|  |  |
| --- | --- |
| 2017/8/21 上午 |  |
| **计划**  1.方法重心转移：解决问题的方法论，要开始使用。【官方文档等--】  2.开始把自己的学习日志，开始用github完全管理。【有重心就有方向】  3.就用文档来编写。每天一个文档。【git管理】 |  |
| 4.**学习重心**要转到**算法学习（数据结构）**上来：  **学习的路径**就是按nowercode的interview experience。提问的来。 | 1.面经文档来写 |

|  |  |
| --- | --- |
| 2017/8/21 下午 |  |
| ~15:40计划1：花半小时看lambda表达式。跑一个demo出来。 |  |
| ~16:40克隆别人的一个python代码来看看，然后，开始分析。提供改进。 |  |
| ~17:30写python爬虫。 |  |

|  |  |
| --- | --- |
| 2017/8/21 晚上 |  |
| 写另外一些人的8.20面试题目。 |  |
| 提交今天的输出文档。 |  |
| 每天坚持写代码。（2道题sql+2道编程，分别用java和python做。） |  |

（真的要知道自己的应该努力的方向才行哦。）

**8.20的面经文档：（按日期排名就可以了。收集，整理好就可以。）**

**面经文档：**

|  |  |  |  |
| --- | --- | --- | --- |
| **8.20的面经文档（**网易Java开发**）** | | | |
| 网易Java开发  1，数组一遍遍历，把所有的零放在数组头部【设两个指针，指向当前0元素和非零元素，交换即可】 2，单链表，一次遍历判断回文数 3，两个字符串A，B，多少步，可以把字符串a转换成字符串b，可以字符旋转，剪切 4，索引类别，创建，为什么快，数据库底层b加树，b加和b树区别 5，jdk7和jdk8的区别 6，怎么设置  jvm参数 | | | |
|  | | | |
| **数组一遍遍历，把所有的零放在数组头部**   1. 【很麻烦的做法】从后面遍历，遇到一个零就放到前面去。计数   若从前面遍历，就把非零的位置记录下来，后面遇上了0，就出队列。然后把这个位置再加入到队列中。 | | | |
| 改进1【空间和时间最佳】：【设两个指针，指向当前0元素和非零元素，交换即可】 | | | |
| 改进2【代码书写最佳】：（如果数组不长的话，空间不计。）建一个相等长度的数组，遍历原数组，只要不等于零就往新数组里加，遍历完毕后往新数组补齐零。 连交换都不需要。 | | | |
| 改进3【只考虑oj输出正确性】：新建的等长数组的确有些浪费空间，节约空间做法（**把所有的零放在数组尾部**，只要一次遍历，碰到0就先不输出，同时记录已经输出的个数，最后将数组的长度减去已经输出的长度，没输出的肯定是0，将这些0输出就行了）。因为我们的oj判定只是将文件简单比对的。所以不用管我们数据是怎么样的。所以不用管数据存储。只用输出就行。  也可以，设一个计数器，记下零元素的个数，遇到非零就输出，遇到0计数器+1,最后按计数器输出0 .... | | | |
| **总结：1.设立一个flag标记。2.设一个计数器count是我们经常的做法。**  [**http://bbs.csdn.net/topics/340177883**](http://bbs.csdn.net/topics/340177883)**讨论。** | | | |
|  | | | |
| **（第一次设计提升路线）提升：1**改用英文搜索==》2改用google引擎+中文==》3改用英文引擎+英文搜索。==》4改用固定网站搜索，比如说stackoverflow。 | （方案，如果英文不好，就先用翻译成英文，再搜索就可以了。这是能力哦。） | | |
| （第一次设计笔记提升路线）。 |  | | |
|  |  | | |
| 2，单链表，一次遍历判断是否为回文结构（是否是回文数） | | | |
| 【解读，考察举一反三的能力，活学活用的能力】：这道题目应该是上一道题的补充，我觉得这就是高明的地方。可能应聘者并不知道第一道题的最好做法，那么面试官会给你提示，帮你解决。那么到了这道题，就是考察你对所学知识的灵活运用能力了。 | | | |
| 【双指针，快慢指针，】 | | |  |
