

第一轮 bar raiser, 一堆 behavior, 把自己现在做的项目讲清楚, 怎么合作, 怎么求同存异; 题目是一道简单的 binary search

第二轮 design, 设计一个类似 ebay 的拍卖竞价系统, 着重讨论了拍卖最后十几秒有大量的 bid 怎么 scale; 在我面过的公司里 Uber 对于 design 的要求不算高, 基本点讲到就可以了

第三轮 problem solving, 纯聊天, 聊一个系统怎么 debug, 怎么 identify potential problem, 感觉都是面试官自己想的题

第四轮 coding, minesweeper 面经; 最后算法都写完了让把之前写的 methods 汇总成一个真正的 game, 问怎么 design; 这里是个坑, 我还以为是继续问算法, 但其实他想要的是 oo design

第五轮 hm, 大量 behavior, 问一个图片上传系统可能会有什么 vulnerability; 其实是考察你有没有了解 security, 我一窍不通但其实问题不大

### LC 133

```
Node* cloneGraph(Node* node) {
    unordered_map<Node*, Node*> copies;
    copies[node] = new Node(node->val, {});
    queue<Node*> q; q.push(node);

    while(!q.empty()){
        Node *cur = q.front(); q.pop();
        for(auto &neigh: cur->neighbors) {
            if(copies.find(neigh) == copies.end()) {
                copies[neigh] = new Node(neigh->val, {});
                q.push(neigh);
            }
            copies[cur->neighbors.push_back(copies[neigh]);
        }
    }
    return copies[node];
}
```

店面: 二分查找的某个变种, 没有见过原题

现场:

设计是自动补全系统

代码是

二叉树的序列化反序列化,

base -2 的数

还有一个题目有点忘记了, 但是要用到 treeset.ceiling/floor 那些函数

大概两个月前面的

第一题: 国人大哥, 让设计一个 heat map, 就是比如有大型体育比赛或者活动, 如何设计一个 heatmap 的系统 让司机看地图可以看到这个 heat map。司机也可以 zoom in 这个 heat map。以及如何提高 performance。后续又问的 cache 都有哪几种 (cache through vs cache aside), 各有啥优劣。又问了常用的 cache method (LRU vs LFU)。然后让简单的说了说 LRU 是怎么实现的。还问了下 DB hashing 是咋实现的, 简单的说了说如何实现 consistent hashing。感觉国人大哥很不错

第二题：拜仁大哥，给了一个非常简单的系统：UI, web server, cache, DB。然后说 web server 如果只有 1 个 thread 来 call db, db 的 schema 是啥（具体的 use case 我忘了，但是非常非常简单）。还问了下 distributed cache 都有哪几种。然后如果有两个 thread 来 call DB 会有 race condition, 然后就讨论如何解决 race condition. 提到的解决方法有加 global lock, 加 timestamp, row level 的 lock。后面讨论如何实现 row level 的 lock。具体的我都记不清楚来

第三题：欧洲大哥，给我出的编程题目 就是类似于 parse 一个 json string 之类的 这道题我写的不好，corner case 太多

第四题：国人小妹妹，特别感谢啊。黎箝 295

前段时间看到过他们自己写的 heat map <https://eng.uber.com/h3/>

面试官是一个台湾大叔。题目是一道地里出现过的类似的题.给一个 stack 的 interface, 有三个 method, 分别是 push, pop, size 写一个类来实现它。

然后问怎么样能随时得到栈里的最小值。

基本上就是 [cc150](#) 原题。

注意这里是实现一个栈, 所以不口

xhjin99, 本帖隐藏的内容需要积分高于 188 才可浏览, 您当前积分为 178。

[查看如何攒积分 Click here for more info.](#)

能直接用 java 里的栈作为数据结构。

最近刚结束 Uber 电面，烤点面筋，造福嘴馋的朋友。纯干货 - - 求大米~~

Given a matrix, with values 'W', 'G', '0', 'W' means wall, 'G' means Guard, 0 means the way you can go.

Find the distance of all

542, 286, 317

面试官迟到了 5 分钟。然后说 recruiter 找不到了。邮件里没有 codepad 的链接。然后乱七八糟耽误了一些时间。问了下 why uber 然后就直切主题 coding 了。。。题目感觉有点怪？不是典型的利口。2 个 input, 一个是长长的一句话，一个是一行的长度 limit。结果就是需要你把长长的一句话分成好几行输出，保证每一行都不超过 limit。单词当然是不能被割断的。

e [□□此部分内容已被隐藏，请点击右下角前往一亩三分地论坛查看此贴] 了，也没有问时间复杂度什么的。感觉和自己预想的不太一样。题目比预想简单很多然而自己基本功太差感觉很可惜惹。

面试官声音听上去就很年轻。搜了一下发现是个小鲜肉。

史上最墨迹的电面经历。黑车 ATG perception 组。

[hide=188]

先走了一轮电面，就是给一句话，给个长度限制，然后输出不同到每一行。地里面经和里扣上都出现过的。有 follow up, 比如一个单词太长，如何如何，如何在最后输出行数，总行数。

[Seattle]

给定多个[start\_time, end\_time] 每个 interval 是乘客需要上车下车的时间，问最少需要几个司机。扫描线秒掉，1 个小时的面试，10 分钟自我介绍，15 分钟做题，5 分钟问问题，然后黑车小哥没有任何 follow up questions 面试半个小时愉快的结束了

## [Marketing]

电面：

二叉树给定一个 target sum，输出所有从任意节点往下的路径和为 target sum 的路线

现场：

1. bar raiser : 各种 bq
2. 算法：trie 的实现，search 的时候有 "." 表示可以匹配任意字符
3. 算法：merge interval 变种
4. hm : project deep dive
5. 系统设计：设计 price track notification，用户可以 track 一些商品的价格 有大幅度变动的时候用户能收到 email 通知

第一轮:BQ, 最后问了个电梯调度问题，也不 OOD, 就是讨论了一下电梯调度哪种算法最优

午餐: HM 带着吃了午餐

第二轮: HM 问了些 BQ 问题

第三轮: Coding, 两个 interval array 的 intersection 和 union, 面试官让我拿出自己的电脑，现场给了个在线 coding 的链接让我电脑上写

第四轮: Coding, LC 原题 琪气伞, 面试官一开始问的最少几步能走出来, how many steps, 所以我函数定义为返回值是 int 类型(如果这一开始就不是面试官想要的结果他也不指出来也不纠正, 呵呵) 用 BFS 写完，面试官等我全写完以后又改口说这个 steps 意思是快照，要记录最短路径走过的每一步的棋盘的样子，我就只能在写了满黑板的代码里边改

第五轮:BR, 问 design 一个 battleship game 的 OOD

## [SF]

Uber ATG 三番无人车 perception 组 computer vision engineer 面经。职位是十月初在校园招聘会上投的，经过一轮 oncampus 面试和 5 轮 onsite 面试，时间有点久远了。

oncampus :

leetcode 原题，trapping raining water。用的 dp 做。follow up : 怎么并行化处理很长的数据（随便扯了一通），show 了下做的 project 的 demo。

onsite :

面的非常 cv，应该是和具体的职位有关。

第一轮，天竺 hiring manager，问简历和 project，问 publication 的具体 idea 和实现。还问了很多 object detection 相关的知识（比如 Faster R-CNN 和 YOLO 的区别），检测小目标的技巧，训练数据不够怎么解决。以及双目视觉相关的知识，怎么从 stereo camera 还原物体的深度信息等等。最后问想来之后做 research 还是工程。

第二轮，白人小哥，design 面试。一上来问我 battleship 游戏有没有玩过，答曰不会。然后上了 leetcode 原题贪吃蛇，让白板实现（也是非常幸运了）。最后还剩下很多时间，问了怎分布式 neural network 训练的实现，答曰 parameter server。然后白板分析了瓶颈在哪儿，最后吹捧了下他们的 horovod 框架的优点。结束。

