

Square 面经

第二轮电面比较偏OOD，设计一个class，实现in mem db，是他们组做payment功能时碰到的问题，从最简单功能开始设计，之后不断follow up增加功能。

onsite 3轮coding + sys design + 之前项目介绍 + manager

coding都是现场上机做，engineer在旁边讨论 + 提follow up。算上follow up题目难度大概算Leetcode比较难的medium，需要边做边compile，边test。需要好好交流同时控制好时间。

sys design 面试之前hr就会告诉是面online hotel booking system。对比其他公司的design难算比较低的。

project轮白人小哥 + 印度小哥，白人来square之前做过跟我类似方向的project，聊的还可以，印度小哥感觉全程走神。。

最后一轮hiring manager是个年轻白人，公司初期就加入了现在已经做到大组manager，现在主要负责square核心功能的用户交互部分。多问了一些他们组的问题，这manager确实能说会道。

面完几天通知挂了。面的几轮自我感觉都没大问题，推测原因一是可能背景没有非常match，之前没有做过payment或者financial相关的proj，start up很看重enthusiasm和culture，没有跟他们聊到很high那种感觉。也有可能他们家bar比较高？有一两轮coding没写完所有follow up。

1. Assign students to classes/teachers in a school as evenly as possible.

Followup: teachers have preferences on students and classes can be full. How to assign. Some students can be left unassigned if all classes are full.

2. Restaurant loyalty program: three levels (bronze, silver, gold). Customers start at bronze level and can upgrade to silver and gold. Silver needs 2 visits, gold need 3+ visits. Customers can be friends. If one customer upgrades from silver -> gold, his/her friends need to be upgraded by 1 level.

Follow-up: how to make the program flexible to add new levels without having to change existing customers.

. check 1point3acres for more.

3. design a hotel booking system support booking, room searching, etc.

一面是个亚裔小哥。我刷题少，所以原题我没见过。但是感觉不难。是个和2d数组有关的题目。我隐约记得有up/down/left/right之类的操作。用char来表示就是：^ > < v。我当时还傻呵呵的说我的键盘上没有表示向下的字符啊。小哥说没关系的，你就用V来表示就可以了啊。啊哈哈哈哈 好尴尬。跑testcase的时候有bug。小哥不光理解了我的思路，还很有耐心的帮我找出了bug。最后超时了10分钟但总算是跑过了test case。还是很不错的面试体验。我自己觉得不是特别理想，但是仁慈的小哥还是算过了。得知过了一面有点儿得瑟，于是很快的约了二面。

二面就比较悲剧。三姐全程心不在焉，中间还挂断了一次我的电话。五分钟之后又打回来。通话的时候老是听见baby的声音。估计是一边带孩子一边面我的。但是不管怎样，最后failed了还是怪我自己刷题不到位。因为我后来发现这特么是个原题：https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life。这道题题目描述很长，所以上来就比较慌。我大概说了我的思路，三姐没说有毛病。然后我就有点儿忐忑开始code了。然后testcase跑不过。后来我自己发现一开始题意理解的不对。然后开始改。改完了还是有bug。但是三姐只是说我的代码看起来好像是对的。她也不知道为什么testcase不过。哎。。。最后当然是failed了二面

利口 原创 雷吧"(gaopin), 这题本身要与出来个定特别难, 我的方法应该和利口DISCUSS里面的差不多。但是! 面我的是一个马内急, 写完之后先是一直问我为什么这行代码这么写, 那行代码这么写, 这题tricky的地方在哪里等等。最后他盯着一个for loop (就是最后format string的时候, 我们会算avg space和remain space来format 结果), 为什么要for loop一个个的word, 然后算space, 问我能不能换个思路, 说了半天我都没太懂他的思路。最后他直接把他的代码写出来, 他是for loop space, 然后把space加到对应的word上。哎, 估计是挂了的, 求米。

Input

(a, b): event a happened before event b-google 1point3acres

(b, c)

(c, d)

(d, e)

(a, d).1point3acres网

(x, y)

Question:. 1point3acres

Given some pair (i, j), did event i happen before event j?

