丢📦 copy

---------- 整理与6.23----------

电面：

duplicate files 2

folder access

Coding：

token buckets + multi-thread

上机结对编程（提供python flask网络开发项目的基本代码，三个要求，一把一个函数从同步执行改成异步，二基于一求出多次执行后返回值的平均值，三求出基于一的最高的五个数值，随机应变即可，不难，有python网络开发经验的很容易）

id allocator. how much memory is needed for each solution? what are the runtimes? how to optimize and make tradeoff between runtime and space.

implement a web crawler and then make it multi-threaded and thread-safe

```
让你写一个API，可以传进去一个function和一个时间，然后这个function会被延迟这么多时间被执行。 这个API可以被反复call， 相当于可以schedule不止一个要被延迟处理的function。
Delay queue
https://www.jianshu.com/p/a8c1458998aa
```

Multi transaction KV - store

第一题NASA Panorama 所有图片都存在disk， 怎么Update和fetch啊？ 就是用hashMap key: x + y，value: image path

```
Photo Max Count
1. 你有一堆photo_id，id每出现一次表示一次访问，返回访问次数最多的k个photo_id
2. 假设输入是stream，设计一个类，实现get_current_top_photos方法，返回当前访问最多的k个图片
```

Sharpness Value

系统设计：

设计Twitter 主页feeds

设计一个调度器，会有很多的任务进来，然后会有很多的worker, 怎么样去判断process 已经成功完成了

设计图片上传后生成缩略图的服务
给你一些producer，发送不同的image url到你的service，你的service需要有consumer，来读取image的url，然后把image download下来。其实就是想让你design 一个mesasge queue。follow up有咱们scale，consumer machine down掉了怎么办，如果message queue down掉了怎么办，down掉后重新开启有什么问题。最开始还让写了一下service的各个API。

技术很强的白人geek。用户发request是一个image的url，让design一个系统process这个request并发给consumer，不同consumer接收，然后处理，变成thumbnail存在db。整个过程还比较smooth，就是他会问各种情况，consumer失败了怎么办呀，process这个queue的机器down了怎么办呀啥的。

design a log system that allow application to record server events such as hardware replacement / request response error. Data can then be used offline and online for analysis.

design dropbox
先是让我自己从头到尾high level设计 然后主要问了一下如果一台设备上传了一个文件 是如何处理的 显示把文件分割 然后上传 保存meta data信息； 之后又问如果另外一台设备登录 如何synch up这个文件。 又问了如果有update文件如何做。 主要问了 更新信息如何存储 这样另外一台设备登录时能够抓取更新信息进行本地更新

非技术：
walk through 1，2个项目

---------- 整理与3.16----------

3.16 第三页
>> 2017年一个小兄弟整理的。。 17 - 19 年，题目感觉几乎没有变化啊。。。
https://www.evernote.com/shards/s424/sh/2804d7e0-c93c-455d-9b79-027caa1750c5/48c247d32264b0e5739c5fc27c7eaff1
https://www.evernote.com/shard/s424/sh/abe1558e-b512-4ff0-929b-f305221166e0/3feb3686c7b488d341ddfec1629a836f

>> System Design
- 写一个算法来实现callback + timeout, 要先发现penalty 然后再写一个算法去分配谁先执行，涉及到notify() 和 sleep()
- 设计Twitter 主页feeds
- Design dropbox
- 不是典型的Design设计，面试官出了他之前工作类似的一个项目。 设计一个电话簿， 假设我的team是application端， 他的team是DB端, 怎么交互，设计API，各种细节。

Message Q: producer - consumer model

他刚开始的描述 queue，producer，consumer我第一反应是单机的多线程p/c问题，其实这个queue，p，c是分布在三台不同的机器上的。。。扯了半天才发现是这样