第三轮，和匹兹堡 office 的远程面试，又是 cv 和无人车相关的。问简历和 cvpr paper，极其细  
xhjin99，本帖隐藏的内容需要积分高于 188 才可浏览，您当前积分为 178。  
[查看如何攒积分 Click here for more info.](#)

#8204;结束。

中饭，一般般，尬聊了好久。

第五轮，匹兹堡 office 远程 behavior 面试，面试官很 nice，于是就坦诚的聊了 30 分钟，结束。

第二周通知 pass，感觉 ATG 的 office 和车真高大上，面的问题很 cv，也很细致，体验不错。缺点就是 package 实在太少，还在纠结去不去。

1. BQ

2. LC 二把变种，needle 可以 permutation。

map 记录 needle 中的每个字符的次数，然后 two pointers sliding window 扫一遍

3. LC 啧啧思，follow up 一些多线程的问题，然后改成 k streams

k streams 的思路可以参考 merge k sorted list

更正一下第三轮：3. 并不是 LC 一一一斯。知识单纯的 merge 2 sorted stream to one。然后 follow up 还是原先的。

没有仔细看 LC 的题，有误解的请谅解哈。

4. 系统设计，设计一个 uber 中输地址的推荐。

有点像 typeahead，但是偏重点会向按地区，自己的记录这方面，比较 open。

5. BQ

最后挂了，hr 不给具体的原因，但说 technical skills 反馈不错，但是最后还是没录，然后我就蜜汁悲剧了。

个人感觉最后一轮可能聊得太放松？不过可以感觉到系统设计这次是有些进步。

想吐槽一下这次的昂赛 recruiter，非常不友好，打电话说话很含糊，email 回复从不超过两行，安排面试也都拖到最后一天。

店面：二分查找的某个变种，没有见过原题

现场：

设计是自动补全系统

代码是

二叉树的序列化反序列化，

base -2 的数

还有一个题目有点忘记了，但是要用到 treeset.ceiling/floor 那些函数

共五轮，两轮 coding，两轮设计，最后一轮 BQ。

1. Find local minimal in given array, 要求现场运行并通过 test case

2. Design authentication system, 实现添加用户，验证用户，获取用户 profile 和用户登出的功能

3. Minesweep, number of islands 变形

4. Merge SQL rows, 用并查集方法

5. BQ, how to build consensus when different strong opinions 等等。

[/size]

面试结果未知，如果挂了应该是最后一轮，问了很多 BQ，头都晕了。  
路过赏点大米，小弟看看面经。

[Seattle]

电面

美国小哥有很多 networking 背景，上来就先问了一些 networking 方面的，how traceroute works, UDP/TCP difference and use case . 聊简历项目

Coding 题比较简单，具体忘记了。。。大概是给你一个有规则（pattern-> 要替代的字符串）的语言，根据

申请的 fullstack 职位，6 月底电面，一道蠢口伊儿伊，一道给四个坐标，判断能不能组成正方形，两天后通知过了，本来申请的是帕洛阿托的职位，独立日后通知职位被 fill 了，被转到 freight onsite 在城里的 office 一共四轮，两轮算法，一轮 design, 一轮 manager, 这次没有关于前端的 codin [□□此部分内容已被隐藏，请点击右下角前往一亩三分地论坛查看此贴] hard 还是得刷，不能心存侥幸，还有要在白板上或者纸上练习。coding round 经常发现白板写不下，会打扰思路 and 心情。

[白人小哥]

今天下午面的 题看着很简单 结果两个都没跑通所有 case 很是伤心 不知道为啥 脑子短路了 还有个插曲就是 hackerrank 不知道出啥问题了 不管选哪种语言代码都是 c 连面试官都有点崩

Ex

```
Input: String "John is planning to move to Seattle after getting a new job there.", int maxLimit 20
Output: String[]
John is planning to
move to Seattle
after getting a new
job there.
```

用了两种方法 都没成功 挂完电话 第二种方法也改出来了 好无奈。。。 之前做过利口留吧 很相似 不过利口的更复杂点 要考虑居中的情况

LC282

LC780

因为都是正整数，(x2, y2)必定一大一小，用大的减去小的可以得到上一步的坐标，这样一直减下去，应该是  $O(\max(m,n))$

**补充内容 (2019-7-24 08:33):**

貌似没规定  $x1$  必须等于  $y1$ ，这样的话解法就不对了。。

[Palo Alto]

第一轮 design 很胖的美国大叔，很 nice，设计一个 uber 车数量查询的系统，输入时间，地点显示当时当地车的数量

午饭，就是第一轮大叔陪吃，相聊甚欢

第二轮 两个烙印，image 二叉树 表示，秒，聊 kafka 的一些实现细节和在项目中的运用，答的不错

LC427

第三轮 东欧小哥，电梯调度算法，没准备，硬撸，挂在这轮是一个电梯，让你设计一个算法，接人送人，实现电梯的基本功能啊。一开始是一台电梯，估计 follow up 会多台电梯

第四轮 忘了，不好意思

第五轮 BQ+Experience 印度小哥，相聊甚欢，最后让选 做一道 coding 还是设计，断然选设计。。别问我为啥，因为设计简单，关键点知道，会吹就行。。

Uber 某组 7 月初 onsite

第一轮 Algorithm：给一串数组代表每天接到的 task 能获得多少 value，然后每个 task 都是固定长度 2 天，如果 took 了第一天那么第二天就不能 took 了，问随意选取 task 最多能获得多少 value，比如说 **[1, 4, 3, 2, 5]** 可以选择  $1+3+5=9$  是可以获得的最大 value，用动态规划来解的。Follow up：如过每个 task 的时间不一样，给你的是一个 tuple 的 array，代表每个 task 的 value 和 take 的时间该如何求最大 value。第二问没有让写代码，就只是说了一下用 dfs + memorial search 的方法来解

第二轮 System Design：设计一个用户实时更新当前位置给朋友的系统，可以有多个朋友 subscribe

第三轮 Manager behavior 以及聊之前的 project

第四轮 Coding：word break I 和 word break II

1. Hiring manager 面，问了个 permutation, 最后问了一道 reverse linked list

5. Design - tiny url

heatmap 的思路

可以看看这个视频：<https://www.youtube.com/watch?v=umWABit-wbk>。其实就是让 driver 把 location 数据每隔几秒上传一次，然后用 geohash index。

[Palo Alto]

第一轮三弟 经典面经题 intersect/union two lists of intervals

第二轮三叔 design Amazon 商品详情页下的 people also brought these items

Grokking the System Design Interview 上有差不多类似的吧

问好需求 DAU 这些 设计 API 然后设计数据库 aggregate 数据 再来个缓存存下数据

这些点都主动去 drive 会给面试官好的印象

第三轮三妹 打印所有的小于某个数的 Jumping Number. Jumping Number 就是相邻位数之间差绝对值为 1

比如打印所有不小于 105 的 Jumping Number 就是 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 21, 23, 32, 34, 43, 45, 54, 56, 65, 67, 76, 78, 87, 89, 98, 101

其实就是 BFS/DFS 遍历二叉树. Follow up 是打印某个区间的 Jumping Numbers 说了 bfs/dfs 写了 dfs

先说从 start number bfs 开始找下一个

然后又说可以 cache 住一堆 然后 binary search

第四轮三哥 Bar raiser 过简历上的 projects + BQ

第五轮三叔 Hiring manager 详细地聊了一个 project + BQ

## [ATG, self driving platform, SF]

两轮 behaviour: 最经典问题, 你觉得 safety 对你来说意味着什么。。。我???