(a, b) => yes, a happened before b

(b, a) => no, b did not happen before a

(a, c) => yes

让写一个class实现key value storage, 里面有store, retrieve, remove, getsize的功能。

扩展开让写另一个class实现key value storage with time.要求保持之前的class里method的interface.

例子:

time 0: store('a', 100)

time 10: store('a', 200)

time 20: store('a', 150)

关键在于retrieve:

retrieve 返回最近一次存的value; 如果retrieve的时间之前一个存进去的信息被remove 的话返回 'key is not found'.

要用一个dictionary of dictionary of list;

做得磕磕巴巴目测要挂。。。。

电面

Edit distance. 1point3acres

Brute force 和 DP秒掉, BUG free,给了高评价

On site

1.5yrs印度小哥, 2sum, 3 sum , n sum

小哥的计划是reuse 2sum (用了hashmap记录访问过的数字) 来写3sum还有nsum, 我提出这样空间上不甚理想, 每个元素都要create一个map (或者clear) 存(i - 1)个元素,

所以implemented了新的sorted + 2 pointers的版本,复杂度 $O(n \log n + n^2)$. bugfree

小哥想了半天跟不上思路, 问了我10分钟为什么变成 $O(n^2)$ 。然后再用递归秒杀的nsum (讨论太久没有跑结果)

也就是这轮让我感觉到水平不行, 明显这个面试官临时抱了个佛脚挑了一道简单的题以及看了基本解法, 但是并没有研究透

Html place holder parser

美国小哥，啥都没准备，上来就直接开项目搞代码

给一个json file还有html file(是真的file, 在corp net上面的),

1.写一个util来serialize这个json object

2.再给一个html file, 把里面每行tag中间的{{key}}替换成json文件里面的值. 跟市面上大部分的前端框架功能相似

基本上就是遇到{{就开辟一个新窗口把中间的内容替换。

-baidu 1point3acres

PS: 小哥自己也是什么都没准备, 让我花了30多分钟上网找java 怎么parse json, 我用过gson的libaray所以一开始就用这个, 然后要加到dependency到pom里面

我没有用过maven所以改了pom也找不到那个库。他自己也不知道该怎么用maven, 最后我自己确定不用gson用org.json然后手动下载库加到project path里面去

最后秒杀了替换的代码, 一跑多了个'}', 立马改了一下游标, 完整的跑出来了

project deep dive, 有个shadow但是主面试官是给了高评价的

design hotel booking

日本人, 做marketing相关的系统的, 口音非常差, 英语非常难懂。

我面的是backend, 一直问我界面该怎么设计, 磨了20分钟我打断他, 跟他确定我是面的backend eng

后面就转为问我数据库schema, 讨论一大轮requests的格式。

其实事后我个人觉得能理解他的出发点, 先讨论要显示什么数据, 然后设计requests, 之后根据requests设计schema

但是全程我感觉他并没有把这个意图表达清楚, 而且他自己也并没有一个完整的计划要考察我什么知识点, 基本上就是指一个地方说一说, 我也感觉不到他有任何有深度的见解

HM 聊聊产品就过了

个人感觉: 如果你面试遇到的题很简单, 那么大约要么HM觉得你背景很适合很想要你, 交代了下面不要为难你, 要么面试官的水平就是这样, 再难的理解不了

自己要甄别。如果是后者在有选择的情况下避免吧, 对自己的职业生涯负责

另外: 感觉Square上市之后内部氛围变化不小, LZ曾经在IPO前面过, 当时一片欣欣向荣, 大家对自己的产品都是有很雄伟的计划, 积极向上

这次感觉到明显面试官士气不高, 办公室依然很华丽, 但是感觉面试官对自己的产品并没有激情, 并没有更深一层的思考

也许就是公司成长的阵痛吧, 混日子的人多了, 况且cash盈利很艰难, point of sales是主业但是竞争也很激烈, 也在尝试不同的附加服务CRM HRM loan什么的能不能走出去是个问题

