Dropbox上门面经--湾区最佳食堂之旅

<https://www.1point3acres.com/bbs/thread-546698-1-1.html>

因为他家我主要是冲着体验食堂面的之前比较忙就没花时间准备，面试前一天才开始搜了下面经然后在飞机上现啃的线段树代码。结果发现他家题库真心很小碰到的基本都是前一天的原题现学现卖。  
第一轮系统设计，设计message queue的读取的API，这个建议看一看SQS的implementation，细节问得比较多也比较细，得面试官多交流use case和tradeoff。  
第二轮算法题，ID allocator，就是[刷题](http://www.1point3acres.com/bbs/forum-84-1.html" \t "_blank)网叁汽酒，这题其实我个人认为用hashset iterator去做的话时间已经很好的，但是感觉面试官就是想要看segment tree的解法（尽管时间并没有更优）我也很无奈。最后就先让我写了只用BitSet O(n)的解法然后优化成Segment Tree的解法。具体代码可以看这里：[https://leetcode.com/problems/de ... gment-tree-solution](https://leetcode.com/problems/design-phone-directory/discuss/119330/My-segment-tree-solution" \t "_blank) 之前如果不熟悉线段树的话写起来很容易会有Bug，可以先看看这个线段树基础知识讲解：[https://www.hackerearth.com/prac ... ent-trees/tutorial/](https://www.hackerearth.com/practice/data-structures/advanced-data-structures/segment-trees/tutorial/" \t "_blank) 然后面试官让我画出具体的树一步一步的go through代码。这里我自己作死画了个四层的树想着recursion好解释结果发现每一层绕的算mid， left， right和node index的时候差点没把自己绕进去。强烈建议画一棵，小，一，点，的，树。  
第三轮是90分钟的pair programming。这轮就是模仿实际工作环境和面试官一起在pycharm上完成一个project的几个task。我之前有说我一直写java没怎么用python，recruiter说这轮不需要python基础碰到语法不懂的可以让面试官帮忙打主要看解题思路，如果实在不comfortable也可以换成一轮java白板。我最后懒得折腾就没让他们换。我这轮是他们有一个queue service的API可以用，让实现一些async的功能去减少latency，然后写unit‍‌‌‍‍‍‌‌‌‌‌‌‍‍ test在local host测试commit代码。很多时候语言不熟悉可以跟面试官说思路她帮着打python   
第四轮是BQ轮，最自豪的项目，跨组合作的项目，和组员意见不一样的时候，项目DDL赶不上的时候，给别人provide feedback和自己收到hard feedback的例子都想一些就好。

Dropbox SF 新鲜昂赛

<https://www.1point3acres.com/bbs/thread-545571-1-1.html>

整体感觉他家的题库真的挺小。。。看地里面经  
一共五轮面试  
第一轮project deep dive  
第二轮Coding web crawler，BFS然后多线程 我是按这个思路写的 [https://www.1point3acres.com/bbs ... adio%26sortid%3D311](https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=277537&extra=page%3D2%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D25%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311)  
然后午饭，在新的HQ，两层楼的餐厅，4-5个不同的stand，每个都有不同的cusine，每餐都有甜品，品种真的多的没得说，presentation也很漂亮，但我实在是没啥胃口。。。可能面试太紧张了，硬逼着自己吃完的，所以感觉味道和之前面过的NYC FB/Google食堂也没太大差别  
第三轮All around也就是他们的behavioral面试 一个engineering manager来面 让我讲一个项目经历 问了一堆问题例如：有没有什么不好相处的同事 如果这个项目让你重新做会怎么改动 在之前公司除了日常工作有没有担任别的角色（面试官，mentor，组织social之类的）  
第四轮Architecture居然没问scalability问了API design，万万没想到。说给你一个Mobile App，像iphone的Contact电话本那样的，让你设计这个APP的backend service的API。我就写了CRUD + getAll (paginated) followup又问如果有头像怎么处理 还有大小两种头像 thumbnail和regular size  
第五轮coding allocate ID 我按着这个写的 大概先写了个LinkedHashSet的解法，然后最后假装在面试官的引导下，写了segment tree + bitmap的solution  
[https://www.1point3acres.com/bbs ... adio%26sortid%3D311](https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=213126&extra=page%3D2%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3046%5D%5Bvalue%5D%3D25%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311)  
第六轮是三十分钟的product demo  
  