看一波kafka 原理。。

给你一些producer，发送不同的image url到你的service，你的service需要有consumer，来读取image的url，然后把image download下来。其实就是想让你design 一个mesasge queue。follow up有咱们scale，consumer machine down掉了怎么办，如果message queue down掉了怎么办，down掉后重新开启有什么问题。最开始还让写了一下service的各个API。
体会就是平时看系统设计有关的文章或者教程的时候很注重计算qps，storage size之类的东西，其实面试官不太关心这个。

queue里面每个节点需要哪些内容？如果message queue这个service down了怎么办？consumer花太多时间working on one task怎么办？consumer掉线了怎么办？

设计内存queue。面经里没有这题。但我很久以前看过activemq之类相关文章，只好现场发挥。被问了很多。什么情况use case下用这种设计？生产者发送消息失败怎么办？消费者读了msg，还没处理，就断电关机怎么办？生产者怎么知道消息已经被消费者读取了？数据库里要存什么数据？怎样计算需要多少生产者和消费者？

问了些producer这边的REST怎么设计？简单的POST，{domainName}/queue/{imageId}
status code应该用什么？400,403,404 客户端错误/ 200 请求成功 / 500 服务器错误
queue里面的message应该包含什么信息？imageId，jobId，metadata，timestamp
如果message没有放到queue里怎么办 ？return 500，log
consumer如果处理不了message怎么办？ log，discard，retry
如果超时怎么办？ kill thread
怎么监控consumer的状态？ expose thread info 到REST？ 记录thread pickup job 的timestamp
consumer会遇见什么问题？ message 不valid没法parse，处理的时候image可能已经不存在了
consumer挂了，没处理完成的图片怎么办？producer放message到queue的时候要log这个Event，如果挂了对比producer/consumer的log找没处理的图片
queue挂了怎么办？ HA，做replication，producer log message
多台机器的consumer要考虑什么？

面试官的问题问的非常模糊，好久没面这种design了，自己把自己绕里面了，感觉说了很多东西他都不太满意的感觉，想了想主要原因是他一直在drive这个设计，如果自己设计的时候就把这些问题抛出来解决的话可能会好一些

设计log系统，

问api咋搞，该不该带timestamp，为啥
deliverable可以有哪几种，优缺点。
选了某一种以后的数据存disk 还是存in mem，为啥，优缺点
读写需求等等等

```
{action: "view",
  page:   "www.dropbox.com",
  location: "GB",
  time stamp: xxxx}
 {action: "download",
  file: "hello.txt"
```

file: "hello.txt",
time stamp: xxxx}
在log里面，一定会有action和time stamp，然后其他的不分根据action是什么来决定，你会怎么设计interface来create log，需要哪些function，之后就是讨论moniter service和web server这两个不分怎么交互，怎么存log，会有什么bottleneck ，怎么样improve，怎么样scale，然后又问了如果让设计个metrics来监测你的monitor service的健康状况，你会收集哪些数据。大概就是这些。

此轮要design的是丢盒子内部使用的logging system。此轮觉得interviewer问题描述的不是很清楚，因为楼主自己是做storage的，说完系统需求之后，楼主开始从log的结构开始设计。可是interviewer并不在意这些细节，也不在意log如何被读出来，interviewer要求楼主把这个系统的high level diagram画出来，然后问楼主某个component坏了的话需要怎么做

系统设计，一个比较老资格三哥，问了我设计dropbox 的sync pipeline，多个client和一个server怎么样sync。从最简单的案例开始（没有conflict），如何添加文件，修改文件，api 该如何定义，push 到server 端后，server 端怎么存，存什么，数据结构长什么样，其他客户端怎么样pull。我总体就是按照git的思路来答的，总体来说三哥全程都笑呵呵的，所以不知道三哥是满意还是不满意

>> Coding

- leetcode 17
- 289 生命游戏 + followup 数据量特别大
- 362, counter
- 求pow（），递归简单写出，followup要求不用递归
- 609. lc Find Duplicate File in System。give api read very large size FILE
- 给二维数组代表海拔高度，找出最高海平面，使得存在从最左列到最右列的路径，每一步往右上 / 中 / 下走。DP。follow up：如果二维数组无限大? 可以每次处理两列。当前值 = 前一列上中下里的最大值，和当前值取min。然后在最后一列中找到最大海拔。
- **Token Bucket**/Leaky Bucket
- **web crawler**： single process -> multiple process
- Find Byte Pattern in a file - Rolling Hash https://leetcode.com/problems/repeated-dna-sequences/discuss/142300/Python-rolling-hash
- 合并文件, 检查下载有没有完成那题。
- implement read/write lock

transactional KV-Store (coding)