Design k means

第二轮电面 设计模拟router之间通信 send message

去办公室面的两轮结伴编程一轮过去经验讨论 两道题 一道是验证数独 一道是找一串字符串的所有最短公共前缀

第一次出题是spread sheet editor, 功能是模仿excel的function功能, 比如单元格\$1的值是1, \$2的值是2, \$3的值是 "1*2" 各自输出合适的值

从1point 3acres bbs

```
String eval(string input){
if input contains function:
    calculate input.
if input is a value
    return input
}
```

第二次，出题是DFS搜索字符矩阵，输出edge，例如AB, BC, AD

A--B---C

| |
| |

D E---F

|
|

G

// The 2-player game of Drawdown is played with a board made up of N groups of stones. . From 1point 3acres
bbs

// At the start of each game a board and a list of moves are provided. Each move is a list of N integers, which represent the number of stones the move adds/removes from the board at each index. All moves result in a net reduction of stones on the board.

// Moves can be re-used, but a move can no longer be performed if doing so would reduce the number of stones in any group below 0. After no more moves can be completed, player 1 wins if there are more stones at the first index than at the last index. Otherwise, player 2 wins.

// Example: Let's say the game begins with a board of [6, 4, 2, 4] and these are the available moves provided:

// 1. [-2, -2, 1, 0]

// 2. [-4, -4, 0 ,0]

// 3. [0, 0, -2, -2]

// Player One: 3, Player Two: 2

// move 1, move 1, move 3, move 3

// Initial board: [6, 4, 2, 4]

// Player 1 decides to perform move 1. New board: [4, 2, 3, 4]

// Player 2 can perform move 1 or move 3. They decide to perform move 1. New board: [2, 0, 4, 4]

// Player 1 has to perform move 3. New board: [2, 0, 2, 2]

// Player 2 has to perform move 3. New board: [2, 0, 0, 0]

// The game is now over and player 1 is the winner.

```
public static void main(String[] args) {
    int[] board = new int[]{6, 4, 2, 4};
    int[][] moves = new int[][]{{-2, -2, 1, 0}, {-4, -4, 0 ,0}, {0, 0, -2, -2}};

    stoneGame(board, moves);
    System.out.println("player 1: "+ ans[0] + " player 2: "+ ans[1]);

}
```

```

static int[] ans = new int[]{0,0};
public static int[] stoneGame(int[] board, int[][] moves){
    if(board.length!=moves[0].length){    // in case of any invalid input

        return ans;

    }
    helper(board, moves);
    return ans;
}

```

```

public static void helper(int[] board, int[][] moves){
    // base case, if the board cannot perform any move

    int falseCount = 0;
    for(int[] move: moves){
        if(!check(board, move)){
            falseCount++;
        }else{
            int[] newBoard = doMove(board, move);
            debug(newBoard);
            helper(newBoard, moves);
        }
    }
    if(falseCount==moves.length){
        if(board[0] > board[board.length-1]){
            ans[0]++;    // 0 means player 1

            return;
        }else{
            ans[1]++;
            return;    // 1 means player 2
        }
    }
}

```

```

public static int[] doMove(int[] board, int[] move){
    int[] ret = board.clone();
    if(board.length!=move.length){
        return null;

    }
    for(int i = 0 ; i < ret.length; i++){
        ret[i] += move[i];
    }
    return ret;
}

```

```

public static boolean check(int[] board, int[] move){

```

```

public static boolean check(int[] board, int[] move){
    for(int i = 0 ; i < board.length; i++){
        if(board[i]+move[i]<0){
            return false;
        }
    }
    return true;
}

public static void debug(int[] board){
    for(int i : board){
        System.out.print(i+ " ");
    }
    System.out.println();
}
}

```

第一轮：

上来随便聊了一下简历，然后直接刷题网105，楼主好久没面试有些紧张写得稍微有些慢， anyway做出来之后他又说怎么测对不对，总而言之就是引导楼主再写个前序和中序遍历拿来测试。再写完又说喜欢 recursive，再用循环来一遍。然后就。。没有然后了，随便问了一下问题就草草结束了。