第七轮是和hiring ‍‌‌‍‍‍‌‌‌‌‌‌‍‍manager聊三十分钟有关product roadmap之类的

丢盒子onsite

<https://www.1point3acres.com/bbs/thread-541559-1-1.html>

一共5轮：第一轮coding design web crawler  第二轮coding Id allocator 第三轮behavior 第四轮project deep dive问了之前做的project，聊了很多细节。 第五轮system design 设计一个文件共享系统，主要focus在后端的数据库设计和处理各种failur‍‌‌‍‍‍‌‌‌‌‌‌‍‍e。 总体感觉不错但最后据feedback挂在systm design上

有可能被烙印黑了的Dropbox7月Onsite跪经

<https://www.1point3acres.com/bbs/thread-541219-1-1.html>

总共6轮，早上11am到5pm，12pm是午餐轮。  
--------------------------------  
11am：behavioral questions  
--------------------------------   
白人妹子，细问工作背景 e.g., 2 projects   
工作时有关“人际关系”的问题  
e.g., did you have experience convincing your TL? did you have to re-do some design? etc.   
-------------------------  
12pm: lunch interview  
-------------------------  
食物好棒！这一轮表现应该不算入面试考核内，建议多问问题。  
-----------------------  
1pm: system design  
-----------------------  
白人男TL，问了设计：  
1. client和queue之间的APIs  
2. server和queue之间的APIs  
他给了个比喻，例如Twitter profile picture upload需要一个queue aysnchronously 处理  
感觉是一道很high-level的问题，需要多问细节，设计一套efficient的APIs。  
-------------------  
2pm: Id allocator  
-------------------  
给1M的size，实现allocate()和free()。  
烙印女，我说见过题目了之后，她态度立即改变，不给时间讲完解释，变得很不耐烦，而且感觉有几次特意误导讨论细节和耽误写代码。  
有3种解法：  
O(1) allocate, O(1) free, O(2\*N) space：如果没记错就是queue + set的那种解法，我用C++，所以是一个vector<int>和vector<bool>。  
O(N) allocate, O(1) free, O(N) space：用bitset就行了。:)  
O(sqrt(N)) allocate, O(1) free, O(N + N/1000) space：bitset + vector<int>，vector<int> has size 1000，每个index记的是count，最多1000，帮助判断某个1000的range (e.g., 0-999, 1000-1999等）是否填满了。  
写了第三种解法，开始说allocate()是log N，最后纠正成 sqrt(N)。。请问地里第三种方式如何？当场想出来的。  
很有可能跪在这一轮。  
-------------------------------------  
3pm - 4:30pm: pair programming  
-------------------------------------  
白人妹子，给几个python files优化cat photo upload server，实现3个任务， server用的是Python Flask。  
这一轮应该是考核平常工作沟通等的技巧。  
建议是不要花太多时间在讨论设计上，应该专注implementation，python不熟悉面试官是会帮忙实现代码的。  
  
被2pm那轮影响了心情，回答得比平常慢，只做成了第一个任务，test codes都没写完。  
所以只记得第一个任务（对不起各位），但面试官表明interviewees通常也都是far from delivering all 3 tasks，但还是推荐尽量至少做到第二个任务。  
  
第一个任务是目前upload photo时都会call 3rd party API to calculate cuteness value，但calculation需要时间，如何优化？  
答：call 3rd party API asynchronously。  
Implementation设计了一个function把callback function带进去，然后那个callback function里调用3rd party API（如果没记错）。之后是跑出来了，但test codes没写完时间就到了。  
也有可能跪在这一轮，但至少面试官很有礼貌和耐心，不像2pm那轮这么惨。  
-----  
总结  
-----  
1. Dropbox氛围不错，建议面onsit‍‌‌‍‍‍‌‌‌‌‌‌‍‍e前准备一些有关公司产品策略(例如how dropbox benefits after collaborating with G-suite and Office)和dropbox system design的问题（例如how chunker works with metadata db)。  
2. 面试时要淡定，尽量避免因为之前有面试官刁难或面得不好间接影响之后的表现。  
3. 遇到烙印不要慌，也不要直接说见过题目，因为ta很有可能会开始挑剔起来或变得不友善。  
  