就是要实现 get, add, update and delete方法，给了start()和commit()方法开始和结束transaction. 基本思路就是用一个hashmap track transactional id as key，考的重点应该是有多个transaction同时更新一个值的时候怎么rollback

这道题我的情况是不涉及多线程，开始便问了小哥两次这个问题。
什么时候abort，可以先从最基本的开始，两个transaction对同一个K操作的时候，abort其中一个。. check 1point3acres for more.
至于优化，到时候可以跟interviewer具体讨论讨论，因该某些情况可以避免abort的

k 保证的是concurrent transaction不会出现race condition。题目说了不会出现concurrent transaction，但是不同的transaction commit之前会touch同一样的id，所以说是需要另外一个map或者其他data structure而不是lock
---
就是要求实现一个in memory的map支持transaction，begin返回一个transactionID，实现put(transacId,

k, v) get(transactId, k ,v)以及commit，需要考虑多个transaction交错的情况以及abort. F

Token Bucket

令牌桶例子：从0秒计时，桶内初始有10个令牌，每秒有10个令牌被放入桶内。问5秒时，可否取出60个令牌？桶内还剩多少令牌？
问10秒时，可否取1000个令牌？
*不要10个10个的填加令牌，记录起始和终止时间。令牌总数则是 (time_end - time_start) * rate

生命游戏 how are yours iiiikilkoloiloiloiii9

Follow up
”如果输入是million * million“
* 3*3，这样频繁改动文件指针很慢，不好，咋办？
* Read memory-size block by block 注意要按整行读入，所以需要更新文件指针

那后面的block的第一行要依赖前面block的最后一行咋办？ calculate state of each cell in block save state of last line to calculate next block
2. “我知道你可能不熟悉文件读写，你能不能试着把你的思路写下来（不是伪代码）”
* 憎逼 硬着头皮写呗 假定memory size 新建两个文件指针。。。。小哥说nice

3. “那么如果你现在需要调用之前写的100*100输入的算法，你怎么改变你的输入”
* 边读文件边创建矩阵，用list可以动态分配空间 小哥说cool，那你实现一下...

follow up问题：如果输入是个100万x100万怎么办？一次读三行，更新数组。
怎样修改辅助的method，使其总是返回三行？如果是开头第一行，或最后一行，外加两行全0的
效率瓶颈在哪里？读文件
怎样提高效率？不要重复读同一行，缓存一下已经读过的行

Allocation ID

---
write a ID generator, supports
- Constructor (given the maximum ID number)
- alloc() return an unused ID 0-max
- release(ID)

---

coding 就是经典的Allocate ID. queue + set ->bitset -> segment tree + bitset ，最后小哥还说可以用时间换空间来优化一点，把二叉树变成多叉树

给一个max，实现两个Method，int allocate()，int deallocate(int id). allocate（）： 要返回一个0到max - 1没有用过的值（已经使用过的值不能再次被使用）， deallocate(id)： 让id重新可以使用。 这个算是高频题， 从刚开始 HashSet 优化到 BitSet， 然后优化到Binary tree + BitSet.

queue，queue+hashmap，queue+BitSet，BitSet+segment tree，然后需要计算每种方法需要多少byte的内存。

interviewer提到ptr也会有overhead，所以才会有了后来的circular array做法

---
alloc id. HashTable + Queue 写了一个简单实现，然后分析这个解法哪里不好，计算了一下memory usage。然后说可以用BitSet来记录每个Id的使用情况，这样可以把内存使用降低到1/32。 简单聊了一下BitSet的实现原理，分析了时间复杂度是O(n)。没有实现，然后继续讨论了优化可可能性，给出了Segment Tree的解法。面试官说suppose给你的max capacity永远是2的N次，帮我降低实现难度，让我实现了一下。中间出了点小bug，被面试官发现了，很快

