

Last updated: Jan 26, 2019



>> L4 Salary:

130k ~ 165k,

6000 ~ 7888 share / 4 year or max:480sk

13.5k cash + 52.5k equity

>> 系统设计:

- design uber
- design一个system查询在某个时间段, 某个城市, 有多少人打开了Uber app
- moviepass online booking
- System design. 设计一个[Amazon](#).
- system design: Hot Map问题, 然后问了下public api怎么design还有怎么聚合data, 都是比较常见的system design问题。
- 让你设计个git repo system with async build & continuous integration & continuous deployment
- constant hashing。
- 如何实现在rider的map上显示地图和车
- rate limiter的api design,
- 从前端到后端设计一个[Amazon](#)的相关产品推荐系统 -- 因为是电面, 所以天马行空的说说自己的设计思路, 然后面试官追问了一些推荐方法的细节, 因为我没有机器学习的背景所以就临场想了想naive的解决方法, 感觉答得一般但还是过了。
- 设计题:做一个网站来卖火车票。这个题目看看12306的卖票系统设计blog应该有很多启发
- 设计 Netflix, 还是一个经典的题, 其中问的比较细节的点是数据在多机房之间的同步问题, 我说可以做 cluster replication, 再做一些脚本 check 之类的, 他说行吧
- 系统设计 设计一个[Amazon](#)类似的在线购物平台 特殊要求: 比如MacBook 这个item的价格需要不停的根据region timewindow来变动 每一个region都有自己价格变动的policy 怎么设计?
- design google spreadsheet, 从online service的角度回答 (分享doc, 在线存储, 同步, 并发等等)
- 土耳其大叔, 系统设计火车订票系统, 觉得答的挺好的, 大体设计出来之后问了动态票价要怎么设计, 交流也没问题, 结果貌似拒recruiter说设计有点弱
- design netflix recommendation system问我应该怎么给用户推荐, 选哪些features, 我只说了按照genre分类用户选了某些题材以后可以推荐同类型的, 然而hm并不满意, 后来在网上看到如果两个用户有相似的浏览记录那可以group together用一个的click给另一个人推荐, 所以正确答案是啥有没有小伙伴知道。
- KV store
- 请设计heat map, 动态显示每个block车的数量。
- what is big table. How it is designed.
- 设计一个发邮件的系统, 给内部用。要自己问他有什么要求, 然后就是要我先写API之类, 再讨论scale和情况之类的。
- Brief system design: given a coordinate, find top K nearest bus stations (geohash + NlogK)
- autocomplete/typeahead
- uber driver会每隔一定的interval发送自己的信息给uber backend service, 设计一个系统能够计算在

driver的ping的数量。

Save Copy to Evernote

排序

感觉实践中这个题目可以有变化，侧重不同部分，建议多想想相关feature.....准备的时候我主要看的是关于Geo的部分和如何实时汇报位置，实际面试中面试官并没有对这部分深入追问，我说了Geo Hash又提了一下可以用google S2就满意了。但是让我具体写了一下数据库schema。还问了如果我们要schedule ride（比如说提前预定明天凌晨去机场的ride）应该怎么办。我答的先提前问司机们谁愿意接，然后在预约时间接近时如果提前接订单的司机cancel了或者距离pickup地点距离太远则重新找司机。

后来答完了时间富裕就有问了payment的问题。ride结束后，payment一般不会立刻处理，而是先暂存信息随后再集中处理，问这种情况下该怎么设计。没准备就瞎说的，不过似乎因为是附加题也不是很重要。面试官给方案是因为可以delay，可以不使用任何数据库，而是直接把raw data写到HDFS，然后集中进行batch处理。

一个细节是开始问payment相关问题之前对方看了下手机说我们还有些时间，所以我认为后面的payment部分是附加题...

system design, netflix选出每个题材下，在7天，30天和总共的时间段内view最多的10个视频，当时给的定义是如果一个用户在一天之内view了video长度的90%以上那么就算一个view，一个用户会不停的往server发(user, video, 看video的start和endtime)，然后脑子里瞬间想到各种算法题merge intervals算video的covered时间（但是如果动态添加interval就不知道怎么搞了），以及当时总觉得有个什么data structure可以根据view的个数维持一个video的sorted array并且能动态更新view的数量，面完以后想了想应该是LFU。但是这不是system design吗。。。中年烙印全程面无表情不引导话题走向，到后面甚至玩上了手机，看来是凉透

berEATS 有時候會發廣告在FB feed or somewhere, 請設計一個系統紀錄使用者點了廣告後, 有多少人安裝app, 註冊帳戶, 甚至點餐. 從user flow 討論到data store, 再討論scalability. 感覺有些部分跟第二道題有點像

