写了个显眼的标题，就真得说几句有用的话。

5 月份一个很偶然的机会，加了叶神的微信，还收到了祝福。一激动就承诺说写篇最详细的面经分享给大家，毕竟用了这么久的牛客网，收获真的很大。

校招真的是段劳心伤神的经历，我把这一路的体会，写在秋招前，也许能给那些和我一样迷茫过，怀疑过，失落过的人一些帮助。

（这篇文章有点长，可能需要点耐心）

0. 写在之前

首先呢我的面试经历和一些面霸和收割机的大神相比绝不算丰富，但我这三个月应该能代表很大一部分人的心路历程：

从无忧无虑也无知的状态，然后遭遇挫败，跌入低谷，连续数天的黑暗，慢慢调整，逼着自己不能松懈，看到改变，收获肯定，分享经历。

先大概说下自己的面试经历吧**（详细的面试过程和面试题解析在最后）：**

相关的公司有四类：

1. 笔试就挂了的：网易、头条。（对于笔试这个东西我到现在都没太多经验）

   网易是第一家参加笔试的，面试会报销费用去总部，所以还是有难度，这个公司呢，内推不内推都要参加笔试。所以笔试要好好准备（当然笔试也不好准备）

   头条只有四道编程题，一个人做确实难度挺大的，尤其是工程为主，不太专搞算法的同学。

2. 笔试过了但因为时间地点原因没去面试的：京东、小米、去哪、携程、招银。

   这几个公司面试一定要现场，很多都要求去总部，还不报销路费。所以在遇到这种情况时，大家要有选择的安排，毕竟时间和精力都是有限的。

3. 最终参加面试的：
   - 阿里（电话 2 面，很轻松地就挂了）
   - 腾讯（现场 2 面，很久之后 hr 电话说要转 C++，就只能算了）
   - 百度（电话 2 面+经理面，offer）
   - 滴滴（视频 2 面+现场终面，offer）

- 360 （视频 2 面+hr 面，offer）
- 美团（电话 2 面+hr 面，offer）。

阿里呢大家都知道，恨不得把全国学生都面一个遍，对于这种策略我只想说三个字，**晚点投**（下面会解释）。

腾讯的面试体验还是很好的，会在酒店的房间里，安静舒适。只是 java 的同学可能要被问到很多 C++和网络的知识，建议还是提前准备一下。

百度内推都是电话面，会在某网站实时写代码，感觉百度的面试官都很不错，给我的体验也很好。内推的时候最好注意下部门，在一个好的部门和边缘部门还是差别很大的。

滴滴面试比较晚，外地可以先视频面，最后会有现场面，报销花费。工作体验很不错，内部氛围很好，因为成立时间短业务扩展太快，技术方面正在沉淀期，工作还是比较有挑战性的。

360 要先笔试，然后可以视频面试。这个公司比较稳定，业务也都很成型，实习的话留用率还是比较高的。

美团的话印象比较深，因为我是二月底内推的美团，然后内推没面我，三月正常笔试通过，然后待面试两个多月。。估计是补招了才给我打的电话（我都已经准备入职了）。经验就是**一定要选一个事业群**，最开始选**都喜欢**的大多是以为这样机会多，相反这样很多部门由于自己的简历池太满而顾不上捞你。

4.还有一投简历就沉的外企：Amazon、FreeWheel（不知道是他们只收北京生源，还是我英文简历写的太烂了）

文章最后的面经当然只包括第三种因为笔试这东西真的不好讲，杂七杂八什么题都有，编程题大多人也要看发挥。

然后这是我最直观的感受：

offer = 心态 *（实力 + 面试技巧）+ 缘分运气

我就从左到右说起吧。

1. 心态

其实每个阶段的改变，也都是心态转变的过程，所以首先要说的一点，就是心态。在上面这种公式里，心态作为一个因子存在。没错，心态可以影响你面试的准备，笔试的状态，实力的发挥,可以影响一个面试过程的方方面面。心态崩了，就只剩运气了，offer 估计也就走远了。