三轮代码: anagram in string 必须  $O(n)$ , 我开始有 bug 后来面试官给了个 test case 自己把 bug 改了, 一个奇迹 easy 的 leetcode, 然后重点来了, 电话号码变单词的那道 leetcode, 给定一个数字组合 (23: ad,ae,af,ba,be,bf,cd,ce,cf) 问对应的单词是不是合法英文单词, 有哪些合法英文单词 (example: collection 合法单词, endjws: 这种随机的字母排列不是英文单词) 且当数字组合长度较大时怎么办。。。。。(他提示不能搞字母 permutation, 最后我抖机灵搞了一个  $O(1)$  的算法, 然后感觉他对我也放弃了。。。。。)

最难的是最后一题, 造 n-ary tree, 给一系列 tuple, 无序的, 不仅 ancestor 和直系的 children 是一对 tuple, 且 ancestor 和任何非直系的 children 也是一对 tuple 打比方说 1-2 1-3 2-4 这样一个树 那么输入就是[(1,1)(1,2)(1,3)(2,2)

我写了下第三题的 topological sort Python 写法, 本质上是删除非 direct 的 edge, 求加米

```
# 1
# / \
# 2 3
# / \ \
# 4 5 6
# /
# 7
import collections
def tuples_to_tree(tuples):
    # build graph
    g = collections.defaultdict(set)
    ind = collections.defaultdict(int) # indegrees
    for x,y in tuples:
        if x!=y:
            g[x].add(y)
            ind[y] += 1
        if x not in ind:
            ind[x] = 0
    # delete edges that are not direct parent-to-children
    ind_0_list = collections.deque()
    for k,v in ind.items():
        if v==0:
            ind_0_list.append(k)
    while ind_0_list:
        node = ind_0_list.popleft()
        ind[node] = -1 # will not consider this node any more
        for nextnode in list(g[node]):
            if ind[nextnode]>1: # has other sources, not a direct parent-to-children edge
                g[node].remove(nextnode)
                ind[nextnode] -= 1
                continue
            elif ind[nextnode]==1: # direct parent-children edge
                ind_0_list.append(nextnode)
    print(g)
```

```
tuples = [(1,2),(1,3),(1,4),(1,5),(1,6),(1,7),(2,4),(2,5),(2,7),(3,6),(4,7)]
```

tuples\_to\_tree(tuples)

# output: {1: {2, 3}, 2: {4, 5}, 3: {6}, 4: {7}, 5: set(), 6: set(), 7: set()})

[another input] 最后一题想到一种算法，对每个 node 记录有多少 parents 和有多少 children，都用 set 来记录 然后那些没有 children 的肯定是 leaf node 了，把他们放到一个 queue 里面去，遍历这个 queue，每一次遍历一个 leaf 就把它从 parents 的 children set 中去掉 如果一个 parent 的 children set 变成 empty 了 说明这个 parent 是 direct parent，把这个 parent 也加到 queue 里面。这样到最后 所有的 node 都在 queue 里面，tree 也做出来了。

runtime 的话因为一次去掉一个 connection，所以是  $O(n)$ ，n 是 tuple 的数量

[another input]

```
NTreeNode *{
    int val;
    vector<NTreeNode *> children;
    NTreeNode(int value){ this->val = value; }
}

NTreeNode *Build(vector<vector<int>> tuples){
    // add all nodes to parent child maps,
    unordered_map<int, unordered_set<int>> parents;
    unordered_map<int, unordered_set<int>> children;
    for(size_t i = 0; i < tuples.size(); i++){
        int p = tuples[i][0], c = tuples[i][1];
        parents[c].emplace(p);
        children[p].emplace(c);
    }
    // find the one without parent
    // Add to queue, the value is NTreeNode *
    queue<NTreeNode*> q;
    NTreeNode *root;
    for(auto it = parents.begin(); it != parents.end(); it++){
        if(it->second.empty()) {
            root = new NTreeNode(it->first);
            q.emplace(root);
            break;
        }
    }

    while(!q.empty()){
        // Whenever pop out a node, reduce all it's children's parent , if
the parent count is 0:
        // Add child to current node's children list, add to queue
        NTreeNode *cur = q.front();
        q.pop();
        for(auto child : children[cur->val]){
            parents[child].erase(cur->val);
            if(parents[child].empty()){
                NTreeNode *new_child = new NTreeNode(child);
                cur->children.emplace_back(new_child);
                q.emplace(new_child);
            }
        }
    }
    return root;
}
```



我面试的时候做的是直接建立一个 **hashmap**, **key** 是数字串, **value** 是该数字串对应的所有合法单词。。。。

这样随意给一个数字组, **O(1)**直接找到所有对应单词。。。。

我个人感觉按照您的方法建立 **TRIE**, 似乎与我建立 **map** 的方式类似, 但是我个人感觉我这个好像搜索的快一些。。。。。

如果我说的不对, 请指正呀, 大神轻拍

你的方法可行, 和 **trie** 是类似的, 读取是 **O(1)**, 时间主要花在 **build** 的过程中; **trie** 的好处是在 **word** 量大的时候, 比较省空间; 如果不让用 **trie**, 我觉得你的方法应该已经挺好了

## [phone]

利口原题吧? 双指针

You are given with an array of 1s and 0s. And you are given with an integer **m**, which signifies number of flips allowed. find the position of zeros which when flipped will produce maximum continuous series of 1s.

```
// # e.g.  
// # input:  
// # arr=[1,1,0,1,1,0,0,1,1,1] m=1  
// # output=[1,1,1,1,1,0,0,1,1,1] position=2  
  
// # arr=[1,1,0,1,1,0,0,1,1,1] m=2  
// # output=[1,1,0,1,1,1,1,1,1,1] position=5,6
```

[phone]是个阿三, 聊了一会简历和 **project**. 代码: 里扣 耳零领 (我用 **DFS** 实现的, 这里他问了给一个 **1000\*1000** 的全为 **1** 的矩阵作为 **input**, 在 **local desktop** 上跑可能会 **fail**, 问为什么? 我完全 **no idea**, 欢迎大家留言讨论)。

follow-up 里扣 留揪撕

## [on-site ATG]

1. **product manager**, 30 分钟纯聊天, **bq**  
午饭, 下午的一个面试官带着去吃的, 不好吃...

2. **hiring manager**, 45 分钟纯聊天, **bq**

3. **sr engineer**, 45 分钟 **zoom** 纯聊天, **bq**

4. 中午带着吃饭的小哥, **coding**

4a. 滑动窗口, [刷题网 76](#)

4b. 三组数字, 第一组其中的一个数字和第二组其中的一个数字相加, 等于第三组的其中一个数字, 找出所有的组合, 每一组的数字可能有重复

谢楼主分享, 我想问下 **4b** 的题意。这题是问 **A** 和 **B** 各取一个, 有多少种组合能加起来等于 **C** 里的一个数字, 并且每种组合不重复是吗? 这题我怎么有点懵, **C** 可以扔进 **hash**, 但是 **A** 和 **B** 的组合除了枚举还有什么更好的办法吗, 好像 **sort** 也不顶什么事。

第六题好想是某网站已而留变种, 不知道楼主是不是 **BFS+ DFS** 做的

**4b** 我也有点懵, 最后 **10** 分钟出的一道题, 分析了一下能用什么方法, 讨论了一下不用额外空间怎么做, 估计重点是在于分析吧

第 6 题我是单纯的 dfs 做的, followup 时间不够了, 就分析了一下大概怎么做, memo 需要注意的事项什么的

5. 系统设计, 设计 facebook

6. 先聊了会简历啥的然后 coding, 有向图, 先找出图中一个点最远能 travel 多远, follow up 是找出图中能 travel 最远的点.

比如, 我有一个图, a->b a->c a->d b->c c->d followup 就是要找 a-b-c-d 这条路径

[on-site, ATG, Pltts]

一道利口斯尔, 一道利口散灵思

[on-site ATG, pits ]

第一轮: hire manager 聊天

第二轮: coding, 大叔自己想的题, 第一道是给定一个矩形区域和一堆矩形 tile, 看这些 tile 是不是完全 cover 矩形区域并且互相没有重叠。第二题是汉诺塔

第三轮: coding, 一个编辑距离

第四轮: bar raiser, 各种 behavior

第五轮: 设计, 股票数据查询分析系统

[on-site, android position]

第一轮本来是设计. 但是面试官说 train delayed 了. 我就自己在面试房间里看了一个多小时的综艺节目. 都笑出声音了. 哈哈哈哈哈. 然后 recruiter 各种安排. 本来说 over the phone. 我心想 design 肯定画画最好啊. 然后就让 recruiter 安排其他轮 可以电话. design 取代那个面试. 最后的是取代了 Hiring manager 的面试 time slot.