修止了。做完以后还有10分钟左右，面试官问有没有一个解法可以做到O constant的复杂度，不是O(1)。 想了一会儿没想出来，只想到了从HashMap的角度出发，时间不多了就结束了，最后面试官简单说了一下思路，没太搞明白，但是确实是HashMap的方向。

---
alloc id 我onsite也遇到了， 最后的最优解法 是建立一个2n个bit的 segment tree 然后o（lg n）的复杂度完成所有操作

setTimeout

类似DelayQueue + producer consumer

就是要实现一个setTimeout 的callback 方法，从主线程可以多次call 这个方法，这个方法会在等待一定时间后执行。应该是个多线程的题，如果等待时间过长，怎么能减少内存损耗，怎么能不conflict。

Photo max count

你有一堆photo_id，id每出现一次表示一次访问，返回访问次数最多的k个photo_id 2. 假设输入是stream，设计一个类，实现get_current_top_photos方法，返回当前访问最多的k个图片

file access

```
A <-- explicit access
|___ /B
|     |___ /C <-- explicit access
|     |___ /D -- /k access
        | ---
|___ /E <-- access
       |___ /F
```

1) check if the user has the access to folder
2) 给某个folder加一个属性，表示这个folder的access不能被subfolder继承，修改代码
面到这里面试官说他可以给我通过了，接下来的时间我可以问他问题，也可以继续做bonus题目，我只能说继续bonus题目了。。。

3) bonus: remove redundant access
我的算法是从accessSet 里选元素往下dfs遍历，面试官说标准答案是从root开始往下dfs，但是我的解法比标准答案更好一些，然后让我问了几个问题就结束了

KV Store

KV Store这道题的问题是设计一个transaction， 有一个start() 方法， 返回一个transaction id. 有一个put(transactionId, String key, int value), 有一个get(transactionId, String key), 和一个commit(transactionID)。 要理解这道题到底什么意思， 首先得先翻翻Database的书， 看下transaction那一章， transaction的四个属性ACID。 主要是Isolation level。 transaction有四个level， Read Uncommitted, Read Committed, Repeatable Read和Serializable。 根据面试官的要求， 你要实现其中的一个level， 比如面试官如果说这个transaction会用在bank system里面， 那么最好就是实现Repeatable Read那个level。 因为这个level可以避免dirty reads, non-repeatable reads, and lost updates。想象下如果有两个transaction以下面这个顺序进行读写(假设a之前的值为1)

```
start()  // start transaction 1
start() // start transaction 2
int val1 = get(1, "a");
```

```
int val2 = get(2, "a");
put(1, "a", val1+1);
put(2, "a", val2+1);
commit(1);
commit(2);
```

那么transaction 1的操作就会overwritten， 这个就是update lost。
解决update lost就需要实现repeatable read， 意思就是当一个transaction得到一个key的读的lock时， 要一直hold这个lock到这个transaction结束为止。 这样一来当例子中的第二个transaction要读a的时候就会被拒绝。根据面试官的要求， 一般会直接throw一个error然后把第二个transaction取消掉（rollback之前所有已做过的操作）。 还有一点要注意的是， 这其实是一个单线程题， 所以不需要考虑多线程， 比如上一个例子中所有code都是有时间先后顺序的， 因为他们都发生在同一个thread里。 至于在实现锁的这块， 可以使用Map来记录当前哪些key已经有读的锁， 哪些有写的锁。 如果一个key已经有读的锁， 那么其他transaction只能获得读的锁， 如果一key已经有写的锁， 那么其他transaction不能再获得读或写的锁。 还有一点要注意的是， 一个key上的锁如果只有一个transaction并且此时要求锁的那个transaction就是holding锁的那个transaction， 那么这个transaction的读或写应该被允许。 如果一个transaction之前改变了一个值， 在后来的操作发现有conflict， 那么要把这个transaction之前修改过的值都改回原先值。
比如说有这些操作，

```
start()  // start transaction 1
start() // start transaction 2
int val1 = get(1, "a");
put(2, "b", 2);
put(2, "a", 2);
commit(1);
commit(2);
```