这个是那个面试官自己实际做的一个项目相关，就是他需要在很多平台投放uber相关的广告，但不知道每一个的效果如何，以及如何控制campaign budget
就是比如uber每天有1000块钱的budget投放uber的广告在各个平台（例如google, fb, ad exchange），怎么知道每个平台的转化率conversion（多少用户在各个平台看了uber的广告并且点击注册了等等），如何根据这些转化率或者其他你觉得有用的点优化之后的budget分配
如果你不是面ads组不可能考这个。这题是那个面试官当前正在做的project. 我做过相关的。所以知道一些

设计一个系统，拿到的input是JSON，但是内容不一。可以是以一个Email的形式包含from, to, cc, bcc, main content, attachment, etc. 也可以是别的任何的record（比如医疗record，交通record，etc）basically the input can be anything，并且我们不要求用户/Input source 标明是什么类型的input。

主要设计出来怎么处理这些输入，然后怎么存储它们。要求是存储的时候需要已知 input type，之后读取的时候可以直接通过 input type 来进行后续处理。

计题：设计一个搜索引擎的自动补全功能。这个题目需要考虑如下问题：

1. 如何补全突发热点事件
2. 最重要的是？ --- low latency, 如何保证低延迟？
3. offline应该有一个data aggregation job，它定期处理最近的搜索记录
4. 补全数据怎么存储比较高效？ -- trie tree
5. 数据库的sharding如何做？
6. cache的redundancy怎么做？

看过之前的面经帖子，老题，设计youtube，面试官是个中国老哥，一上来就问我能不能讲中文，我说当然可以，之后就是全程中文交流。其实一直不放心system design，自己经验不足，又不太会准备，但今天遇上老题和中文交流就让我舒心不少，最后老哥觉得还可以，把白板上我画的东西拍照带走了。

就是把不同城市的请求，放到不同的server中处理，问怎么分配，如果两个城市很近（距离低于threshold）就要放到一个server中，还要平衡server之间的load

ck traffic of ad clicking
ice proxy, 每個被clicked Ad的link 都會發經過nign: Save Copy to Evernote
這邊聊了一下, 怎麼offline process nginx 的log, 並且怎麼filter request 中的

from every source.

第一轮: 更像是数学题, 一个load balancer后面有n个host, 每个host的weight或者权值不同, 问怎么设计一个数据结构, 使得源源不断的request在call LB的时候, LB选择host的概率分布是基本按照hosts的权值大小的。例如, LB后面有三个host, 权值分别为1, 4, 5, 这样一来, 如果我call LB 100次, LB大约10次选择第一个host, 40次选择第二个host, 50次选择第三个host。 . 1point3acres

第二轮: 给一个树, 然后一个API, shouldRemovable(TreeNode node), 其返回tree/false, 表明节点是否应该被删除。问题是遍历一遍树, 然后根据这个API删除应该被删除的节点, 返回森林的所有root节点在List中。

第三轮: 设计一个可以scale的metrics系统

第四轮: 设计一个股票系统, 支持查询当前时刻某只股票的股价, 当天的涨幅最大和跌幅最大的股票, 和某只股票历史股价。

第五轮: behavior轮, 跟hiring manager说说以前的工作经验, 还有怎么解决问题等等

>>BQ:

- 举一个合作比较好的project的例子和一个合作不好的例子?
- 如果用户发现bug怎么办, 怎么保证不再出bug
- 这一轮面试官问了半天我的一个project, 不停地follow up, 如果需求变化了应该怎么修改, 如果请求量大了, 如果数据量大了怎么办之类的。
- 重新设计自己做过的project
- 最challenging的project
- 如何给同事feedback
- 如何对待批评

>> OOD:

- music player
- controller

让你设计一个类, 实现excel表格的get, set
每个单元可以输入字符串, 如果字符串以"="开始,
后面会跟一个表达式, 比如x+y-z, 只有加减, 没有乘除和括号。
对于表达式输入, 要求值再存入表格单元。

>> CS fundamental

- telnet
- linux kernel
- sharding
- restful api

>> Counting.

topological

Serialize and Deserialize Binary Tree

interval problem

Trapping rain water

LRU

meeting room ii

设计advanced hash map多一个功能是从map里面返回一个随机的key, 完了问corner case如果map是空的随即返回key应该怎么办,

Given: an array of unsorted numbers, and a target number;-baidu 1point3acres
return: all the permutations of combinations of array elements sum up to the target number.