第二轮: 2 个面试官. 一个 shadow 写 app. 跟 <https://www.1point3acres.com/bbs ... android-526544.html> 这个跟这个哥们一样的. 但是我 debug 自己搞的 bug (傻死了. 复制黏贴. 改了 variable 名字却没改 value. 面试官们都没看出来. 最后自己改了.). debug 了太久了. 导致剩下没多少时间了. 就没怎么写点城市打开城市的 weather 页面. 就说了下如何写这个 feature. 然后如何改现在的 code base. 不过面试官人真的蛮好的. 笑嘻嘻的.

第三轮: 吃饭. 但是由于 hiring manager 的面试 time slot 被 design 拿走了. 本来这轮就是 Hiring manager 陪我吃饭. 然后直接这轮 他就拿来面了. 其实很不推荐这样做. 到头来没吃多少. 都在说话. 下午都在饿. 问了你想要的 team 是怎么样的. 然后你如何帮助 team 形成了那个氛围. 然后问你如何 approach 一个 project. 很迷. 问的有点不是 behavior 问题.

第四轮: 第一轮那位 train 迟到的面试叔叔. 贼壮. 那肌肉真的是吓到我了. 设计 facebook ( 要求 login, facebook timeline, and write post). app 要支持 offline. 还有个要求我忘了. 这轮学到了不少. 回去好好补习补习. 大叔会一直问. 这样会有啥问题. 这样会有啥问题. 你知不知道这个. 你知不知道那个. blablabla.

第五轮: 2 个面试官. 还是一个 shadow. Valid Sudoku. 一听到就知道完了. 因为忘了那个小 block 的 math 了. 想了 2 分钟. 没想出来. 面试官就试着帮我用他的方法. 结果我就被越带越偏了. <https://leetcode.com/problems/valid-sudoku/> anyway.

第六轮: bar raiser. 一个怀着孕的 ios 国人小姐姐吧. 问了些 behaviors. 有没有 conflict 跟别人. 举了个例子跟 teammate. 她就再问有没有 conflict with manager 然后下半部分又是一个设计. 设计 notification system. 面的蛮迷的. 她刚开始问的 facebook timeline 知不知道. 然后我以为她要面 facebook timeline. 结果就说不是. 囧. 需要支持 pagination. 然后没了.. 然后就开始自我发挥了. 需要支持 poll + push notification. 但是没做 push 的东西. 回去补习补习这个

### 【phone, ATG】

invalid parentheses  
stock price profit one

### 【phone, ATG】

打电话的，是一个不是非常热情的国人大哥

一上来也就两句话就开始讲题目，老实讲上次面试还是两年前，不知道现在面试这么简单粗暴了直接写题了。

题目是面经里有的，要用 `array, list` 实现 `hash map`。一个重复造轮子的问题，实现 `push, get, update`

问了一下 `collision` 怎么办 直接提供了 `c++ hash library` 的连接，用 `C++` 的小伙伴可以看一下 `hash` 这个 `library`，以前没接触过，挺有意思的

我是直接把 `hash` 求 10 余数直接放到一个大小为 10 的 `array` 里面了

我过程中除了一些 `C++` 独有的 `memory bug`，大哥也没能帮忙看出来，浪费了很多时间，其实根本不管逻辑的事，唉  
感觉是要挂了

### 【on-site】

10 分钟聊经历和项目

然后做题

第一问 利口 亮白第二问是找最大的岛（返回最大值）给了四组测试数据，需要编译测试通过

第一问一开始有点小问题不过很快改过来了

然后面试官说那你说说第二问思路就可以问问题了

心想面试一共一小时这才 30 分钟就要跳过第二问是要挂么.....

### 【intern】

刚面完的 `uber sde intern`，双方自我介绍后就开始做题。

题目是刷题网站 `word search` 那道题，第一次面试碰到做过的题有点小激动，然而跑出来结果错了。。。当场 `debug` 时间不够也没发现错哪，面试小哥还安慰我没关系 `hhh`

对 `uber` 印象不错，`hr` 回邮件很快，面试小哥人也很好，感觉很喜欢自己&#

### 【phone】

俩印度人面的，上来废话不多说直接扔题目箝匙全，秒掉之后出个新题，给哈西脉浦加个新功能，一个方法同时给里面存的所有 `entry` 改变 `value`

### 【phone, ATG】

编程题是 `LeetCode Subarray Sum Equals K` 基础上，要求函数返回所有满足条件的数列组合。需要调试编译测试通过。

### 【phone, ATG】

`Uber` 电面。给 `BST` 找 `closest node`, `follow up` 找 `k` 个。

国人大哥，人不错。

### 【phone】

准备了很多题，遇到的是经典的 `DFS` 和 `BFS`，也算是运气吧。但根据面试表现，还真不一定能过，希望能再运气一点。

1. 题目描述很简单就是给一个二维矩阵 `image grid`，每个单元里的值为颜色代码，给一个起始点和目标颜色，任务是将起始点所在的区域（相同颜色）改为目标颜色。

代码主体就不用说了, 我认为这里的考点主要是 `coding style`, `input validation`, `avoid revisiting` 的方法以及自己能举出多少 `test cases`。

2. 由于我选择了 `DFS` 来做第一问, `followup` 是如果图片过大, 会怎么样, 要怎么做。

这里的关键词是 `stack overflow`, `iteration` 以及 `BFS`。在面试过程中, 我中间有点跑偏在考虑怎么分区  
域分别做 `DFS` 再把结果合起来, 面试官很快就提示了用迭代的方法将我拉到了 `BFS` 的正轨上。。

实现中有个地方用错变量名导致输出结果不对, 用 `Print` 大法调试了一下有点困。

总结一下, 我可以做的更好的地方在于:

1. `corner case` 并没有想全, 中间遗漏了对于用户输入的检查。
2. 中间他问我会不会 `stack overflow`, 我只说了会, 但量化的说明多少数据规模会超过 `heap space` 就更好了。
3. 对于图的遍历的题目, `DFS` 和 `BFS` 这两种选择最好在一开始都列出来再选择实现哪一个。而我是直接就选择了 `DFS`, 思路上一开始就狭隘了, 导致了第二问的转换并没有很流畅。

### 【phone】

贡献一个黑车的电话面试面经。赞人品, 希望能帮组到大家。

面试的是一个阿拉伯女生, 题目是 `deep copy` 链表 (有 `random pointer`), 写完代码后, 主要的时间用在了写 `test case` 和 `follow up`。她希望一个  $O(1)$  的算法。感觉自己的在 `follow up` 上面卡了好久, 在快结束时候, 在有 `hint` 的情况下, 打出来了。感觉, 整体发挥很一般, 交流也一般。希望和小伙伴们一起共勉。继续加油准备。

### 【phone】

`uber freight` 一个亚裔小哥 - 先问问目前的项目, 然后出了一道 `setTimeout for loop` 改错题目

第二道 脑抽筋 没做出来

### [hide=150]

给了这个 二维数组 求最长的 路径

```
[ ["SFO", "EWR"], ["SJC", "LAX"], ["DFW", "SJC"], ["EWR", "OAK"], ["LAX", "SFO"] ]
```

["

利口 散散尔

第二题似乎是这个 <https://www.geeksforgeeks.org/find-longest-path-directed-acyclic-graph/>

用 `topological sort`, 然后一个一个算从起点的最长距离。起点可能是任意一个没有排在自己前面的点。

感觉面试问 `graph` 的东西有点难啊。

这个还得请教楼主一下, 如果条件很简单, 就是每次都会有一个 `chain`, 每次都会有一个最长的队列, 那么就会简单一些. 只需要找到 出发 和 结束的机场, 然后通过 `hashmap` 连在一起就可以啦. 或者 `union find`

对的应该是这个方法

不需要 `dfs`

追问下, 什么是找出 起点和终点?

->我之前回复的利口的题和这个不是一个题. 不过都可以做一下.