| 【思路1，两次遍历】：空间O(n) 整个链表遍历两边， 开一个栈，第一遍遍历把链表中每个元素push进栈中，这样堆中的元素的pop顺序就是链表的倒序输出；第二遍就开始pop栈中数据，每pop一个数据，就把这个数据跟链表中进行对比，如果相同继续往下走，如果不同返回false。 | | | |
| 【思路2】：空间O(1)，使用快慢指针法，第一步设置一个块指针和一个慢指针，快指针一次走两步，慢指针一次走一步（慢），当快指针下一步为null的时候说明慢指针已经走了一半，这就可以找到中间节点。第二步反转中间链表后面的指针，第三部从头尾向中间扫描，对比每个元素是否相等，如果都相等，则是回文数，否则不是回文数。 | | | |
| 【思路3】：综合两种思路：那么就可以一次遍历就判断出是否是回文。1）开启两个指针，这样一次遍历就可以找到中间位置。2）同时在快指针走到最后之前，慢指针读入的元素就只用push近栈就可以了。3）在快指针走完，慢指针还没走完的时候，我们的操作是从栈中pop元素a，和慢指针接着读到的链表数据进行比对，如果相同就继续往下走，如果不同就返回false。  反思：这道题目如果是数组判断回文，那么就很简单。找到中间位置后，直接从头部和尾部遍历比对就可以了。  http://blog.csdn.net/tangli555/article/details/51384296 | | | |
|  | |  | |
| 两个字符串A，B，多少步，可以把字符串a转换成字符串b，可以字符旋转，剪切 | | | |
| 表意不明： | | | |
|  | | | |
|  | |  | |
| 4，索引类别，创建，为什么快，数据库底层b加树，b加和b树区别 | | http://www.jb51.net/article/49346.htm | |
| 聚簇索引和非聚簇索引 | |  | |
| CREATE INDEX indexName ON mytable(username(length)); | |  | |
| 相当于已经排好序，可以快速定位。 | |  | |
| http://images2015.cnblogs.com/blog/576154/201609/576154-20160907130956629-1833512478.png  如图所示，区别有以下两点：  1. B+树中只有叶子节点会带有指向记录的指针（ROWID），而B树则所有节点都带有，在内部节点出现的索引项不会再出现在叶子节点中。  2. B+树中所有叶子节点都是通过指针连接在一起，而B树不会。    B+树的优点：  1. 非叶子节点不会带上ROWID，这样，一个块中可以容纳更多的索引项，一是可以降低树的高度。二是一个内部节点可以定位更多的叶子节点。  2. 叶子节点之间通过指针来连接，范围扫描将十分简单，而对于B树来说，则需要在叶子节点和内部节点不停的往返移动。    B树的优点：  对于在内部节点的数据，可直接得到，不必根据叶子节点来定位。 | | | |
| http://www.cnblogs.com/ivictor/p/5849061.html | | | |
|  | | | |
|  | | | |
| Jdk7和jdk8的区别： | |  | |
| http://www.cnblogs.com/langtianya/p/3757993.html  jdk的各个版本新特性。  http://www.cnblogs.com/lknny/p/6085050.html | | | |
|  | | | |
|  | | | |
| 6，怎么设置  jvm参数 | | | |
| 【临时修改的话】：如果是单个程序，那么就可以用run as =》【runConfigeration】=》【在arguments中输入。】 | | | |
| 【如果是整个环境的话。】  在左侧导航栏中依次点击【Java】-->【Installed JREs】🡪【双击一下安装的jre就可以】🡪找到【Default VM arguments】，输入“-Xms128m -Xmx512m”后点击右下角的【Finish】， | | | |
| 【如果是tomcat中修改的话】是在catalina配置文件中修改。 | | | |
| http://jingyan.baidu.com/article/624e7459653ca534e8ba5a26.html | | | |
|  | | | |
|  | | | |

