但因为这时候已经面了快 5 小时了，脑子感觉糊成了浆糊，debug 效率史前最低（甚至看了很久也没发现有个 bug 只是打错了字）。后来没写完就把思路讲出来了，面试官也说是就对的就结束了。

VO 四轮

两轮 coding + 一轮 system design + 一轮 BQ

coding 其中一轮是 lc，我抽到了 merge intervals 那一套，另一轮是很简单的“按要求写程序”

可能因为 entry level，SD 考的非常简单的 case

BQ 深挖简历和过去经历，更像是聊天

感觉总的来说还可以 recruiter 都挺亲切

最后的 bar raise / value 面面的很艰难，对面因为 covid 换成了一个印度 director，感觉很难 impress...

full stack

- toast notification

- debug

system design

shopping experience for customer

- retailer app check 库存

{商店, 商品, 薯量}

- customer app 架入 商品

{商店, 商品, 薯量, 订单}

- 跑腿小哥 checkout

{商店, 商品, 薯量, 订单}

onsite:

1. 密码 连续三问

2. 找相似 pattern, 连续两问

3. 设计个仓储系统

4. 行为

VO Coding 1: 实现在字符串中找 target 字符串那道题，第二问是如果有 wildcard (*)怎么办

2. Decode password, 详情见 <https://www.1point3acres.com/bbs/thread-742300-1-1.html>

3. 验证 payment 那道 <https://www.1point3acres.com/bbs/thread-777511-1-1.html>

4. Bar Raiser 问了很多怎么跟 cross functioner 尤其是 PM, designer 合作; 怎么影响 management 的决定之类的