For example, [2,5,7], 7

return: [2,5], [5,2], [7]

一个event流, 每个event代表log in或者out以及对应timestamp, 每次in用户+1, 反之减1, 不考虑user id匹配
写两个function, 一个是handle_event(event), 另一个get_active_user_count(timestamp), 返回对应timestamp之前范围内的系统内已登录用户总数

例子

login 10

logout 20

login 30

login 40. 1point3acres

logout 50

get_active_user_count(45) -> 2

get_active_user_count(20) -> 0

get_active_user_count(70) -> 1

解法:

linear scan -> O(N)

prefix sum + binary search -> O(lgN)

follow up: 如果event是乱序插入怎么办

这个没答好, 最简单是sort但复杂度有点高, 感觉用segment tree没法处理random insertion (因为需要更新后面的segments)

写一个trie, 有function -

void insert(String s)

boolean search(String s)

boolean startWith(String s) //就是看prefix

boolean delete(String s) //不存在就return false

然后跑test cases

1. 判断一个tree是不是bst

2. 求bst中第n大的数

element是一个数组。例如: `[[1, 5], [10, 16], [2, 7],[1000,`

时间, 第二个数是送乘客到达时间。问, 从这个输入数组来给: [Save Copy to Evernote](#)

你和你的兄弟都有一个wishlist, 找出和你相似度最高的。follow up是给出一个max值, 找出你的buddy的wishlist里不在你的wishlist里的最多max个城市, 根据buddy和你的重合程度来排序

例如:

你的wishlist是 a,b,c,d

buddy1 的wishlist 是 a,b,e,f, 有两个和你的一样, 所以是你的buddy

buddy2 的wishlist 是 a,c,d,g, 有三个和你的一样, 也是你的buddy

问题是输出一个size最多为max的推荐城市列表。当size为10时, buddy1和buddy2的wishlist中不在你的wishlist中的城市都可以加入推荐中, 因为buddy2的重合度更高, 所以先输出buddy2中的, 所以推荐为 g,e,f 当size为2时, 推荐是g,e 或 g,f

代码我只写了重合度排名, 推荐的话可以按照相似度从高到低遍历, 找出不在于你的wishlist中的输出, 输出过程中可以标记是否已经输出

```
def find(self, nums, nums2):
    return sorted([(sum([num in set(nums) for num in nums2[i]]) / float(len(nums2[i])),
i)
                  for i in xrange(len(nums2))])
```

input:

boa, chase, 100 (代表boa给chase打了100刀)

chase, boa, 200

boa, wellsfargo, 300

wellsfargo, chase, 400

chase, wellsfargo , 150

输出

boa, chase, 100 (代表boa最后收到了100, 而不是chase, boa, -100, 因为要求数字是正数)

wellsfargo, boa, 300

chase, wellsfargo, 250

1轮, 白人小哥, lz以为面的是software engineer, 结果小哥上来说你这个是sre职位, 来个scripting吧。卧槽? 我说我和recuriter说了不是sre啊, 他说可以skip scripting啊。然后要求小哥换了一道题。地里见过但是写砸了

parse一个log, 按request method+root path计算count。比如:

```
#199.120.110.21 - - [01/Jul/1995:00:00:09 -0400] "GET /shuttle/missions/sts-73/mission-sts-73.html HTTP/1.0" 200 4085. Waral 博客有更多文章,
```

```
# burger.letters.com - - [01/Jul/1995:00:00:11 -0400] "GET /shuttle/countdown/liftoff.html HTTP/1.0" 304 0
```

```
# 199.120.110.21 - - [01/Jul/1995:00:00:11 -0400] "GET /shuttle/missions/sts-73/sts-73-patch-small.gif HTTP/1.0" 200 4179
```

```
# burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 304 0
```

```
# burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/video/livevideo.gif HTTP/1.0" 200 0
```

```
# 205.212.115.106 - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/countdown.html HTTP/1.0" 200 3985. From 1point 3acres bbs
```

```
# d104.aa.net - - [01/Jul/1995:00:00:13 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985.1point3acres网
```