**补充内容 (2019-8-5 14:22):**  
4:30pm - 5:00pm还有一轮跟某个组的老板（白人男）聊天，他说这一轮就是到我反过来问面试官问题。问了Dropbox的一些技术问题，结束后和HR聊了下就被送出门啦。

丢盒子上门

<https://www.1point3acres.com/bbs/thread-531705-1-1.html>

设计图片上传后生成缩略图的服务  
设计一个系统有开始（无输入，返回交易码），读（输入交易码和键，返回相应值），写（输入交易码，键，值，无返回值），确认服务（输入交易码，返回成功/失败）接口  
午饭  
工作经历项目结合行为测试问题  
挑一个项目讲清楚技术细节，越细越好，做决定背后的权衡，可以提高的地方  
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）  
和招聘经理聊天（组，文化，技术，等）

第一个任务就是把上传任务从同步改成异步，然后通过测试，只需要对网络编程有基本常识，然后和面试官积极沟通就可以做出来，但是动作尽量快，给后两题多留些时间。  第二个任务是基于第一个的异步结果，累计后求平均值，难点在于怎么记录多个异步计算的结果。  第三个任务我没有做完，理论上用一个最大堆就可以。

丢盒子电面+onsite新鲜面经

<https://www.1point3acres.com/bbs/thread-525005-1-1.html>

今天onsite，总共4轮半  
1. design 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。  
2. coding 看起来是韩国人的大叔加中国小妹妹shadow。token buckets + multi-thread。基本解法就很快写完了，然后要加锁，表示我问卷填的不会muti-thread，面试官略有点panic，我说没事咱们顺着走走吧，磕磕绊绊写了比较基本的锁，面试官问了下conditional wakeup之类的，我就顺着他意思说说，最后还比较满意?  
3. 午饭。中国大哥，满nice的，我问的问题都很诚实没有隐瞒的回答了（最后发现我要面的就是他们组）。饭还可以，咖啡非常非常好喝。注意这轮虽然不算分，但是要问一些behavior问题  
4. all rounds。一个很experienced的白人，dive deep projects，要准备两个，一个是比较细节的，要求主要是自己deliver的，另一个是team interactions较多的，问的问题很细。但我个人很不喜欢他，太爱吹逼了，说自己经验丰富，带12个mentee以及非要给我讲time，resources和features的关系etc。我就微笑脸看他装逼  
5. coding。白人姐姐+天竺shadow。问的是id allocator，queue的解法秒了，开始计算时间空间复杂度以及内存占多少。然后开始optimize，用bitarray，一开始给自己写懵逼了，卡了很久很久，她也被我绕进去了，后来她发现我哪里写错了，提示了一下，把代码改了。这轮感觉不好，就是脑子已经不转了，好喝的咖啡效力用光哈哈哈。如果挂估计就这轮。  
6. 半小时hiring manager来聊天+wrap up。问了下team结构，在做啥。愉快的结束了。

哦对，tip3: 可以request macbook写代码，不用白板，也可以各种操作系统的laptop，这个recruiter跟我打电话居然都没有说，我主动提了说要，然后就有人给我准备了一个。用电脑写代码真的好节约时间。

Dropbox onsite

<https://www.1point3acres.com/bbs/thread-524141-1-1.html>

• Design  
• 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.  
  
• Coding 1  
• implement an ID generator that supports 2 methods: alloc() and release(id), alloc() ID from 0 to MAX. how much memory is needed for each solution? what are the runtimes? how to optimize and make tradeoff between runtime and space.  
  