起点可以有多个,也可以只有一个, 终点也是一样. 这要取决于题目的难度. 感觉如果是 FE 面试的话,不会搞那么难.

我假设楼主这题是一个相对容易的设定.即只有一个起点和一个终点. 把起点和终点连接在一起就是答案

给定的 input, 所有的 itinerary

```
["SFO", "EWR"],  
["SJC", "LAX"],  
["DFW", "SJC"],  
["EWR", "OAK"],  
["LAX", "SFO"]
```

for loop 一遍, 生成两个 set

```
startSet = Set(["SFO", "SJC", "DFW", "EWR", "LAX"])  
endSet = Set(["EWR", "LAX", "SJC", "OAK", "SFO"])
```

```
const startAirports = [...startSet].filter((airport) => !endSet.has(airport)) // DFW  
就是找到所有起点机场 not in 终点机场, 就是初始起点 (1 个或多个).
```

### [phone]

反常的面试

白人面试官

各种问过去的 project, 感觉很像 system design 了

然后让写 java 的 connection pool

没 requirement, 自由发挥。。。

工作多年, 面试也很多, 这样的电面还是头一次, 完全 get 不到他想要什么。

### [phone+on-site]

两个月前有 Uber 的 recruiter 联系, 当时还不确定哪个组适合我. 先安排了技术电面, 考的 C++, 题目如下: Given an array of references to nodes in a doubly-linked list, count the number of contiguous blocks of nodes in the DLL (doubly linked list) that exist in the array. Doubly Linked List:

1 <-> 2 <-> 3 <-> 4 <-> 5 <-> 6 <-> 7 <-> 8

Input - [3, 1, 8, 2, 5, 7]

Output: 3

他会给你提供一个节点的模板类和双向链表的模板类

[<https://www.1point3acres.com/bbs/plugin.php?id=attachcenter:goto&aid=MzA2MDMwfDIzZGI4ZjB1fDE1NjYwNTM3MzV8MjIxMzk4fDUzODc4NA%3D%3D>], 然后你去写一函数来实现算法, 写完后会让你自己在 main()函数里写 test case 去测试, 最后测试过程中有小错误他会问你你怎么 debug?

那个类好像不一定不能修改, 我中间主动在里面加了 getter 和 setter 他也没阻止我, 还说他忘写了。。。

最后可能是因为我用了 set 数据结构, 他就问我你的算法的时间和空间复杂度是什么? 又问我怎么优化成线性时间复杂度? 答案就是把 set 简单的换成 unordered\_set, 就是哈希表, 用空间换时间. 附件里有我当时写的代码

后来 recruiter 说电面过了来安排 on site, 但是等了好久也没消息, 然后有个小插曲:  
recruiter 说 perception 组的老板之前出差了, 现在回来了想先打个电话聊聊, 主要就是问问我现在做的是  
什么, 他们组做的是什么。打完后老板说聊得不错, 接下来给你安排个技术电面, 考过了就来 On  
Site, 我说我上个月都考过了呀, 他说呀不好意思他拿错简历了。。。然后马上安排我 On Site。

On Site 先是临组老板来面, 说是为了有一些 unbiased 的评价, 然后就是本组老板面了。  
这两轮主要都是问问我目前做的是什么, 遇见过的最大的 challenge 是什么? 怎么解决的? 以及围绕  
我的问题下的一些探讨。  
本组老板主要以介绍组内工作为主, 人非常赞, 最后十分钟让我给他教一个他可能不会的知识。我被  
这个问题震惊到了, 后来聊了聊经济学原理。。。

下午的三轮就是组员来面算法了。  
算法题应该是 leetcode 其二, DP 可解  
果然! 应该是这个题。大牛啊! 但是这个题不给已有单词的集合所以感觉就不能用图论的思路了。一  
下子就难了许多。没有, 这是 DP 的一种类型题, 其实本质上还是最小子问题。我觉得 bfs 暴利也是  
可解的, 不过反倒麻烦了。

电面的这题感觉用蠢口要儿吧的思路就行。DLL 感觉没啥用, 不知道有没有什么高深的解法。  
onsite 第三轮可以借鉴李口 已而期

#### [on-site, seattle]

吓图 office. 2 轮算法, 一轮 system design, 一轮上机写 app。  
题目比较常规, serialize/deserialize binary tree, lru, 加 bq. system design 设计 battle ship

<https://instant.1point3acres.com/thread/454344>

好像没在常考面经题里看到过

#### [hide=180]

就是按照格式逐月 print 出某一年的日历

example:

```
// January 2018
// Su Mo Tu We Th Fr Sa
// 1 2 3 4 5 6
// 7 8 9 10 11 12 13
// 14 15 16 17 18 19 20
// 21 22 23 24 25 26 27
// 28 29 30 31
// .
// .
// .
// December 2018
// Su Mo Tu We Th Fr Sa
// 1
// 2 3 4 5 6 7 8
// 10 11 12 13 14 15 16
// 17 18 19 20 21 22 23
// 24 25 26 27 28 29 30
// 31
```

格式很重要

[/hide]

[phone] <https://www.1point3acres.com/bbs/thread-531453-1-1.html>

input 是一个 integer array, 要判断能不能通过它来建立一个 BST

The query 【1, 3, 2】 is valid, the root is 1, and 3 is its right child and 2 is the left child of 3

The query 【1 3 4 2】, is not valid. The root is 1 because it is the first value in the list. The second value of 3 must be the right child of 1 because it is greater. Likewise, the third value, 4, must be the right child of 3. For 2 to be the last value in the traversal, it has to be the left child of 4. It is less than the root value 3 above it and is on its right subtree. Return the string NO.

The query 【3 4 5 1 2】, is not valid. The root

xhjin99, 本帖隐藏的内容需要积分高于 188 才可浏览, 您当前积分为 178。

[查看如何攒积分 Click here for more info.](#)

;了 O (NLGN) 的 solution, 最坏情况是 O (N^2), 和 quicksort 一样

follow up question 是给出 O (N) 的 solution, 这个我没有回答上来, 看看大家能不能给我点提示  
创建两个变量 leftBound 和 rightBound, 初始化为 Integer.MIN 和 Integer.MAX, 从第二个元素开始遍历数组, 如发现元素不在两个边界之间即返回 false, 否则更新两个边界中的一个(比如下个元素比之前大, 则更新左边界为之前元素值, 如小于之前元素则更新右边界), 然后到下一个 iteration, 直到数组末尾。如果遍历完数组则返回 true

we could use a stack to store the range of right child and left child. Ans traverse through the tree to see if we could rebuild the BST.

For example, 【1 3 4 2】.

for 1:

push (1, Integer.MAX\_VALUE)

push (Integer.MIN\_VALUE, 1)

for 3

pop (Integer.MIN\_VALUE, 1)

pop (1, Integer.MAX\_VALUE)

push(3, Integer.MAX\_VALUE)

push(1,3)

for 4

pop(1,3)

pop(3, Integer.MAX\_VALUE)

push(4, Integer.MAX\_VALUE)

push(3,4)

for 2:

pop(3,4)

pop(4, Integer.MAX\_VALUE)

return false

```
public boolean isValidBST(int[] pre) {
    Stack<Integer> stack = new Stack<>();
    int root = Integer.MIN_VALUE, n = pre.length;
    for (int i = 0; i < n; i++) {
        if (pre[i] < root) {
            return false;
        }
        while (!stack.empty() && stack.peek() < pre[i]) {
            root = stack.peek();
            stack.pop();
        }
    }
}
```

```
    }
    stack.push(pre[i]);
  }
  return true;
}
```

[on-site] Alien dictionary 的所有排列组合

<https://instant.1point3acres.com/thread/445476>

[phone, 2018/3]

LC212

[phone] <https://instant.1point3acres.com/thread/445970>

实现一个 message broker class, 3 个 methods

1. register, 能注册 listener 到 t, 多线程怎么做, 有哪些方法, 有哪些不同

[phone]