那怎么调整自己的心态呢？

- **首先是要正视自己的能力。不轻视，不高估。**

不轻视指的是我们都要对自己有信心，毕竟选了这个行业和方向，说句不好听的话，机会那么多，就算你不怎么努力也会有个差不多的 offer 不是么？千千万万的初中创公司，各种拥有垂直领域稳定份额的二三线公司甚至有些已经上市，除此之外还有银行，投资，金融的 IT 岗，还有各行各业为了互联网+的策略而扩展的互联网分支（当然像链家这种已经算互联网+传统行业的典范了，我前些天还瞅到了像万达德勤一类的各行业巨头也在招程序开发...）所以啊，要对自己有信心，在这个行业十分缺人的年代里（当然各个层次的公司缺人的标准是不一样的），总不至于会失业吧。

不高估就是要清楚自己的能力范围，不是说期望过高不好，但过高的期望会让你的心理变得脆弱，稍有不顺心态就有崩掉的趋势。因为面试毕竟有太大的偶然性，就算你达到了一定的水平，相应水平的岗位也不是百发百中的，更不要说身边有那么多大神和收割机，天天拿 offer 拒 offer，对心理都是不小的冲击。

- **不要总给自己消极的暗示，心态差了积极调整。**

大多数的人，总暗示自己说什么时间不多了，怎么每天过这么快效率怎么这么低。到笔试了，跟自己说这个算法太难了，肯定做不出来；临到面试了，跟自己说千万别问我 linux 内核，别问我分布式，问我肯定完；面试过之后，没有结果，就天天想肯定又挂了，唉我怎么这么菜。

如果这一系列的表现形成习惯，那心态这个系数最多 0.5，能发挥出来的东西也都打了一半的折扣。凡事都不要太悲观，一个 offer 没拿到，正常情况是这个失败的经历产生的经验和总结是会让下一次的成功率提高的，千万别因为这些消极的暗示，反而让该有的提高都没有了。

心态差的时候反而要停下你重复而没有效率的工作，去调整一下，可以出去玩一玩，吃吃喝喝，不要把这几个月看得有多么不一样，好像耽搁两小时就要来生再见一样。

- **对于身边的同学，多交流，不比较。**

有一些一起准备的好伙伴是件非常好的事情，不仅可以互相督促，而且可以在交流和分享的过程中取长补短。（哪怕是偶尔一起吐吐槽发泄一下也比一个人崩溃要好得多）。但对于每一次面试，只跟自己比就行了，面试的整

个过程都是最怕比较的，偶然性大，而且每个人适合的方向和技术栈都不一样，结果不能说明一切问题。如果身边有些收割机，那更不要太在意这些天天拿 offer 的（尤其是拿一个就跑过来讲一句的，他们也许没有恶意，但他们可能真的想不到这种方式会对别人产生怎样的冲击）。

对自己的水平有个很清楚的认识，并选择自己合适的公司，好好准备自己的笔试、面试，不怕失败，但保证每次面试都有收获和提高，那满意的 offer 早晚会来。

2. 实力

这里说的实力指的是硬实力，也就是技术上的真实积累（当然产品岗的就是对于产品相关知识的积累），而其实呢软实力在面试过程中也尤为重要（有时候真的要更重要），主要是指和面试官的沟通，对一个问题的阐述方式和表达方式，逻辑思维能力以及的价值观和为人等。

然后关于这个软实力我想放到下面的技巧中去说。

如果你平时有些项目积累，不擅长也从来没有怎么系统地总结过；如果你学习呢不算那么认真刻苦，研究搞得也不算出众，但是你该努力的时候也会努力，可以为了一个好的工作逼自己一把。

那也许下面说的实力和技巧都可以在一定程度上帮到你。（因为我就是这样的情况）

- **实力怎么能体现出来**

这么说是因为很多人其实在这些年的项目或是学习中有一些积累，但是在刚开始面试的时候完全感觉使不出力，就是有种你问的我都能讲上一点，但根本说不清楚的感觉。（往深了问自然是完蛋）

知识面或者或者说技术栈都是有宽度和深度的，我们要做的就是在短时间内提升宽度，抓住以往本身就熟悉或是感兴趣的几个点去深入。

我是认为知识的**宽度可以很大程度上决定你能不能通过面试**（这里说的宽度并不是简历上了解、知道或者你都没往简历上写，仅仅是听说过的层面修饰的方面，而是写熟悉的那种，通俗地说是可以讲清原理，不涉及横向对比和优化的方面）