6 GET /shuttle. 牛人云集,一亩三分地

1 GET /images

1 GET /

需要有set(key, value), get(key) 和 setAll(value)。setall就是把所有的value都设置为一个新的value, O(1)复杂度。细节没有处理好, 日常犯蠢。

每次set(key-value)的时候, 把时间也存上;

2.setall的时候, 就存一个global value, 并且也把时间存上;

3.get(key)的时候, 判断你的这个key的local时间点, 在global的时间点前面就返回global value, 否则就返回自己的value

每个event都有时间startTime和endTime, 比如

event(2019-01-01 00:00:00, 2019-01-01 12:00:00)

问最多有多少个event同时发生

给一堆线条, 每个线条由两个点表示, 求所有的blob, 输入为list of lines

[(p1, p2), (p3, p4),]

or

[l1, l2, l3...]

返回list of blobs

blob的形式可以自己定义

[(l1, l2, l3), (l4, l5, l6, l7)...]

or

[(p1, p2, p3), (p4, p5, p6, p7)...]

题目非常open, 主要看思路, 不要求写代码

construct string -> multiple white space -> one space

acyclic graph, give start node, find all nodes k distance from this node

longest distance of increasing subsequence

题目是处理字符串

输入有两个参数, 一个是字符串, 这个字符串是html, 比如

<p>helloworld</p>

第二个参数是整数, 表示所读入字符的个数。

这里要注意的是html的tag是不算在读入个数里的。上面这个例子里, 比如我要读5个字符, 那应该读到hello, 所以最后输出结果应为

找开头就是用index强行扫一遍string

用stack存见到的tag

读满以后把stack里的东西pop出来，在结果里append对应的closing tag

valid preorder BST

有M个task, 每个task执行时间为t1, t2, t3, ...tm, 然后机器同时只能执行N个task, 其中N远小于M, 问 怎么安排, 才能使所有的task全部执行完所用时间最短。

find common elements from arrays

Parse CSV???

随鸡数按比例产生

stock buy and sell

implement binary search tree. Insert, find.

trie类题目。

sort colors

find median in two sorted array,

k sorted list 要用iterator

第二轮 lru 变形。值是字符串。capacity是字符串长度。

一堆数组, 找所有数组重合的element,

给你一个起点和终点的时间, 按月输出这个中间的所有时间。如果给你2019-1-10, 2019-3-15, 输出2019-1-10 to 1-31, 2-1 to 2-28, 3-1 to 3-15

给一组string, 形式都是string+int, 比如aaa130. 输出排好序的string. 排序规则是先排string再排int. 比如说aaa2比aaa100要小, 因为aaa一样大但是2比100小. java里的东西可以随便用.

实现一个Map<String, Integer>并具有如下属性: O(1) set, get, del, getRandom. 难点在于O(1)的getRandom.

给levelorder和
inorder数组 构建二
叉树

比如 set1: [1, 2] [4, 7], [9, 10], set2 : [2, 3], [7, 9] 返回: [1, 3] [4, 10]

来update一个面经。国人小哥。

刚刚面完估计是挂掉了。

给一个n*n的slashes 2d matrix如[['/', '/'], ['/', '/']]

要求

1) 生成一个n+1*n+1的图, value是1-n*n, 连线是按照slash的方向, 每条外边也要连上

123

456

789. From 1point 3acres bbs

对应slash的内联: 2-4 3-5 5-7 6-8

2) 实现给两点输出所有路径的方法

完第一步时间就差不多了，gg

每一步不难的，但是算value啊boarder用掉太多时间了，关键还是编程不熟悉吧

大家要是遇到这样的超别紧张，慢慢来，一步步写会好很多。

补充内容 (2018-10-11 11:42):

```
slashes=[['/', '/'],
          ['/', '/']]
```

要求输出是下图

```
1--2--3
|  /   / |
4  5   6
| /   / |
7--8--9
```

短信分割

输入：长度限制 = 20, "Hey Alice, your Uber is arriving now!"