• Coding 2  
• implement a w‍‌‌‍‍‍‌‌‌‌‌‌‍‍eb crawler and then make it multi-threaded and thread-safe  
  
• Deep dive  
\* Behavior

Dropbox 4月onsite 算法面经

<https://www.1point3acres.com/bbs/thread-523051-1-1.html>

上周Onsite面试  
  
第一轮面的是getTokens 面试官没有把问题讲清楚， 理解了很久之后才明白是怎么回事：  
  
假设有一个bucket以S token/的速率在装Token，现在有很多线程要去拿token，设计一个thread-safe的get tokens method. 关键：1. 没有background thread 在装这个token，而是在getToken method里等待多少秒直到有足够那么多token再返回.  
基本上第一问可以这样

class Bucket:

def \_\_init\_\_(self, C, rate):

self.lock = Lock()

self.capacity = C

self.available = C

self.last\_update\_time = 0

self.rate = rate

def getTokens(self, m):

with self.lock:

begin = time.time()

wait\_time = max(0, m - self.available)

time.sleep(wait\_time)

self.last\_update\_time = begin + wait\_time

self.available = min(self.capacity, self.available - m + wait\_time \* self.rate)

第二轮是database atomic transaction design，实现几个methods:  
read(txid, key)  
write(t‍‌‌‍‍‍‌‌‌‌‌‌‍‍xid, key, value)  
createTransaction()  
commit(txid)  
其中有几个关键:  
1. 要支持roll back  
2. 一个txid不能写同一个key到别的txid

丢盒子虾图新鲜面筋

<https://www.1point3acres.com/bbs/thread-518593-1-1.html>

第一轮，写爬虫，单线程到多线程，怎么加锁，怎么终止  
第二轮，id分配问题，先给个naive solution用set，follow up问怎么节省空间，答用树，计算树的总空间是多少byte，发现多了很多额外开销，如何省空间？用bitmap  
第三轮，deep dive，没啥特别的  
第四轮，behavior questions，没啥特别的  
第五轮，系统设计，基本上就是设计一个丢盒子，不同之处是文件类型是视频，用户可以在不同设备上编辑视频，需要云同步。  
最后和面的组的manag‍‌‌‍‍‍‌‌‌‌‌‌‍‍er聊了聊，结束~

丢盒子西雅图昂赛

<https://www.1point3acres.com/bbs/thread-517752-1-1.html>

1. 上来就是面design，design dropbox, 问的非常细， 前台后台， 如何sync 不同的client， version化。  
  2. Leap bucket.  这轮比较有意思， 因为仔细研究过面镜和github上的例子， 做了一个比较优化的办法，面试官似乎有点儿迷惑，解释了很长时间。  
  3. 经验轮， 解释以前的project， 这轮面试官非常牛叉， 懂得很多。  
  4. delay method call， 上机， 做的不是很顺， 花了挺长时间沟通， 感觉他想要很特定的一种实现，但是‍‌‌‍‍‍‌‌‌‌‌‌‍‍不愿意给明显的方向， 所以做了好几种实现然后根据意见又改， 最后五分钟他才说行了。  
  5. bq

丢盒子西雅图onsite

<https://www.1point3acres.com/bbs/thread-517580-1-1.html>

1. KV store  
2. design dropbox  
  3. bq  
  4. allocat‍‌‌‍‍‍‌‌‌‌‌‌‍‍ion id

dbx三番面经

<https://www.1point3acres.com/bbs/thread-507126-1-1.html>

1. merge/delete intervals, 如果delete连续的intervals，怎么做比较好？

2. id allocator，先说tree的方法，后来说要用bit array，考到怎么用cpu cache locality

3. design ring buffer

4. implement read/write lock, 当多读少写，如何优化

这里主要想考太多readers会不会starve writers，我当时就说track pending readers and writers，如果有一个pending writer，就停止acquire新的reader。特别注意写的时候conditional variable不要搞乱。

5. product demo

6.‍‌‌‍‍‍‌‌‌‌‌‌‍‍ manager 闲扯

|  |
| --- |
|  |

丢盒子昂赛

<https://www.1point3acres.com/bbs/thread-502917-1-1.html>

第一轮: NASA Panorama 所有图片都存在disk，如何update和fetch。之后是如何fetch LRU。  
吃饭+带逛: 随便聊天  
Demo: 随便听&提问  
第二轮: Top K View Photo  
第三轮: W‍‌‌‍‍‍‌‌‌‌‌‌‍‍eb Crawler 单线程+多线程版