当transaction 2 改b的值时， b的值可以被成功修改， 但是当transaction 2 修改a的值时，因为此时a的lock已经被transaction 1 hold， 所以这里有一个conflict， 所以transaction 2要被cancel掉。这里可以用一个Map来记录一个key和它原先的值， 每一个transaction都有一个这样的map。 当发现一个transaction需要被cancel时， 把这个transaction之前改过的所有key都要恢复原来的值。这道题的基本就是这个意思了。面试的时候要跟面试官讨论到底要实现那种isolation level。 isolation level越高越复杂。

File full path
https://leetcode.com/problems/find-duplicate-file-in-system/description/

给一些档名,但有些内容相同,找出这些文档

```
/foo
  |---/bar
       |---/test.png
  |---/img.png
  |--/test
      |---/test2
          |---/test.txt
  |--blabla.txt
```

假设 /bar/test.png and /img.png 有相同内容
/test/test2/test.txt and blabla.txt 有相同内容
输出
[[/bar/test.png, /img.png], [/test/test2/test.txt, /blabla.txt]]

Give a directory name (like /user/foo), return a list of files, in full path, under it.
For example, we have

```
/user/foo/fil1
            /bar
                  /file11
                  /file12
                  /xyz
                        /file111
                        /file112
                        /file113
For the retured list, identfiy which two files are identical.


1. hash是怎么算的 -> whole file hash + sample chunk hash
  2. realworld 大概算一遍多久 -> guess 150gb 10000files -> seconds to mins
  3. 如何speedup -> use metadata
  4. metadata里没hash -> 先metadata 大概20mb 晒完了再第二部看文件 大概0.1% -> 10files ->
seconds

还问了symlink怎么办。
就是读一下link就好啦，面试官说不会有circle的。

其他面经有提出只hash文件前k bytes的内容以及查metadata这几个optimization，但不知道楼主是不是还有更进一步的follow up

需要先沟通，假设取folder， 和file， 都已经有API了。 在这上面浪费了时间，不知道如何读folder和file。

md5sum做一个hash，然后再用文件的大小作为另外一个check。hash作为一个dict的key，里面存一个list放文件的目录，
基本这样的数据结构 unordered_map<long, pair<string, int>>我觉得就可以了。拿到文件先做一个md5sum的hash，或者反过来拿文件的size做key也行，
关键得问清楚同样size的文件会不会很多。基本上hash加上文件size可以unique确定一个文件是否相同。如果不保险每个bit都比一下。

--------

最好按照size filter-> sample hash filter-> full content hash filter -> full content compare这么处理，效率会高一点，而且对大文件也适用。
public List<Set<String>> findDuplicateFiles(String root) {...}

hmm 其实你不用搞这么复杂，你只需要两个hashmap， 一个是size和对应的文件list，一个是md5 hash和对应的文件list。首先读所有文件的metadata，把相同size的文件丢到对应的list中，然后只处理list size > 1的entry，一行行地读文件内容，生成md5 hash，然后hash不一样的就又自成一个entry，直到读完所有的文件。其实每一轮只需要继续读size > 2的list中的文件，所以比较快。简而言之就是size不一样的一定不在同一个集合中，sample hash值不一样的一定不在同一个集合中，这些都可以过滤掉，不用花时间再去读文件。最坏情况就是所有的大文件都duplicate，那么只能所有文件都得读一遍
--------------
1）如何确定文件类型？遍历目录文件时候，如何确定这是个regular文件，是一个符号链接，是一个特殊的设备文件，还是目录？这是在遍历时候的一个必要步骤。不同的语言和操作系统下可能有不同的方法。如果能够谈到linux下文件系统下有多种文件类型，Linux提供了多种API，比如is_regular_file(), is_block_file等保证确定文件w类型，应该是加分项。
(2）如何确定两个文件相同？所有文件对每个文件跟其它文件逐个byte比较是一个很直接了当的办法。但这样非常不高效。改进的办法应该是对文件产生指纹，比如用MD5，或者SHA256等hash算法来产生指纹。这样在指纹库里面进行匹配查找就容易多了。
(3）但MD5等hash算法依然存在着冲突的可能性，也就是说，两个文件可能有同样的MD5值。怎么解决这个问题？MD5值相同的情况下，再对两个文件进行逐个byte比较。
(4）都是大文件怎么优化这个重复文件检测？重复文件意味着两个大文件的每一个块都是相同的。可以将文件进行切块，对每个块生成MD5，这样大文件的比较就是比较每个文件的MD5列表。只有两个MD5列表完全一致，才说明两个文件完全相同。但在比较过程中，只要遇到不同的MD5，就可以跳出检测，节省时间。
(5）如果文件数量非常多怎么办？这个考虑两点优化：第一点要对MD5指纹库查询并行化，首先是单机内多线程的查询，
```