输出：["Hey Alice, (1/3)", "your Uber is (2/3)", "arriving now! (3/3)"]

要求：

- 1) 词不能截断
- 2) 需要返回短信的条数以及第几条，如(1/3)

2D rotated sorted matrix找target,

321456

216789

456321

在HackerRank上做題~

一個關於下棋~ 寫一個function for checking who win. 給一個2D Matrix, 還有位置~

要測當時這個狀態是誰贏~

規則是像圍棋那樣~如果上下左右都有對手的棋~ 也就是會死掉~ 自己的棋就會被拿起來~

如果上下左右還有空間~ 就還會活著~

如果旁邊是自己的棋子也就還活著~ 只要自己的棋沒有被包圍~ 就還活著~

最後return true/false.

基本上~ 我用BFS做~ 加上Set去判斷走過了沒~

第二個題目：

關於Uber pool~ 可以先pick up A, Drop A, Pick B, Drop B...etc~

基本上~ 寫一個function去回傳所有的可能組合~

給有幾個人~ 然後回傳所有的可能的pick-drop組合~

例如~ 有AB~ 可以先Pick A, Drop A, Pick B, drop B,~ 也可以PickA, pick B, drop B, drop A.

總之一定要先pick再drop~

這題我基本上用DFS加backtracking~

Sort string.

{aa, ab, ad, aabc} ->{aa, aabc, ab, ad}

文件夹下(包括所有子文件夹中)所有重复文件组。重复文件的定义是内容完全一样。每个
写pseudo code。Follow-up是如何提升效率的查找? (Save Copy to Evernote

How to generate a random number with weight (LC乌尔吧)

问how to generate a random number with weight, with update weight function

LC暂时没有, 答案是segment tree轻松秒杀, 没时间写了有点长, 讲了步骤

利口物流的变种, 输入是两个array of intervals, 求这两个列表的intersection 和 union。举个栗子: a: [[0, 2], [5, 10]], b: [[1, 5], [10, 12]] intersection: [[1, 2], [5, 5], [10, 10]] union: [[0, 12]]

uber需要给客户发短信, 整条message可能会很长, 他们就需要把一条长的短信分成很多个短的短信发出去。但是用户接受到的短信顺序可能和原顺序不一样, 所以每条短信后边都要有一个 (k/n) 举例: "this is a message, this is a message, this is a message" --> 第一条: "this is a message (1/3)", 第二条: "this is a message (2/3)", 第三条: "this is a message (3/3)", 你的任务是写一个程序来计算每一个短信的内容。保证每条短信的字是完整的, 而且每条短信要塞进去尽量多的字。注意 (k/n) 里边的n有可能是多位的。有可能是 (2/5) ... 也有可能是 (100/1000) ...

给一个 M x N 的网格, 表示一个没有岛屿的海洋, 需要实现两个函数:

1. insert(i,j) 在 (i,j) 放置一个岛屿

2. query() 返回大陆数量, 上下左右连接在一起的岛屿算同一个大陆, 其实就是连通块的数量

先给了每次 query 做一次 dfs 的方案, 问有没有更好的做法。肯定有啊, 然后用 UnionFind 秒了。

但是后面花了几分钟查代码里的小问题, 没有做到 bug free, 尴尬。

M x N 的网格, 扫雷游戏, input 告诉你所有雷的位置。

第一步, 找出并标记数字, 就是循环每个格子, 看其周围的 8 个格子一共有多少雷。

第二步, 游戏开始, 实现一个 mine(i, j) 函数:

1. 如果挖到雷, 返回 Game Over

2. 如果挖到数字, 返回数字

3. 如果挖到空白格子, 需要打印此次点击打开的区域 (遇到数字为止)

整体就是一个 bfs 的过程, 不麻烦。

Find targeted number in sorted list.