丢盒子挂经

<https://www.1point3acres.com/bbs/thread-481344-1-1.html>

1 allocid  
2 爬虫  
3 设计log系统, 问的很细节很底层，没有按照系统网上系统设计的套路来什么先给high level啊啥的，直接给我问懵逼

丢盒子昂赛

<https://www.1point3acres.com/bbs/thread-477440-1-1.html>

1. System design: 设计Twitter 主页feeds， daily active user 1M

2. 网页爬虫，多线程讨论  
3. BQ  
4. setTimeout 实现，类似DelayQueue + produ‍‌‌‍‍‍‌‌‌‌‌‌‍‍cer consumer

丢盒子昂赛

<https://www.1point3acres.com/bbs/thread-472597-1-1.html>

1. Token Bucket，地里其他面经有代码  
2. Allocate ID ，先写了queue 的解法，问了一下memory，然后讲到了bitset index tree，代码写完手动测试了一下  
3. BQ  
4. System Design: 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。  
  
面试完第二天过了HR，送到总部hiring comitte，第三天通知跪了。feedback是coding和architecture都行，但‍‌‌‍‍‍‌‌‌‌‌‌‍‍是mentorship和leadership经验不够，没有带过人，没有大型project的经验。明明是SDE II的职位，为什么需要带人的经验...

丢盒子 当面

<https://www.1point3acres.com/bbs/thread-471572-1-1.html>

* 设计一个类似于电话簿的APP， 从设计API, 页面 到怎么scale ，pagination 都要聊一下。
* 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。
* BQ 一个小时，不停地说，口干舌燥的。
* 还有轮，我也不知道算coding 还是system design。 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。
* 最后一轮是和HM 聊天，介绍一下组里和project

丢盒子面经+详解KV Store/Web Crawler/Token Bucket

<https://www.1point3acres.com/bbs/thread-467473-1-1.html>

接下来详细讲解KV Store， Web Crawler和Token Bucket

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。  
  
接下来讲解web crawler  
这道题给了一个get\_links(String initialUrl) 要你写个方法来搜集能从这个initialUrl能够搜到的所有url。最先的方法是BFS， 用一个Set记录已经crawl过的url， 用一个queue来记录还没crawl过的。 这里楼主使用的是BFS， 因为in practice大部分搜索引擎都是用BFS。 用DFS会有若干劣势， 比如1. Takes long time to find high quality pages. 2. make the crawler focus on a few sites  
3. it's easy for web crawler to get stuck in artificially crafted web crawler trap. 然后面试官会问你最花时间的部分是哪， 然后如何改善。 然后你可以说使用多线程。 在多线程的时候， 很多人会用现成的ConcurentHashMap或者LinkedBlockingQueue来代替， 这么写当然没问题。 但是考官想要看到的还是你使用noncurrent的数据结构， 用synchronized block（如果你用Java）把修改数据结构的部分wrap起来， 然后再加上wait()和notifyAll(). 代码如下  
public void crawl() {  
    OUTER\_LOOP: while(true) {  
        String nextUrl;  
        synchronized(this) {  
            while(queue.isEmpty()) {  
                if(workingThreads==0 && queue.isEmpty()) { //<======= exit condition  
                    break;  
                }  
                try {  
                    wait();      
                } catch(InterruptedException e) {  
                    e.printStackTrace();  
                }  
            }  
            nextUrl = queue.poll();  
            workingThreads++;  
        }  
        List<String> urls=getUrls(nextUrl);  
  
        synchronized(this) {  
            for(String newUrl: urls) {  
                if(!visited.contains(newUrl)) {  
                    queue.offer(newUrl);  
                    visited.add(newUrl);  
                }  
            }  
            workingThreads--;  
            notifyAll();  
        }  
    }  
}  
  
这个代码是带有exit条件。 判定是否可以exit， 可以使用一个workingThreads的变量， 当一个thread拿到新的url要call getUrls前， increment workingThreads。 在第二个synchronized block结束前decrement workingThreads. 然后在代码中标记的那行查看是不是没有thread在work并且queue是空的， 如果成立即可break 那个OUTER\_LOOP。 这样所有thread都会顺利exit。 如果没有加exit的代码， 那么所有thread在搜完所有url以后就会停在wait()那行永远在那等待被唤醒。 记住getUrls() 那行一定不能在synchronized block里， 因为那行我们需要它能被多个thread同时执行， 如果放进了synchronized block里也就变成单线程， 就失去了把它写成多线程的意义。