第二轮：

上来小哥非常客气，告诉我不在乎时间空间复杂度只要code简单易懂能scalable就行。于是楼主所有对数组的遍历全部直接forloop。。绝不考虑二分。

题是要构造一个node object，“a->b; b->c; c->d”这种input能够得到b的parent是a， c的parent是b， etc 非常简单。

第二问是说在node里面再加上一个value，这个value是一个boolean，他被set的规则是他的parent如果有true他是false，else 他也是false；理解了之后就是一个拓扑排序搞定，然后随便聊了聊。

题很简单了

输入一字符串 “a->b,b->c,b->d”

是parent->child关系

要求输出 child: parent1, parent2。。关系的list

follow up 是node 有个value是true or false， 如果没有parent就是false， 如果有至少一个parent的value是false那就是true；

要求在之前输出结果的基础上加上这个node 的value

刚面完的**方块**公司电面， 1h， 三姐，大概有30%说的话都听不清，全程我都在ha? can u repeat?

题目很简单：给一个list of string和一个string，判断能不能从list拿出俩个来组成这个string，在这个list里有可能有重复的string出现，比如["Apple", "pie", "Apple"] 那么 "Applepie" 和 "AppleApple"都应该返回true，list里的string每个只能用一次。

我就写了个map存了list，value放了count，然后iterate over string的index，切两半判断一下左右俩能不能行map里有没有，要处理一下左右一样的情况。

我大概没听懂面试官的意思，问了好几遍，面试官不耐烦了，说：你听懂了没？

找大概说了下思路三姐就问复杂度了，问了算不算substring的cost说要算上。

需要自己写test case，写完跑了发现几个bug还有error修了一下，感觉挺不好的。。

搞半天终于写完跑过 但是已经过去45分钟啦，三姐又加了个follow up，说俩个变k个，需要exact k个，我就用了backtracking做了一下= = 又写了点小错，然后跑过了，已经到1h，随便问了俩问题就结束了。

不知道这个k个还有没有更优做法..跟word break也不太一样，如果有大神知道优化做法求告知呀。

sq一直对代码要求比较高，感觉题目不难最后踩点写完跑过，但是疙里疙瘩代码也不够整洁的，也是稳跪无疑了

求！大！米！

店面题很多细节忘记了 算法是给你的 只要能听懂 能实现就行 就是数组的基本操作

现场面试

第一轮 给一段文本 计算ngram ($n = 2$)

<https://en.wikipedia.org/wiki/N-gram>

公式都是给你 照着实现就行

第二轮 给一堆文本 计算 tf-idf

<https://en.wikipedia.org/wiki/Tf%E2%80%93idf>

公式也都给你了 照着实现就行

第三轮 给一堆点 算convex hull

https://en.wikipedia.org/wiki/Convex_hull

我直接上的 $O(n^2)$ 的算法实现 只大概描述了一下 $O(n)$ 的算法 面试官表示可以

第四轮 设计一个旅馆预定系统

基本需求：旅馆查询，预定，结账，收到确认

扩展需求：

1. 定价怎么工作
2. 结账挂了怎么办

第五轮 经验面 讲一个过去的项目 问得比较细

第六轮 行为面 为什么想换工作 成功的经历 失败的经历等等

面试的特点是要求能运行的代码 测试用例都要过 并不太纠结于算法复杂度

这点还是比较赞的 对工作较长时间刷不动题的朋友比较友好

第一轮coding，基本就是实现trie数据结构，有一些api要实现，比如返回所有符合prefix的词，删除一个词之类的。
第二轮实现简单的画板程序，比如画布是二维矩阵，实现drawLine（可以是斜着的），然后实现undo操作之类的。
第三轮比较简单 top k element。
经验轮不说了。

系统设计是design hotel [booking](#) system

面了有一段时间了，发个面经，回馈地里，顺便也求一发大米。
背景：（太长了，当时读题就读了好几分钟）

The 2-player game of Drawdown is played with a board made up of N groups of stones. The stones at index 0 belong to player 1, the stones at index N - 1 belong to player 2, and the stones at indices [1..N-2] have no owner.