他说如何利用第一次二分, 优化第二个二分。

就是第一个二分的过程中。改一下left, right, 这样upperbound的时候, 可以直接从left, right开始而不是0, length

算法 很简单 类似于number of island 一个棋盘 有empty, 黑棋, 白棋 被别的颜色围绕的话 中间这个棋子可以被captured... 问一个coordinate 可不可以被captured follow-up 如果很多相连的 问所有相连的可不可以被captured 一个bfs就可以了 问如果棋盘巨大 100M的量 如何处理 可以用uber用的geo cut 分区进行mapper reduce 就是要记录一下边界的状态

Construct n-ary tree from ancestor relation, and return the root of the tree

```
class TreeNode {
    int val;
    List<TreeNode> children;
    TreeNode(int val) {
        this.val = val;
        children = new ArrayList<>();
    }
}
```

input : [1, 1], [1, 2], [1, 3], [1, 4], [2, 2], [3, 3], [3, 4], [4, 4]

output

```
1
 / \
2   3
    \
     From 1 point 3 acres hbs
```

```

"(* 1 2 3)" --> 1 * 2 * 3 = 6
"(* (-4 3) 2 (+ 1 2))" --> (4 - 3) * 2 * (1 + 2) = 6

```

题目是写一个hashtable, 不用考虑多线程和generic, key是string, 需要有get, delete和update, put其实就是update。就像普通的hashtable一样bucket是linkedlist就可以了, 用string的hashCode做bucket的index。楼主好久不写java, debug花了一会时间, 不过还是没啥问题。

之后是followup这个就比较坑了, 要求实现一个功能updateall, 可以把所有的key都update成这个新的值, 我确实不知道hashtable有这样的功能, 需要常数时间。正确的做法是设置一个global value, 每个value, 包括global value都要有last updated timestamp。这样每次update都update timestamp, get的时候就比较一下global的和真正value的timestamp就可以了。楼主一开始确实是这么想的, 但是因为刚写完处理conflict的代码, 所以觉得这并不是O(1)。就没说, 后来说了global value和timestamp。面试官说这就是正确的思路, 但是我觉得这样不是O(1)啊, 面试官说不用考虑哪些细节, hash function够好就可以, 好吧。。。可能这里是挂点吧

```

"+12"
      +
      1 2
  变成
. check 1point3acres for more.
"++123"
      +
      1 2 3
      +
      1 2

```

(+, -, *, /) is parent node. number is leaf node.

Given a stream of integers, output a sequence of size k whenever it is possible.

For example, if the stream is as follows and k = 4

10, no output, buffer contains 10

8, no output, buffer contains 10, 8

12, no output, buffer contains 10, 8, 12

13, no output, buffer contains 10, 8, 12, 13. check 1point3acres for more.

11, output [10,11,12,13], buffer contains 8

follow up:

Scalability. If we have multiple streams of inputs, and integer values are not equally distributed, how to handle such a thing?

uber的电面, 问了一道题。给定一串log events, 每个event包含timeStamp和login/logout, 那么给定一个timeStamp, 问有多少人login.

Example:

100, login

300, login

500, logout

700, login

如果timeStamp = 600, 那么只有一个login

题目很直接, 给一个message string, 要把它break 掉, 要在空格的地方断掉
每一行限制字符数, 比如说每行只能有10个字符

写代码的时候有一些corner case比较烦, 比如说可能有的单词很长怎么办。

题目我表述错了, 不是BST, 是普通二叉树, 判断是否有大小大于等于4的重复子树, 例子:

```

      5
     / \
    4   3
   / \ / \
  3  4 2  3
 / \
1   2
   /
  1

```

```

// Create a new object called Coverage
// Coverage stores a list of ranges (aka intervals) for integer values
// ranges are inclusive
// Within a single instance of a Coverage, the ranges within must not overlap (uses
fewest number of ranges possible)
// implement 2 methods fro this class.
// 1) addRange(int start, int end) : add range will add the specified range, and resolve
any overlapping issues
// 2) toString() : print Coverage contents for debugging

```

```

// example
// Coverage c = new Coverage()
// c.addRange(0,10)
// c.addRange(20,30)
// c.toString() --> [(0,10),(20,30)]
// c.addRange(5,25)
// c.toString() --> [(0,30)]

```

```

interface Limit{

    public void setLimit(int qps);

    public boolean canSendRequest();

}

```

```
res
  ()=>true
```

```
time=0.1: canSendRequest()=>true
time=0.5: canSendRequest()=>false
time=0.6: canSendRequest()=>true
```

需要写实现和test case

. From 1point 3acres bbs

一开始对题目理解的有点偏差，还好后来经过提醒给出正确解法。多谢国人大哥放水。

c380变形，给定两个数组， $E=\{1,2,3\}$ ， $W=\{2,3,5\}$ ，代表元素和相对应的权重。设计一个API能够根据权重随机获得相应的元素。

解法1：放一个很大的数组，数组大小为权重之和，然后把元素依次放进去以后随机取

Follow up: space/time complexity? 如何改进?