1. 找出一个 list 里面第二小的数。

2. merge 一个 list 里的 interval。如 input:  $[[1,2],[2,4],[1,3],[5,6]]$  返回  $[[1,4],[5,6]]$ 。我用  $O(n \log n)$  的算法做的

[phone]

Uber ATG

电面第一轮

面试官迟到 10min。

coding

一个给定 array, 找所有三元素组满足和大于某 threshold。

[hide=101]

sample:  $[2,1,3,0,3]$ , 5

output: 5

description:

$[0,3,3]$

$[1,3,3]$

$[1,2,3]$

$[1,2,3]$

$[2,3,3]$

[/hide]

一个 3 sum 的变种吧。

[phone]

三个工作日内约了下一轮的 coding interview 的时间, HR 小姐姐回邮件超快, 基本都是一个工作日内回复。

在约定了 coding interview 时间之后还发了一个关于怎么准备 uber 面试的 tutorial, 超级 nice! 好感大增 =w=

其实 tutorial video 说的应该基本都是 <crack the coding interview> 这本书上的东西。但楼主之前没



看完，而且感觉听人说比自己看要生动很多，所以还是蛮有用的感觉。

### 【Coding interview - 60 min】

小哥人也很 nice，前面随便聊了两句就直接开始做题了。（问了下 why Uber? what do you want in next job?）

题目内容：

给一个非空字符串 `input_string`，给一个正整数 `char_limit`。要求完成一个函数，将 `input_string` 里的内容按长度不超过 `char_limit` 的 string，输出到一个 `output_list`。e.g. `input_string = "Hi, Uber! Here is Tony!"`, `char_limit = 10`. Thus, `output_list = ["Hi, Uber!", "Here is", "Tony!"]`。

这个题目很简单，楼主听完题目大概就有思路了。然而。。。

整个面试做题过程 review (50min)：

(2 min) 面试官介绍题目

(4 min) 楼主理解题目并且向面试官确认各种细节/补充条件，并且写完函数的 comment

在此过程中楼主得到了关于题目的一些补充限制：

确定 `char_limit` 是正整数(本来没说)，确定 `input_string` 没有前后和多余的空格(假设是 "perfect" string 没有什么乱七八糟的情况)

====

此时，楼主心里有点小开心，觉得蛮简单的，于是大概说了两句觉得可以怎么做，然后问面试官是直接写还是先说思路在写，面试官表示都可以，于是楼主。。本来打算先做做笔记的，结果一打字。。就一下子没忍住就直接开始写 code 了！！！！【埋头痛哭！吸取教训！】

====

(12 min) struggle in 1st try ...

(5 min) struggle in 2nd try ...

====

至此，楼主为“不思考清楚就开始写”付出的代价大概是浪费了将近 20min...以及...好紧张啊！啊啊！【吸取教训！】

====

(2 min) 写崩两次的楼主深吸一口气，鼓励了自己一下，双手离开键盘，闭上眼睛，开始重新整理思路并且大声说出来！（好吧也没有很大声，就是正常音量）是的我就是要面试小哥听到我的思考过程！你听你听，我会做！就是有点小紧张而已！看！我心理素质多好！

在此过程中楼主又得到了关于题目的一些补充：

(1)输出的最后一个 word 后面不用加空格，下一个 word 前面也不用加空格。比如，举例中输出的不是 `["Hi, Uber! ", "Here is", "Tony!"]`。(2)word 后紧跟的标点符号，除了空格都算在这个 word 里。比如，举例中如果 `char_limit = 8`，则输出为 `["Hi,", "Uber!", "Here is", "Tony!"]`。

(7 min) ok，想清楚了，从头开始一口气写完了。。【看，我说了我会吧。。】

(12 min) 写 test case 测试，debug，继续 test，继续 debug（是的，不是 doc 文件里写，是 ide 里写，写完还能 run test case 来 debug，还能写 print 来 debug，感激涕零。。是楼主运气比较好吗？）

(5 min) double check, found a corner case, implement and test.

楼主向面试官询问了 corner case 处理方式：

如果 `char_limit` 比 `string` 里每个 `word` 的长度都小，就把 `word` 分开输出吧。比如，举例中如果 `char_limit=3`，就只能 `["Hi", "Ube", "r!", "Her", "e", "is", "Ton", "y!"]`

===

最后楼主还想再想想，然后面试小哥说差不多了我来测一下。他跑了几轮测试好像都通过，然后就说 ok 了。

最后大概 7 - 8min 楼主问了些比较关心的问题。这个每个人可能关心的点不一样。

#### 【总结经验教训】

1. 一定要想清楚整个解答过程再动手！一定要想清楚整个解答过程再动手！一定要想清楚整个解答过程再动手！特别是简单的题目！不然写崩一次可能就心态崩了（对 楼主是大心脏 写崩两次还顶住了！）

2. 觉得可能比较好的点：

(1) 楼主上来先写了 `comment`

(2) 在最开始还是说了思路的（所以面试官知道我大概知道怎么写）

(3) 中间写的时候都有边写边讲思路（所以面试官会知道我是因为什么卡住了卡在哪里了）

(4) 中间写的时候虽然没分模块但是大概每一段都有个注释解释这一段要做什么

(5) `test case` 跑的不多，但是每个 `test case` 都有想要测什么，不是随便写的，每个 `test case` 都有测出错误，能帮助 `debug`

(6) `test case` 跑不过能立刻反应过来问题在哪里并且解决（两次 `typo` + 一次边界检测问题都瞬间解决，还有个 `infinite loop` 的问题卡了下，靠着 `print` 解决了）

其实说到底写崩了还是蛮影响的，虽然不知道面试官在不在意我写崩了，也不知道是不是我哪一点打动了面试官所以最后能进 `onsite`。

但是写崩了可能面试官会觉得“这么简单的题目都能写崩”，或也可能导致自己心态崩，特别紧张的话，可能明明很简单的地方就想不过去。

(ps: 楼主这里说的写崩是指推翻原来的代码构建方式重头写，比如一开写了个 `for loop`，写不太下去了觉得可能换一种数据储存方式会更好，就不用 `string` 用 `list`，然后又觉得不用 `for loop` 用 `while loop` + `index pointer` 更好)

这是利口遛巴？

[phone] 鑫疆 萨污舅 变种, 参数变成了每秒是个国人小哥

第一次用 `hackerrank interview`, 感觉他们的 `judge` 系统十分奇怪。

1. 我另外申明一个 `public class` 来写代码, 过不了编译, 说必须弄成一个单独的文件才行. 于是我直接写进 `main class`, 又遇到莫名其妙的 `"on-static variable cannot be referenced from a static context"???` 来来回回调了十几分钟才好, 浪费了不少时间

2. 代码是写好了, 但是写 `test case` 的时候, 要用到 `..timestamp` 函数, `thead.sleep` 函数之类的, 有印象, 但是实在写不对, 小哥看不下去, 说你可以去 `google` (不早说), 又浪费了不少时间

3.然后这个 sleep 函数,明明是吃毫秒数为参数的,不知道为什么在他们那个编译器下变成了秒数了..又调了一会儿才发现.这时已经 over 了,小哥说很可惜.然后问,你还有什么要问的不?

[phone]

蠢口奇拔灵, 原题, 感觉面试官也对这道题不太熟悉。。。说什么都是 good good cool cool awesome awesome 也不接个话。。。

补充内容 (2019-7-28 15:17):

BTW 我就面试前一天刷了 Uber 标签 frequency 前几道, 没想到第二天真的考了原题, 看来刷 LC 标签还是很重要的

[phone]

<https://stackoverflow.com/questions/10005544/interview-question-data-structure-to-set-all-values-in-o1>

3周前

每次setall的时候clean map

hashset<key> historical\_key

hashmap<key, value> map

string all

每次setall的时候 clean map,把value赋给all

put的时候加入historical\_key 也加入map

get的时候先去map里面找 再去historical\_key看之前存过没 存过的话就是all

2周前

可以这样。

设一个global reset value

latest counter

default counter

和Map<Integer , Integer> key 是 index, value 是 index counter

idea 是: no matter resetAll or set default count++:

如果call resetAll, 把default count = latest count

如果call set. 把当时的latest count 设置到map 里。

当call get 的时候, 对比 map 的count 和 default count, 谁大用谁的值。

其实原理和模仿真正的 cache 差不多。

[phone]

国人大哥面的，全程很友好一直给 hint，下面上题。

<https://leetcode.com/problems/longest-substring-with-at-most-k-distinct-characters/>

[hide=200]

<https://leetcode.com/problems/lo> [□□此部分内容已被隐藏，请点击右下角前往一亩三分地论坛查看此贴] end tail。

最后问了 high level 怎么 design uber，感觉面试官想听的就是 build microservices，随便说一就好。

[/hide]

[phone] <https://www.1point3acres.com/bbs/thread-539774-1-1.html>

黑车面试，一小时一道题。permutation 的变形体。题目描述：there is a set of {p1 d1 p2 d2 ... pn dn}, produce all permutations of the set and make sure pi always appear before di.