文件规模巨大的情况下，就是多机分布式查询。这个涉及到system design方面了。第二点是考虑文件自身的特性。文件大小相同才有可能是重复文件。所以在查询时，首先找到系统中其他相同大小的文件，然后只跟这些个文件的MD5列表对比，这样就缩小了查询范围。另外，对于一些小文件，是否可以有特殊的处理。比如空文件。比如文件特别小，那是否可以不产生MD5，直接进行byte比较。这个是开放性的提问，只要保证回答合理就行。
(6) 如果目录里面有soft symbol link怎么办? 这个是要考虑是否出现死循环的情况。所以如果提到了第一个问题的解答，在这里就可以表明如果是symbol link就直接跳过。
(7) 如果目录很深怎么办? 这个需要考虑，用来存储文件路径名的字符数组是否有溢出的可能性。
(8) 如果重复文件查询过程中死机了怎么办? 第一点可以考虑checkpointing，定期保存进度。这样重新启动查询的时候，不需要从零开始。

NASA

```
import java.io.*;
import java.util.*;
. 牛人云集,一亩三分地
/**. 留学申请论坛-一亩三分地
* NASA selects Dropbox as its official partner, and we're tasked with managing
* a panorama for the universe. The Hubble telescope (or some other voyager we . Waral 博客有更多文章,
* have out there) will occasionally snap a photo of a sector of the universe, . from: 1point3acres
* and transmit it to us. You are to help write a data structure to manage this.
* For the purpose of this problem, assume that the observable universe has been
* divided into 2D sectors. Sectors are indexed by x- and y-coordinates.
*/
public File {
    public File(String path) {}
    public Boolean exists() {}
    public byte[] read() {}
    public void write(bytes[] bytes) {}
}
public Image {
    public Image(byte[] bytes) {}
    byte[] getBytes() {} // no more than 1MB in size. more info on 1point3acres
}
.本文原创自1point3acres论坛
public Sector { 来源一亩.三分地论坛.
    public Sector(int x, int y) {}
    int getX() {}
    int getY() {}
}
/**
* row-major indexing to be consistent.
*/
public class SpacePanorama {
    /**
     * initializes the data structure. rows x cols is the sector layout.
     * width, height can be as large as 1K each.
     */
    public SpacePanorama(int rows, int cols) {}
. 围观我们[url=home.php?mod=space&uid=306102]@1point[/url] 3 acres    /**
     * The Hubble will occasionally call this (via some radio wave communication)
```

```
     * to report new imagery for the sector at (y, x)
     * Images can be up to 1MB in size.
     */
    public void update(int y, int x, Image image) {}
    /**
     * NASA will occasionally call this to check the view of a particular sector.
     */
    public Image fetch(int y, int x) {}
    /**
     * return the 2D index of the sector that has the stalest data.
     * the idea is that this may help the telescope decide where to aim next.
     */
    public Sector getStalestSector() {}
}
```