一两个点的**深度呢在保证你能通过面试的同时**（尤其是你的点正好 cover 了对方部门的技术栈），**并且还是影响 offer 等级的关键因素。**（达到足够的深度，不只能说出原理，还能进行横向技术对比，纵向的延伸技术，优劣点及优化，或者在这个点写了几篇很透彻的博文，更厉害的同学甚至有相关的开源项目的参与与贡献）

关于怎么提升宽度和深度可以看下面的积累（其实说真的短期内宽度是好补的，深度确实要看个人，是代价较高，性价比不一定高的一方面，看自己的决定吧）

所以我的建议是，先把宽度提上来，把你能 cover 的知识点及原理搞懂是第一步。建议对自己之前的项目和技术积累做一个总结和分类（可以参考下面的技术路线），然后对已经了解的方面尽可能延伸，对盲区或是薄弱的地方进行针对性的学习和练习。

当你的知识面覆盖到一定程度的时候，你自然会把他们的联系搞明白，慢慢理解这整个技术体系，在面试的过程中结合表达技巧可以关联起来说各个方面，也就可以很大程度上展现自己的真实实力（更厉害的同学甚至可以有超出自己真实实力的表现。。）

- **实力包含哪些方面**

（我也只是脑子里过了一遍，肯定有遗漏的，大家自己补全就行了，毕竟每个人的知识覆盖范围也不同）

1. 基础知识：
    1. 算法和数据结构
        1. 数组、**链表、二叉树**、队列、栈的各种操作（性能，场景）
        2. 二分查找和各种变种的二分查找
        3. 各类排序算法以及复杂度分析（**快排、归并、堆**）
        4. 各类算法题（手写）
        5. 理解并可以分析时间和空间复杂度。
        6. 动态规划（笔试回回有。。）、贪心。
        7. 红黑树、AVL 树、Hash 树、Tire 树、B 树、B+树。
        8. 图算法（比较少，也就两个最短路径算法理解吧）
    2. 计算机网络
        1. OSI7 层模型（TCP4 层）
            - 每层的协议
            - url 到页面的过程
        2. HTTP
            - http/https 1.0、1.1、2.0
            - get/post 以及幂等性
            - http 协议头相关
            - 网络攻击（CSRF、XSS）
        3. TCP/IP
            - 三次握手、四次挥手
            - 拥塞控制（过程、阈值）
            - 流量控制与滑动窗口
            - TCP 与 UDP 比较
            - 子网划分（一般只有笔试有）

        - DDos 攻击
    4. (B)IO/NIO/AIO
        - 三者原理，各个语言是怎么实现的
        - Netty
        - Linux 内核 select poll epoll
3. 数据库（最多的还是 mysql，Nosql 有 redis）
    1. 索引（包括分类及优化方式，失效条件，底层结构）
    2. sql 语法（join，union，子查询，having，group by）
    3. 引擎对比（InnoDB，MyISAM）
    4. 数据库的锁（行锁，表锁，页级锁，意向锁，读锁，写锁，悲观锁，乐观锁，以及加锁的 select sql 方式）
    5. 隔离级别，依次解决的问题（脏读、不可重复读、幻读）
    6. 事务的 ACID
    7. B 树、B+树
    8. 优化（explain，慢查询，show profile）
    9. 数据库的范式。
    10. 分库分表，主从复制，读写分离。
    11. Nosql 相关（redis 和 memcached 区别之类的，如果你熟悉 redis，redis 还有一堆要问的）
4. 操作系统：
    1. 进程通信 IPC（几种方式），与线程区别
    2. OS 的几种策略（页面置换，进程调度等，每个里面有几种算法）
    3. 互斥与死锁相关的
    4. linux 常用命令（问的时候都会给具体某一个场景）
    5. Linux 内核相关（select、poll、epoll）
5. 编程语言（这里只说 Java）：
    1. 把我之后的面经过一遍，Java 感觉覆盖的就差不多了，不过下面还是分个类。
    2. Java 基础（面向对象、四个特性、重载重写、static 和 final 等等很多东西）
    3. 集合（HashMap、ConcurrentHashMap、各种 List，最好结合源码看）
    4. 并发和多线程（线程池、SYNC 和 Lock 锁机制、线程通信、volatile、ThreadLocal、CyclicBarrier、Atom 包、CountDownLatch、AQS、CAS 原理等等）
    5. JVM（内存模型、GC 垃圾回收，包括分代，GC 算法，收集器、类加载和双亲委派、JVM 调优，内存泄漏和内存溢出）
    6. IO/NIO 相关
    7. 反射和代理、异常、Java8 相关、序列化
    8. 设计模式（常用的，jdk 中有的）