E.g. p1 d1 p2 d2 is good. d2 p2 p1 d1 is bad since d2 appears before p2.

想一想

后面会 post solution。

[phone]

考了一题不是李寇上的，面经我也没见到过。。。可能看得比较少。可能是新题，跟面试官确认了很多条件，面试官也是现场发挥。

Design APIs for compatible checking

```
public void addNewVersion(int ver, int boolean isCompatibleWithPrev);
```

```
public boolean isCompatible(int srcVer, int targetVer);
```

addNewVersion 是表示现在的 version 是不是 compatible with 之前的 version

isCompatible 是 check 两个 version 之间存不存在一个 false compatible with previous version

LZ 一开始就用简单的方法用 HashMap 存然后 check 的时候 from srcVer to targetVer 这样 check 面试官问有没有更好的办法，于是我想到用 HashMap<Integer, List<Integer>> List 是 sorted 的之前所有的 false 的 version 然后每次 check 用 binary search 找最小的大于 srcVer 的 version，找到的话就 return false, 找不到就 return true。

Add 类似于 新加入 edge, isCompatible 类似于检查是否联通

如果 addNewVersion(int ver, int boolean isCompatibleWithPrev) 参数只有两个，一个 int 是新的 version，一个是 bool 型代表和紧邻的前一个是否 compatible，感觉可以做个链表加一个 hashmap 解决问题：

链表添加的时候，如果和紧邻的前面 compatible 就指向它，反之就断了。 head 指向最新刚加的 version。

hashmap 存 version -> node pointer 的 map。

call isCompatible 的时候，先看 map 里 src 和 target 在不在，然后从 src 沿着链表往前找。断了就返回 false，找到就返回 true。时间复杂度貌似不太好。

[phone, financial payment]

Uber Software Engineering

面的 software engineering 有一年工作经验

电面一个小时

5 分钟 tell me about yourself

50 分钟 问了两道题关于 merge two list of intervals

1. <https://leetcode.com/problems/interval-list-intersections/>

2. <https://leetcode.com/discuss/int...-two-interval-lists>

### 【on-site】

因为上一轮的 onsite, 不同的面试官给的 feedback 不 consistent, 所以..hiring manager 人很好, 又给我加面了一轮

面试官是一个非常 communicable 的小哥哥, 前半个小时类似 hiring manager 的 bq 问题, 讲了一些之前做的产品。follow up 是 best seller of the day, of the month, etc.

我分析, 我之前的系统设计之所以 performance 没有那么好, 是因为 jump too far, 太想表现出来我知道很多系统设计很多时间确定需求和计算 qps 之类的, 所以说到 storage 的时候就没有什么时间了。只能用很短的几分钟来 sum up

在系统设计的准备方面, 除了 <https://github.com/donnemartin/system-design-primer> 这个大家都推荐的 github, 手于 188 才可浏览, 您当前积分为 178。

[查看如何攒积分 Click here for more info.](#) 啥打不开。youtube channel 的名字是 Tech Dummies - Narendra L。链接

链接是 <https://www.youtube.com/channel/UCn1XnDWwhsLS5URXTi5wtFTA>

### 【on-site, NYC】

背景: 楼主面得是纽约办公室的安卓职位 finance platform. HR 在 linked 上找到我的。感觉他们家现在好像是想解决每天 pay out 产生的成本问题, 所以想弄一个自己的 finance platform, 估计是想跳过银行吧, 或者起码可以 negotiate transaction fee.

面试:

题目不难, 但是要求能运行出结果, 感觉也考察一定的熟练度。此题一共两个部分。

第一: create a class 1. implements java Iterator interface 2. receives an int[] as the constructor parameter

然后给一个 int 数组 初始化你写的类, 调用 hasNext() & next() 来测试

第二: create a class 1. implements java Iterator interface 2. receivers an Iterator<Iterator<Integer>> as the constructor parameter

然后创建一个 Iterator<Iterator<Integer>>, 初始化你写的类, 调用 hasNext() & next() 来测试

电面的小哥人很好, 会给很多提升, 帮助 debug. 总之在最后写完了。但是我发现大部分时间都用来 debug java 部分的东西了, 题目倒不难。其实楼主自从工作就一直用 kotlin 写安卓, 一些诸如如何把 list convert to iterator 的 api 都忘了。或许建议大家 kotlin 熟悉的话直接上, 别担心面试官看不懂了。

结果: 过了三天收到邮件被 move on 了。anyway, 因为最近一直在找工作, 有几个面试同时进行, 也没再继续和 hr 探讨原因了。祝大家好运了。

[phone, june] indian 利口斯伞期 这题可以利用 hashset 得到 O(N)的解法

输入一个 二叉树, 和一个 整数, 要求输出 path 的和等于 这个整数的 所有 path

eg: 5

3 9

1 6 7 4 另外一个整数是 9

输出为 5 3 1, 3 6, 9

[phone] indian,

input array of start and end timestamp

找出最多可以 deliver 几个

找了一下 类似这道题

<https://www.geeksforgeeks.org/maximal-disjoint-intervals/>

鑫冠 舅芭疏

[phone] 年长的不会算法

经典电梯设计问题, follow up 如何在多线程环境 实现

【phone】 LC986

[on-site]

1. Data modeling code and design

[hide=200]

需求是 需要知道 Uber rides 和 uber eats 有什么内在关系

问可以看什么 metric

列了许多 诸如 DAU WAU MAU Revenue etc, interviewer 表示够了 我们着重看 WAU

然后从 log data 设计一个 table, 可以提供这个 数据

我给了一个 facebook 面镜里经常用到的 一个 latest date table, 就是 user id 和 最近期一次使用服务的日期和时间

然后通过看 最近一次时间 距离 as of date 距离 是否在一周以内看他是不是 WAU

然后要求 写 group by, 问这个 group by 如果所有日期都跑的话, 量太大, 怎么办

答可以利用前一天的数据做 Increment change, 具体就是 取前一期的 整理好的数据, 然后再取这一期 新 take ride 的人的日期, update 这个前一期的数据

然后问有什么 visualize 简单画了一下

然后问现在加上 Uber eats 怎么办, 其实都差不多, 就是把 eats 也当作一种 ride 加上一个 ride type

然后和 普通 rides 放一起

然后写 query 怎么得到 ride WAU, eats WAU, both rides eats WAU

然后画了画图

期间讨论了一下大概 column 怎么设计, attributes 大概有啥, 什么要 index

[/hide]

2. 两个印度小姐姐。主要问 business sense

[hide=200]

现在 uber 推出了 uber rewards

问怎么设计 data modeling 来 keep rewards data, rewards transaction fact table

再问 现在有不同的 currency, 需要统一 currency, join with currency rate convert to usd

请用 sql 实现

现在给你一个 rewards balance tier 的表，比如上了 1000 就是 gold，怎么 label customer with tier 请用 sql 实现

再问 rewards date 不是 ride date，而是 rewards 进入系统的时间 怎么办，具体步骤差 [□□此部分内容已被隐藏，请点击右下角前往一亩三分地论坛查看此贴] model 的 code，再改进到 10 个 model 的，他说你别写，一个 model 太简单了，我不要，你直接上 10 个

没有进展 给 hint，说如果你有 sql table，里面有一个 column，每一个 cell 都是 json format 的 data 要怎么写 query，表示不会。。。没在 sql 里处理过 json

45 分钟以后也没什么进展，最后 interviewer 说他想让我用 Apache Parquet/ ORC，我表示还没学习过这个，你让我学一下保证没问题

他表示呵呵，拍了拍我肩膀，说 take care (你挂了)

[/hide]

## 6. hiring manager

常规介绍 team，有什么 职能，多少人，有些什么任务

behavior question，ideal next job，what team you want join，conflict，critical review 等

## 7. team lunch 1 hour

两个 team member 一起吃饭，闲聊，了解 team

[on-site, SF, android]

一共五轮，第一轮和最后一轮都是 bq，最后的 bar raiser 还问了一道题利口散逸

第二轮设计推特

第三轮 weather app

第四轮利口散留意

[phone] 很和蔼的中国小姐姐

写了一题 轮换哈希 各位自己脑补英文吧

求昂赛 求加点米 没米了!!