[数据库中自然连接与内连接的区别](http://hongtaq.iteye.com/blog/1530954)

2017.8.21第一次用英语搜索来解决的问题。使用的bing中输入英文，就搞定了。这样搜索到的答案就是，我们看到的。

<https://stackoverflow.com/questions/8696383/difference-between-natural-join-and-inner-join>

其实只用写好一些基本的单词就可以了。 Sql inner join and nature join difference。这样多好，还复习了英语，并且这里的答案来自全世界。都是比较好的答案。而且你能看懂啊，以后一定都这样来搜索。不要用中文了。成功表述为：what’s the difference between the inner join and nature join。这个非常的好。

从此改变自己对这个的用法。

总结，学习一门语言的正确姿态应该是看别人的代码，看不懂的地方，就去查官方文档，就这样学习。比如看python爬虫的东西，别人用到的基本方法。看不懂就查文档就好。

你会发现国外会把一个评分很高的回答放在前面（大多含有例子说明）。二而国内的回答，普遍很片面，仅仅文字罗列。

|  |  |
| --- | --- |
| 注意on与where有什么区别，两个表连接时用on，在使用left  jion时，on和where条件的区别如下：  1、  on条件是在生成临时表时使用的条件，它不管on中的条件是否为真，都会返回左边表中的记录。  2、where条件是在临时表生成好后，再对临时表进行过滤的条件。这时已经没有left  join的含义（必须返回左边表的记录）了，条件不为真的就全部过滤掉。 | |
|  |  |

开始第二轮面经。这个非常重要。你要知道的，重中之重就在这里。不是简简单单的两个月就搞定的东西。别人华南理工已经关注了nowcoder一年多，也还是不熟悉。所以说明，你如果要脱颖而出，那么你就要别人人多做准备。