At the start of each game a board and a list of k moves are provided. Each move is a list of N integers, which represent the number of stones the move adds/removes from the board at each index. All moves reduce the total number of stones on the board. Moves can be re-used, but a move can no longer be performed if doing so would reduce the number of stones in any group below 0.

After no more moves can be completed, the player with more of their own stones on the board is the winner. In the case of a tie, player 2 wins.

Example: Let's say the game begins with a board of [6, 4, 2, 4] and these are the available moves provided:

1. [-2, -2, 1, 0]
2. [-4, -4, 0, 0]
3. [0, 0, -2, -2]

Player One: 3, Player Two: 2

move 1, move 1, move 3, move 3: p1 victory

Initial board: [6, 4, 2, 4]

Player 1 decides to perform move 1. New board: [4, 2, 3, 4]

Player 2 can perform move 1 or move 3. They decide to perform move 1. New board: [2, 0, 4, 4]

Player 1 performs move 3 (which is the only move available). New board: [2, 0, 2, 2]

Player 2 is now forced to perform move 3. New board: [2, 0, 0, 0]

The game is now over and player 1 is the winner.

. From 1point 3acres bbs

题目；

1. 给定一个board, 判断是不是player one win.
2. 给定board 和List<int[]> moves, enumerate all the possible board moves and then print out the number of times Player 1 wins and Player 2 wins

题目：入参是一个word和一个正确单词列表，返回这个word是否正确

Phase 1: 写完方法，再写unit test，并测试通过

Phase 2: 加一条约束：如果word的拼写正确，但是大小写有问题，返回正确拼写

Phase 3: 加一条约束：如果word中的元音字母aeiou有错，返回正确拼写

phone

directed graph, parents -> children,

1. counting number of children for each node
 2. given rules, calculate boolean values for each node (simply solved by revised DFS)
- followup: what if there is loops in the graph :)

onsite

1. coding, given a list of string representing patents and their authors respectively, sorry, i forget the rest of the question, but it could be found in 1point3acres :(

2. randomly pick up a url of a maven pom, e.g [http://central.maven.org/maven2/ ... 19.0/guava-19.0.pom](http://central.maven.org/maven2/.../19.0/guava-19.0.pom)

reading xml from the link, parse xml, print all dependencies of this library in a given format.

i spending too much time reading documents of xml api though, i would recommend that you may learn a little about how to parse xml in a DOM way

3. given lists of dependency, find out if there is a conflict or not. (topological sort)
4. projects deep dive
5. hotel booking system :)
6. chatting with hiring manager

LC #

Class

_____ |

128

LongestConsecutiveSequence

_____ |

146

LRUCache

题目:入参是一个 word 和一个正确单词列表，返回这个 word 是否正确

Phase 1: 写完方法，再写 unit test，并测试通过

Phase2: 加一条约束:如果word的拼写正确，但是大小写有问题，返回正确拼写

Phase3: 加一条约束:如果word中的元音字母aeiou有错，返回正确拼写

Ant's move. 大概意思是蚂蚁在棋盘上走会涂色，算几步之后的 status。

Design in memory DB

Game of Stones: <https://www.hackerrank.com/challenges/game-of-stones-1/problem>

588	DesignInMemoryFileSystem
981	TimeBasedKeyValueStore

--	--

282	ExpressionAddOperators
635	LogStorageSystem
289	GameOfLife
355	DesignTwitter
068	TextJustification
621	TaskScheduler https://www.quora.com/How-do-I-design-a-job-scheduler
127	WordLadder
271	EncodeAndDecodeStrings
573	SquirrelSimulation
139	WordBreak

699	FallingSquares
969	PancakeSorting
015	ThreeSum
098	ValidateBinarySearchTree
013	RomanToInteger
057	InsertInterval
079	WordSearch
017	LetterCombinations
210	CourseScheduleI
415	AddStrings
211	AddAndSearchWordDSDesign

091	DecodeWays	
167	CountOfRangeSum	

465	OptimalAccount Balancing	
288	ExpressionAddO perators	
284	PeekingIterator	
072	EditDistance	