判断 给定串是否是源串的子串 规则如下

abc is sub string of bac

abc is sub string of abcd

也就是不关心顺序的意思

是 rolling hash  $O(n)$  找 pattern is part of str 吗?

[phone]

倒序版的 meeting room II

[phone, June]

李口伞玲耀.写了个  $O(N)+O(N)$ ,code 现跑现测,想到的 case 全都一遍过了.然后 followup,优空间到  $O(1)$ .卡着时间讲完的

时间差不多了开始问问题.一个小时整结束

【phone, seattle, March】

最原始的版本：给两个排序的，没有重复数字的数组，找他们的公共数集 (intersection)

Example:

```
array1 = [1,2,4,6,8] size m
array2 = [3,4,8] size n
return [4, 8]
```

解法 1: time  $O(n)$ , space  $O(n)$  直接用 set, 然后 返回 `set(array1).union(array2)`

解法 2: time  $O(n \log n)$ , space  $O(1)$  assume  $n < m$ , for  $i$  in array2, binary search  $i$  on array1

解法 3: 双指针: merge sort:  $O(m+n)$

follow up 1 :这个电面里没问, 不过常考: 如果数字有重复怎么办, 其实解法 3 依旧使用, 只不过要 implement 对

follow up 2: 这里只有两个 array, 如果有  $M$  个 array, 每个 array 平均  $N$  个数字, 现在要求  $M$  个 array 之间的两两之间的公共数集, 也就是说原题这样的 query 要进行  $M^2$  次, 在什么情景下有什么样的优化 ?

> 说实话这里我愣了一下, 没想出来什么情景, 后来考官问, 如果  $M \gg N$  呢? 这里是个很重要的提示, 这意味着这  $M^2$  个 query 里很多的答案都是 0! 然后我就马上想到了倒排索引, 得到确认果然这是考官想要的(国人就是好说话, 哈哈), 然后 implement 一下

这里要引入一个新的估计量: 假设每个数字出现的数组个数平均为  $L$  个, 则  $L \leq M$  (因为  $M \gg N$ ) 举个例子

```
array1 = [1,2]
array2 = [1]
array3 = [2]
array4 = [1]
```

...

```
array99 = [1]
array100 = [2]
```

(1) 搞一个 map, 就是倒排索引了: 以数字为 key, value 是一个 list, list 里面的内容是这个数字所在的数组,

比如上面的例子,  $1:\{\text{array1}, \text{array2}, \text{array4}, \dots, \text{array99}\}$ ,  $2:\{\text{array1}, \text{array 2}, \text{array3} \dots \text{array100}\}$ ,

time:  $O(M*N)$  space  $(M*N)$

(2) 然后再建一个 map, key 是 array 对子, value 是公共数集, 这个 map 的建立方法是遍历倒排索引, 对每个 key 所在的 list, 两两枚举成 array 对子, 这样的  $\text{array}\{i\}_{j}$  有  $M^2$  个

例子: 利用 (1) 中的  $1:\{\text{array1}, \text{array2}, \text{array4}, \dots, \text{array99}\}$ ,

对 array1\_2 所在 entry append1, 对 array1\_4 所在 entry append1, 对 array1\_99 所在 entry append1...

最后这样的  $\text{array}\{i\}_{j}$  有  $M^2$  个, 但是事实上这个操作只进行了  $L^2*N$  次

复杂度:  $O(L^2 * N)$ , space  $O(M^2*N)$

然后比较一下复杂度

如果按原来的解法 3, 复杂度为  $O(M^2 * 2N)$

如果按倒排索引, 复杂度为  $O(L^2*N)$

可以看到哪怕是  $L=M$ , 依旧是倒排索引占优, 减少一般时间, 如果  $L < M$  越小, 优势越大, 事实上  $L \ll M$

[phone, May]

2D array, black and white blocks. Only white blocks 可以走。

给两个点, 求从一个点到另一个点的 path。followup : 求最短的那条 path

[phone]

find overlap in two interval, 并实现 interval class



[phone, onsite, June]

Phone:

通过 BST 的前序排列和层级排列, 恢复 BST

Onsite:

1. bq

2. 航班行程数据找出最长的路径长度, 时间复杂度以及如何优化(dp)

3. 设计 Uber, 主要 focus 在 Dispatch 和 ETA estimation

4. 检查 sudoku 的合法性

5. bq

一天之后通知给 offer

[phone]

面经的题

[hide=200]

给两个排好序的 list of i [□□此部分内容已被隐藏, 请点击右下角前往一亩三分地论坛查看此贴] 的输入

[phone] basic calculator

[phone, May]

most engineering challenge experience

利特口的 岛的个数 + test cases+ followup 岛形状的个数所有代码都要 run 结果

老师好, 看到这样一个面经题目, 请问怎么处理比较好?

能否不开 thread 然后当有新数据或者需要查询 value 是检查是否 expire。谢谢!

题目是实现一个 hashtable 能够对每个 entry 设置 ttl(time to live), ttl 一过就删除。一开始感觉有点像 LRUcache, 就往 linkedlist 上想了, 但是纠结于什么时候 check 是否要删除 entry, 后来又想每个 entry 插入的时候开一个 thread, 然后 ttl 后这个 thread 把对应 entry 删除, 问了有什么优缺点, race condition, 如果系统 thread 没有准时删除怎么办?

添加评论

分享

## 1 个回答

0

刘同学

学员

发表于 12/9/2018, 7:27:53 AM · 0 个赞

一般来说, 设计 expire key 有两种思路:

1. **Passive delete:** 当 key 被访问的时候检查其创建时间，如果跟当前时间之差超过其 ttl 的话就删除。容易想到 passive delete 的优点是把对 ttl 的检查次数降到了最低，只在一个 key 被访问的时候才检查是否应该删除。此方法同时也不存在“没有准时删除”的情况（准确地说，虽然应该被删除的 key 不一定都已经被删除了，但是用户观察不到这一点）。但是这样可能会造成大量的 expired key 在内存里，造成空间大量浪费。
2. **Active delete:** 开一个后台线程周期性地遍历整个列表，并删除超时的 key。Active delete 解决了上述的空间浪费的问题，但是当 key 很多的时候每遍历一次都会占用大量处理器资源，造成系统间歇性卡顿。此外，如果周期设得太长，可能会造成本应该已经被删除了的 key 还能被访问到。

Redis 中采用了一种组合的办法：当访问某个 key 的时候对其进行检查，同时在后台进行以下周期性任务：

1. **随机抽取**一部分 key，进行 active 的检查，删除超时的 key。
2. 如果删除超时的 key 占随机抽取的 key 超过一定比例的话，说明当前可能存在大量的 expired key，因此再回到第一步继续删除，否则结束此次扫描

这种方法应该是一个不错的 trade off。

至于 race condition 的问题，如果后台线程在删除一个 expired key 的同时有用户访问了该 key，则会出现 race condition，因此需要加锁。

(Reference: <https://redis.io/commands/expire#how-redis-expires-keys>)