|  |
| --- |
| 华工 tencent 头条 |
| **腾讯一面**  **网络**   1. TCP 四次挥手过程  |  | | --- | | <http://blog.csdn.net/omnispace/article/details/52701752>  **4次挥手过程详解**  三次握手耳熟能详，四次挥手估计就少有人知道了。所谓四次挥手（Four-Way Wavehand）即终止TCP连接，就是指断开一个TCP连接时，需要客户端和服务端总共发送4个包以确认连接的断开。在socket编程中，这一过程由客户端或服务端任一方执行close来触发，整个流程如下图所示：   https://static.oschina.net/uploads/space/2016/0926/165553_emXw_737747.png  由于TCP连接时全双工的，因此，每个方向都必须要单独进行关闭，这一原则是当一方完成数据发送任务后，发送一个FIN来终止这一方向的连接，收到一个FIN只是意味着这一方向上没有数据流动了，即不会再收到数据了，但是在这个TCP连接上仍然能够发送数据，直到这一方向也发送了FIN。首先进行关闭的一方将执行主动关闭，而另一方则执行被动关闭，上图描述的即是如此。   * 第一次挥手： Client发送一个FIN，用来关闭Client到Server的数据传送，Client进入FIN\_WAIT\_1状态。 * 第二次挥手： Server收到FIN后，发送一个ACK给Client，确认序号为收到序号+1（与SYN相同，一个FIN占用一个序号），Server进入CLOSE\_WAIT状态。 * 第三次挥手： Server发送一个FIN，用来关闭Server到Client的数据传送，Server进入LAST\_ACK状态。 * 第四次挥手： Client收到FIN后，Client进入TIME\_WAIT状态，接着发送一个ACK给Server，确认序号为收到序号+1，Server进入CLOSED状态，完成四次挥手。 * (1) 三次握手是什么或者流程？四次握手呢？答案前面分析就是。 * (2) 为什么建立连接是三次握手，而关闭连接却是四次挥手呢？、这是因为服务端在LISTEN状态下，收到建立连接请求的SYN报文后，把ACK和SYN放在一个报文里发送给客户端。而关闭连接时，当收到对方的FIN报文时，仅仅表示对方不再发送数据了但是还能接收数据，己方也未必全部数据都发送给对方了，所以己方可以立即close，也可以发送一些数据给对方后，再发送FIN报文给对方来表示同意现在关闭连接，因此，己方ACK和FIN一般都会分开发送。 |  1. TIME\_WAIT 等待两个 MSL 作用（两个）  |  | | --- | | TIMEWAIT状态也称为**2MSL等待状态**。  可靠实现全连接的终止。 每个具体TCP实现必须选择一个报文段最大生存时间MSL(Maximum Segment Lifetime)。 它是任何报文段被丢弃前在网络内的最长时间。  我们知道这个时间是有限的，因为TCP报文段以IP数据报在网络内传输，而IP数据报则有限制其生存时间的TTL字段。  这样可让TCP再次发送最后的ACK以防这个ACK丢失(另一端超时并重发最后的FIN)。 这种2MSL等待的另一个结果是这个TCP连接在2MSL等待期间，定义这个连接的插口(客户的IP地址和端口号，服务器的IP地址和端口号)不能再被使用。 这个连接只能在2MSL结束后才能再被使用。 |  1. 如果在 TIME\_WAIT等待时客户端手动指定该端口再次发起连接，那么可以建立连接吗？（没懂这道题，考察端口复用？） 2. 客户端收到一个窗口为 0 的包怎么处理 3. 讲一下 TCP 拥塞控制算法 4. 是否用过 select、epoll，什么是IO多路复用，epoll 水平触发和边缘触发 5. 给定一个图片 URL，讲一下获取这张图片的整个过程   **算法：**  LeetCode原题：设计一个最优的算法，[求一个集合的所有可能子集](https://leetcode.com/problems/subsets/description/" \t "_blank)  **操作系统：**   1. C++ 进程内存空间（不懂。。讲了 Java 内存区域） 2. 为什么要设计进程 后面告诉我可以从进程演化历史、批处理、多道程序设计、进程间通信考虑   **Linux：**  工具使用：lsof、tcpdump、gdb、Nginx  **腾讯二面**   1. TCP 三次握手，服务端回复时可以将 ACK 和SYN分开发送吗（考察四次握手？） 2. B树叶子节点个数、非叶子节点子树个数 3. 二叉树查找复杂度、最差时间复杂度、最好时间复杂度 4. Linux 如何抓包 5. 合并两个有序链表的时间复杂度（没听清合并还是归并。。） 6. 多进程、多线程什么区别 7. 怎么优化一条 sql 语句   **头条一面**   1. 剑指offer面试题 42：翻转单词顺序 2. 单机版 Top K：给 N 个数，找出第 K 大的数（快排的 partition，最小堆），如果用堆的话，算法时间复杂度；海量数据（不能在单机完成）：分布式 Top K？（面试官说考察 map-reduce 思想） 3. TCP 与 UDP 区别 4. TCP 可靠机制保证 5. Python 协程，协程与线程、进程等有什么区别   **头条二面**   1. LeetCode 原题：[650. 2 Keys Keyboard](https://leetcode.com/problems/2-keys-keyboard/description/" \t "_blank) 2. LeetCode 原题：[166. Fraction to Recurring Decimal](https://leetcode.com/problems/fraction-to-recurring-decimal/description/" \t "_blank) 输入两个数字，分别为 a 和 b（都是整数），求出 a/b 并以以下字符串形式返回： 如果 a / b 能除尽，如 1/4，1/2，返回： 1/2 -> 0.5 1/4 -> 0.25 如果 a/b 不能除尽，如 4/3，1/7，那么返回： 4/3 -> 1.(3) 1/7 -> 0.(142857) 即把循环部分用括号表示   <http://blog.csdn.net/u012162613/article/details/41998617>  ===================================================  更新： 今天发现头条二面两道题都是 LeetCode 原题...  （挂，这两道题，楼主都没答出来。）  腾讯特别重视网络基础的 |