9. Web 相关（servlet、cookie/session、Spring<AOP、IOC、MVC、事务、动态代理>、Mybatis、Tomcat、Hibernate 等）
10. 看 jdk 源码

2. 项目经历
    - 这个每个人的项目不同，覆盖的技术也不一样，所以不能统一去说。
    - 这里的技巧呢，在下面也会详细说明。
    - 无非是找到自己项目中的亮点，简历上叙述的简练并且吸引眼球，同时自己要很熟悉这个点（毕竟可以提前准备）
    - 最好自己多练，就像有个剧本或者稿子一样，保证面试中可以很熟练通俗地讲出，并且让人听着很舒服。
3. 实习经历
    - 这个很抱歉，因为我是找实习的经历，所以也没有实习经历的讲述经验。
    - 但我想如果你有实习经历，那面试过程的重点也会在实习做了什么上面，所以大家最好对实习所做的工作做一个总结，并且同样抓出亮点，搞懂内部原理，提前锻炼讲述的过程。
4. 其他扩展技能（这个方方面面太多了，全部掌握基本上不可能，只是作为大家其他时间扩充技能的参考）
    - 分布式架构：（了解原理就行，如果真的有实践经验更好）
        - CAP 原理和 BASE 理论。
        - Nosql 与 KV 存储（redis，hbase，mongodb，memcached 等）
        - 服务化理论（包括服务发现、治理等，zookeeper、etcd、springcloud 微服务、）
        - 负载均衡（原理、cdn、一致性 hash）
        - RPC 框架（包括整体的一些框架理论，通信的 netty，序列化协议 thrift，protobuff 等）
        - 消息队列（原理、kafka，activeMQ，rocketMQ）
        - 分布式存储系统（GFS、HDFS、fastDFS）、存储模型（skipList、LSM 等）
        - 分布式事务、分布式锁等
    - 脚本语言：（只是作为横向扩充，一般问到 linux 也会问问 shell 脚本）
        - python
        - php
        - shell
        - golang
        - ...
    - 大数据与数据分析：
        - hadoop 生态圈(hive、hbase、hdfs、zookeeper、storm、kafka)
        - spark 体系

        - 语言：python、R、scala
        - 搜索引擎与技术
    - 机器学习算法：
        - 模型和算法很多。不细说了，如果很熟练就去投算法，国内很多公司都算法岗都很稀缺，其他岗可以大概了解下理论。
    - 其他工具的理论和使用：
        - 这个更多了，问的多的比如 git、docker、maven/gradle、Jenkins 等等，自己需要的话选择性地去学。

5. **实力要怎么积累**

积累实力最好的方式就是平时在项目中或是学习中，多学多问，多思考多钻研。这里就说说短期内学习的一些方法和路径：

总结下来，一方面是通过看书、看视频、看面经来不断扩展自己的知识面，一方面是通过不断的面试积累经验和知识盲区，在每次总结的过程中积累实力。

-
    - **先说说看书学习这一点**

这里应该是有个推荐书籍的环节，这几个月确实看了不少书，但是并不是所有都有很高的性价比，在这边大概列一下，前后顺序也一定程度上代表了我认为的重要度先后，'/'做分割的是一类的书籍，一般来说看一个就够了：