题目是什么背景有点记不是太清楚了，大概就是有一个row*col的矩阵（row，col的值可以up to 1000），然后每一个点上有一张image（size up to 1M），让实现两个函数，一个是update(int row, int col, Image image), 另一个是fetch(int row, int col)，楼主一开始没有完全理解要求，说直接用一个二维矩阵去存，后来面试官提示说你仔细想想，这个是memory存不下的，让换个方法。后来我提出说用一个key value storage（some nosql solution）来存，但是限于电面是在coderpad，我就说用一个hashmap来代替这个key value storage吧，面试官说可以。主要就是要用row，col生成一个unique key作为键值，value当然就是这个图片了。

follow up问题是要求返回一个stalest的image，就是说最长没有被update的image的row和col，如果某个（row，col）从来没有被update过，那个它应该就是最stalest的。. 1point3acres

接口大概如下：

```
class Solution {
  public Solution(numRows, numCols) [
  }
  public void update(int row, int col, Image image) {
  }
  public Image fetch(int row, int col) {
  }
  public int[] getStalest(){
  }
}
```

ollowup1: 返回最不常更新的那个位置
followup2: 如何做分布式存储

没怎么听懂题意。
大概就是有个nasa 望远镜系统跟他们合作。然后要按坐标存取image

1.给了几个api，update,fetch啥的。参数分别是row,col(算是坐标吧)
我一开始开了个二维数组。他说如果文件太大的话。很废空间。
2.然后又给了个api，write,read 到file
类似于序列化吧

然后就改成类似Image.read(new (File(row + "," + col))); 实现fetch,
update函数也同理，就是用write. check 1point3acres for more.
3.最后又新给了一个函数让实现，就是我没听懂的

3.最后又新给了一个函数让实现，就是找没听懂的
如何update stalest sector
大概就是如何更新最久没访问过的sector
我觉得类似LRU吧，然后写了个linkedHashMap
每次把队尾的node拉到最前，就是更新了。但他问我有啥潜在的问题

highest minimum sharpness

地面的题目是老题 ，就是二维dp解决的。写的时候有一个小bug，bp[r-1][c-1]写成了bp[r-1][c]，bp[r+1][c-1]写成了bp[r+1][c]。运行程序以后发现了这个bug，然后改过来了。然后面试小哥说可以优化空间，我说用一维dp保存前一列数据，没有要求实现code。

```
// Given a 2-d array of "sharpness" values. Find a path from the leftmost column to the rightmost column which has the highest minimum sharpness.

// Output the highest minimum sharpness. Each move can only move to the top right, right or bottom right grid.

// Example: 3*3 matrix

// 5 7 2

// 7 5 8

// 9 1 5

// The path with highest minimum sharpness is 7-->7-->8, because 7 is the highest minimum value in all the paths.. check 1point3acres for more.

// Idea: Use DP dp[r][c] = min(max(dp[r-1][c-1], dp[r][c-1], dp[r+1][c-1]), grid[r][c])


follow up

For a very large board we could rotate the board clockwise by 90 degrees. Then process line by line.
To analyze the rotate strategy let's simplify the board to N*N. If we read a row and write to N rows
we have N disk seeks for reads and N*N disk seeks for writes. If we read a col and write to a row we
have N*N disk seeks for read and N disk seeks for write. We could balance the two. Say the mem can
fit K*K cells. We could read K rows of length K then rotate in mem then write to K rows of length K.
For this operation we have K disk seeks for read and K disk seeks for write. There are `N*N/(K*K)`
blocks so in total we have `N*N/K` disk seeks for read and `N*N/K` disk seeks for write.


-------

followup和之前的面经一样，问的是如果是100million * 100 million怎么办。因为看过面经，我先回答的是答案是把这个matrix翻转90度，然后一行行处理，但翻转的时候，读行输出列会有硬盘写文件耗时，读列输出行会有硬盘读文件耗时。

然后又回答说可以读一个正方形，一个正方形一个正方形处理。小哥让把code写一下，我就写了一段pseudocode。

然后小哥给分析了下发现这样有问题。如果处理matrix中间5x5矩阵，已知第一列中的五个值X，第二列只能算出来中间
```