|  |  |
| --- | --- |
| 四川大学。 百度 |  |
| 百度运维部到学校宣讲，可以现场投简历。投完简历可以当天面试，一共进行了三面：  **一面：北京远程电话面试（40 min）**   * 自我介绍 * 介绍项目 * 介绍一下项目中所使用过的**设计模式** * 进程间通信方式？大概聊聊它们如何通信的。 * TCP四次挥手，为什么需要四次？ * hash\_map底层实现？tree\_map底层实现？ * 算法题1：如何对一堆数列“1, 2, 1, 2, 3, 2, 1, 2, 5”按照其**出现次数**进行**排序**，如上述数列中数字1出现了3次排到第一位。 * 算法题2：如何得到一**字符串数组**的**最长公共后缀**？如“aabb, abcd, aacd, adsf, sdfasff”，**最长公共后缀**为“**cd**” * 数据库懂不懂？如何一次性插入1000条记录？ * 如何获取新知识或技术？   **二面：学校现场面试（1 h）**   * 介绍一下到**公司实习**参与的项目，你做了些什么？这个问题问得比较深入。 * 介绍项目 * 解释内存结构中的**堆**和**栈** * 有一**日志文件**，里面包含很多日志记录行，一行的形式为“10.10.10.1 + 字符串 + 10.20.10.2 + 字符串”，在日志文件记录行中找到某一IP是否存在。 * **网页爬虫**：访问某一URL，获取到其中的图片数据，并继续访问该URL里面所有的链接。对其所有链接进行同样的操作（保存图片，访问其链接），爬取深度为4。这道题主要考虑并发，URL去重和树的遍历方式（DFS or BFS）... * 用两个栈实现一个队列 * 判断一链表是否有环？还有没有其它解决方案？ * 对类似“1, 0, 10, 9 , 7, 6, 5, 4, 5, 9, 1”的数列进行排序，并用代码实现。 * TCP的拥塞控制 * 有一计算集群，如果得到某一计算节点的CPU利用率。   **三面：经理面（20 min）**  谈生活，谈人生，愉快的聊天中很容易挂... （比如性格不合适呀，工作地点不合适呀... 这些他不会直接问，可能是夹杂在一些无关的问题中。）  等面试结果.... | |
|  |  |

|  |
| --- |
| C++ 清华大学。 头条一面。 |
| 一直在看别人的面经，自己也来反馈给牛客网。。。  我的经历是直接上code（仅做参考）  三个问题：  1、unordered\_map的实现原理。。很具体的描述  2、单链表多路归并排序  3、LRU cache的实现，插入、查找、删除时间复杂度不能超过O(1) LRU和LFU  有一个场景，现在cache满了，需要淘汰频率最低的那个 |
|  |
|  |
|  |
|  |

|  |
| --- |
| 哈工业 。 头条一面。 |
| 唉 其实面试之前就觉得自己没什么希望。 自我介绍 讲一下自己熟悉的机器学习算法（我讲了LR）然后面试官问了各种优化问题 又讲了决策树，这个因为自己实现过，也做了一些速度上的优化，说的比较顺利 讲了一下实习，但是我实习只是做了一些简单的工作，所以再往深了问细节又是懵逼。 bagging和boosting BP神经网络做了哪些优化 梯度消失是什么，有哪些解决办法 dropout了解吗 手写Trie树 头条面试很注重对算法的深入思考，要有自己的理解。也很注重实践和优化。面试官很好，很谦虚，很好的一次面试体验，再次认识到了自己的短板。虽然没过，但是也很有收获，校招再战！ |
| 评论：同算法一面没过，面试官听完项目一个问题都没问(感觉他并不是做dl的)，撸了三个代码，卒……代码还是练的少(面试官还是很好的给引导) |
| 是的了 感觉面试官很不错啊 我倒希望他多让我写几个代码呀 一直在问**优化优化** 唉 |
| 跟你一样，上来说完自己对机器学习和深度学习比较了解，面试官说了句好，咱来写个代码吧……一面卒 |
| [dahaoshen](https://www.nowcoder.com/profile/398736) 回复 [小诸葛080](https://www.nowcoder.com/profile/2739206)：  我的是，好，来咱们做两个题吧。。。一面卒  [dahaoshen](https://www.nowcoder.com/profile/398736)（东北大学） 回复 [Ronald](https://www.nowcoder.com/profile/7566549)：  准备了很多机器学习的问题一个没问 |
| 光问这个的话可以答的应该很多吧，梯度更新的方式，dropout，bn这些小trick，还有可以从模型传输，模型集成的角度都可以说 |
| BP神经网络的优化应该就是答深度学习的一些改进吧，激活函数、损失函数、特征探测等 这些都是在做优化吧？ |