四月份的电面，答得不是很好，最后几分钟才调完了bug，感谢没直接放弃我，给了我一次加面，不过当时有其他offer了，就没再继续。。。

```
// example input:
private static HashMap<String, String[]> MEMBERS_BY_GROUPS = new HashMap<String,
String[]>() {{
    put("Group0", new String[]{"Group3", "User0", "User1"});
    put("Group1", new String[]{"Group3", "User2", "User3"});
    put("Group2", new String[]{"Group3", "Group5", "User4", "User5"});
    put("Group3", new String[]{"Group4", "User6", "User7"});
    put("Group4", new String[]{"User8", "User9"});
    put("Group5", new String[]{"User10", "User11"});
}};

// example output
private static HashMap<String, String[]> GROUPS_BY_USERS = new HashMap<String,
String[]>() {{
    put("User6", new String[]{"Group1", "Group0", "Group3", "Group2"});
    put("User7", new String[]{"Group1", "Group0", "Group3", "Group2"});
    put("User4", new String[]{"Group2"});
    put("User5", new String[]{"Group2"});
    put("User2", new String[]{"Group1"});
    put("User3", new String[]{"Group1"});
    put("User0", new String[]{"Group0"});
    put("User1", new String[]{"Group0"});
    put("User8", new String[]{"Group4", "Group1", "Group0", "Group3", "Group2"});
    put("User9", new String[]{"Group4", "Group1", "Group0", "Group3", "Group2"});
    put("User10", new String[]{"Group5", "Group2"});
    put("User11", new String[]{"Group5", "Group2"});
}};
```

>> Leetcode:

word break

(longest substring), 17, 20, 34, 36, 49, 51, 56(要找到合并之后最长的interval), 62, 68, 75, 88, 97, 98
 103, 133, 138, 139, 161, 165
 200, 208, 217, 221, 239(no tree map), 273, 297
 302, 329, 341, 359, 361, 362, 378, 381, 385
 450,
 528,
 616, 629, 636, 640, 658, 695,
 736, 745, 767
 915

>> 一个水友的leetcode 2018 4~6月 总结。

```

# Math
* Excel Sheet Column Number
* Excel Sheet Column Title
* Roman to Integer
* Happy Number
* Fraction to Recurring Decimal

# Bit Manipulation
* Power of 2
* Power of 4

# Array
* Spiral Matrix
* Group Shifted
* Meeting Room
* Number of islands
* 3Sum Smaller
* Merger Intervals
* Insert Intervals

# String
* Reverse Words in a String
* One Edit Distance

* Reorganize String. From 1point 3acres bbs

# Anagram
  
```

Palindrome

- * Valid Palindrome

- * Palindrome Permutation

- * Palindrome Pairs

Hash

- * Two Sum

- * Word Pattern

- * Valid Sudoku

- * Encode and Decode TinyURL

Linked List

- * Reverse Linked List Recursion

- * Swap Nodes in Pairs

- * Copy List with Random Pointer

Stack

- * Min Stack

- * Exclusive Time of Functions

- * Asteroid Collision

Binary Search

- * Search in Rotated Sorted Array

- * Find Minimum in Rotated Sorted Array

Binary Tree

- * Maximum Depth of Binary Tree

- * Using recursion to find the maximum of left and right subtree

- * Serialize and Deserialize Binary Tree

Binary Search Tree

- * Delete Node in a BST

- * Inorder Successor

- * Binary Search Tree Iterator

- * Kth Smallest Element in a BST

- * Serialize and Deserialize Binary Search Tree

- * There's no difference than serialize and deserialize binary search tree.

Heap and Priority Queue

- * Find K Pairs with Smallest Sums

- * Top K Frequent Words

- * Top K Frequent Elements
- * Merge K Sorted Linked Lists

[Save Copy to Evernote](#)

- # Trie
 - * Implement Trie
 - * Replace Words
 - * Add and Search Word

- # BFS and DFS
 - * Employee Importance
 - * Flood Fill
 - * 01 Matrix
 - * Shortest Distance from All Buildings

- # Graph
 - * Course Schedule
 - * Clone Graph

[Terms of Service](#)[Privacy Policy](#)[Report Spam](#)