接着讲Token Bucket 。TokenBucket就是让建一个TokenBucket的class， constructor需要传入一个capacity， 就是这个Bucket里面所能容纳的token以及产生token的速率。然后要写一个getTokens(int n)方法， 需要支持多线程。 我之前在网上找到了一个git。大家也可以去搜索一下。 我的版本就是那个git的简化版。代码如下  
public class TokenBucket {  
        int size = 0;  
        int maxCapacity;  
        int tokensPerUnit;  
        int timeInNanos;  
        LocalDateTime lastFillTime;  
  
        TokenBucket(int maxCapacity, int tokensPerUnit, int time, ChronoUnit unit) {  
                this.maxCapacity = maxCapacity;  
                this.tokensPerUnit = tokensPerUnit;  
                timeInNanos = (int) Duration.of(time, unit).toNanos();  
                lastFillTime = LocalDateTime.now();  
                size = 0;  
        }  
  
        public void getTokens(int n) {  
                if(n<0 || n>maxCapacity) {  
                        throw new RuntimeException("You can only take a number of token greater than or equal to 0 and less than or equal to maxCapacity");  
                }  
                int taken = 0;  
  
                while(true) {  
                        synchronized(this) {  
                                refill();  
                                if(size>=1) {  // <====优化后的代码  
                                        size-=1;  
                                        taken++;  
                                        System.out.println("Thread "+Thread.currentThread().getId()+" gets 1 tokens");  
                                }  
                        }  
                        if(taken==n) {  
                                // System.out.println("Thread "+Thread.currentThread().getId()+" gets "+taken+" tokens");  
                                break;  
                        }  
               ‍‌‌‍‍‍‌‌‌‌‌‌‍‍         try{  
                                Thread.sleep(1);          
                        } catch(InterruptedException e) {  
                                e.printStackTrace();  
                        }  
                }  
        }  
  
        public synchronized void put(int n) {  
                size = Math.min(maxCapacity, size+n);  
        }  
  
        private synchronized void refill() {  
                LocalDateTime now = LocalDateTime.now();  
                long durationInNano = Duration.between(lastFillTime, now).toNanos();  
                long tokensToAdd = Math.max(0, tokensPerUnit\*durationInNano/timeInNanos);  
                if(tokensToAdd>0) {  
                        size = Math.min(maxCapacity, (int) (size+tokensToAdd));  
                        lastFillTime = now;  
                }   
        }  
}  
  
大致意思就是每次call getTokens() 方法时， 计算当前时间和上次refill的时间差， 如果算出的时间差\*token产生速率>1的话我们就要更新桶里token数量和lastFillTime。 然后对比getTokens()要取的token数和桶中的token数， 如果足够那就给token， getTokens()方法顺利结束。如果不够， 那么就进入循环一直休息直到有足够token产生。 注意我给出的代码是优化版的。在代码里标注了优化的部分， 这么写的好处是如果有一个thread call getTokens() with a large number， 但是一直有其他thread call getTokens() with a small number， 那么那个thread将会一直处在starving的状态， 因为桶里的token一直被拿走无法达到足够数量。 加入优化的部分后， 每个thread无论要几个token， 每次都给一个直到给足为止。

dropbox onsite

<https://www.1point3acres.com/bbs/thread-467135-1-1.html>

1. allocate Id（一开始给的是circular array的solution，最后小哥把我导到bitmap）  
2. project dive  
3. design a job processing system/queue  
4. 爬虫，先单线程，再多‍‌‌‍‍‍‌‌‌‌‌‌‍‍线程

丢盒子 onsite

<https://www.1point3acres.com/bbs/thread-465051-1-1.html>

- logging design  
   注意要深入  
  
- token bucket  
     
- web crawler  
   单线程，‍‌‌‍‍‍‌‌‌‌‌‌‍‍多线程，terminatiom

丢盒子onsite

<https://www.1point3acres.com/bbs/thread-459414-1-1.html>

都是面经题， AllocateID， WebCrawler和Design Logging System

丢盒子

<https://www.1point3acres.com/bbs/thread-453728-1-1.html>

1. 详细介绍一个自己的项目

2. 多线程网络爬手  
3. 多线程令牌桶。  
4. 设计日志文件系统  
  
    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