|  |  |
| --- | --- |
| 南邮 拼多多 服务端。 |  |
| 今天下午在南大面了拼多多服务端的内推，刚进去让填一个个人信息，之后会叫你的名字去面试，我面的是服务端开发，现在介绍一下我的面试经历： 一面：上来手写代码，题目比较简单，类似归并排序，取两个有序数组的并集，结果要有序；然后问了下实习经历，http和tcp的区别，分别适合什么样的场景，用过什么关系型数据库，索引有什么用，为什么索引能加速查找，jvm的GC机制，同步和异步区别，最后又出了一道题，大概是怎么保证在相同url请求的情况下，只处理一次，将结果回应给不同客户端； 二面：上来先给你一张纸，上面有各种名词，解释他们都是什么，包括java类，数据库，linux等；然后讲一下项目和实习过程中用到了哪些技术；介绍statement和preparedstatement的区别，mysql中有哪些数据类型可以表示字符串；http和tcp的区别，适用什么样的场景； 三面：也就是hr面，简单聊一下简历，聊聊对拼多多的认识，拼多多的作息时间，薪资期望，能不能提前实习等等； 感觉问的相对基础，但是本人比较渣，回答的也一般般，大家有后续面试的可以做个参考。 | |
|  |  |
| 基础一个没问，3个大数据的处理算法，一个稍复杂的 SQL 语句，然后分页，然后优化处理；算法面试官都不满意，基本都是没见过的大数据量处理，我明明是面的是后台开发岗，不知道搞什么。 | |
|  |  |
| 一面HR，聊了聊简历，二面技术两个算法聊了很多，三面写了两个算法后就问了三个问题就让走了，很诧异的问了三面面试官这么快？应该GG了吧 | |
| 楼主薪资期望说的多少？顺便问一下拼多多现在的平均市场价是多少？  发表于 今天 11:28:51[回复(3)赞(1)](javascript:void(0);)  [Mr\_sunrise](https://www.nowcoder.com/profile/4496653)：  我说14\*14，他说一年基本14薪，另外有0-4个月年终奖  今天 11:40:50[回复赞(0)](javascript:void(0);)  [Ben\_3738](https://www.nowcoder.com/profile/2033953) 回复 [Mr\_sunrise](https://www.nowcoder.com/profile/4496653)：  这么少，牛客上不是说都是30W+吗  今天 11:48:15[回复赞(0)](javascript:void(0);)  [Mr\_sunrise](https://www.nowcoder.com/profile/4496653)：  不知道啊 | |
|  |  |

|  |  |
| --- | --- |
| 华科，安卓，头条 |  |
| 1. Activity的生命周期过程，从A1跳转到A2按时间先后顺序描述对应的回调过程，singleTask的启动过程。  2. Java垃圾回收机制算法。  3. HashMap实现的方式，链式哈希如何取出正确的那个碰撞元素。  4. Handler是如何实现线程间的通信的。  5. 浏览器打开网页的中间发生的全过程，TCP是怎么解决丢包的情况的，滑动窗口的作用和具体实现。  太菜，结果就挂了，发个帖子把自己打醒。入则无法家拂士，出则无敌国外患者，国恒亡。 | |
|  | |
|  | |
|  | |
|  | |
|  |  |

|  |  |
| --- | --- |
| 华科，安卓，头条 |  |
|  | |
|  | |
|  | |
|  | |
|  | |
|  |  |

|  |  |
| --- | --- |
| 华科，安卓，头条 |  |
|  | |
|  | |
|  | |
|  | |
|  | |
|  |  |