书单：

算法与数据结构：

数据结构（严蔚敏）/大话数据结构 //如果觉得教材无聊就可以看大话系列，印象中里面还有很多诗

剑指 Offer/程序员面试金典/编程珠玑/编程之美/牛客网+leetcode

程序员笔试面试最优解（左程云）/不如直接看左神的笔试面试指南视频

Java 的版本（不是很推荐）：

数据结构与算法经典问题解析（Java 语言描述）

图解数据结构（使用 Java）

计算机网络：

计算机网络（谢希仁）

TCP/IP 详解

HTTP 权威指南

图解 TCP/IP

图解 HTTP

数据库：//数据库主要是多用，书上主要看索引和性能的部分

高性能 MySQL/深入浅出 MySQL

操作系统：

OS 原理：操作系统（课本，黑色的那个）

Linux：

Linux 私房菜 //鸟哥写的，很全，包括 bash 部分

跟阿铭学 Linux //主要偏重于命令和操作，比较浅显

java：

Java 疯狂讲义/Java 编程思想/Java 核心技术 卷 1

深入理解 Java 虚拟机

并发编程的艺术/多线程编程核心技术

Effective Java

Java 程序员面试笔试宝典 //何昊的那本，个人感觉是突击知识点的神器

Java 程序性能优化

实战 Java 高并发程序设计

Java Web：

Spring 实战/轻量级 JavaEE 企业应用（红皮，讲 SSH 的） //主要看最后一部分 Spring 的就可以

深入 JavaWeb 技术内幕（阿里 许令波）//这个讲的还是比较深的

SpringBoot 实战/深入实践 SpringBoot

设计模式：

大话设计模式 //通俗易懂

各类博客的总结

分布式与大数据：

分布式服务框架原理与实践

大型网站技术架构

Hadoop 实战（hadoop 体系包括得很全）

//还有一本我暂时想不起来名字了

其他：

Git：

Git 权威指南

Git 官方讲解视频（牛客网有带字幕的）

Redis：

Redis 实战

还买了 docker、springCloud 等等一些工具书，因为太小众就不列举了

除了上面说的书和视频，最有用的还是大家分享的各种面经。

面经是个很不错的东西（嗯，想看的可以直接翻到最后一节）。记住不要一扫而过（除了那种岗位不太匹配可以快速抓重点看），岗位匹配的你可以顺着面经逐条去看，模拟一次面试过程（虽然是单向的，但是你心里应该是知道每个问题你能答道什么层次），这种不断地模拟可以让你知道盲点或者说弱点在哪，对于一个你心里没底或者想不太起来的问题一定要当时记下来或者当时就弄懂。

我当时是每天晚上在床上看几篇面经，然后把存疑的问题 copy 到记事本中，第二天找个固定的时间短查询解决和总结。长期下来会养成一个很好的习惯，你的知识点会不断地扩充。

-
  - **在一次次面试中提高**

这一点其实跟刚刚看面经的那个很类似（那个我不是叫做模拟面试了吗），每次面试完，一定要把自己不会的东西尽快记下来（当然你如果有记面经的习惯就更好了，还能再分享出去），然后找时间弄懂和总结。

除了知识点，每次面试（跟模拟面试不同的地方）还要总结的是哪里发挥得不好，哪里有可以提升的地方，下次一定要注意之类的。（就是有关面试技巧的东西，比如这次语速太快，导致面试官反馈了，又比如这次发现对方让你在纸上画个项目逻辑图突然一时脑梗，那回去就好好在本上画一遍）

3. 技巧

- **一些自身的软实力**

软实力这个东西与面试的准备关系不大，基本上是长期形成的。

包括和面试官的沟通（有的时候也会成为聊天瞎扯的能力）；对一个问题的表达方式，逻辑思维（像有些人的发言就让人听上去很有层次感，很舒服，这方面欠缺的可以推荐玩玩狼人杀）；除此之外对方也会很在意你的价值观和为人（这个是我进了公司发现的，很多公司在内部的面试细则上面都会注明这一点，如果价值观或是人品问题会直接否决。）想想也是有道理的，因为这个是入职之后能不能好好相处的关键，设想如果你是一个面试官，面对一个有实力但是说话太有棱角聊不太来的和一个不算出众但基础不错很听话可以培养的，你会选哪一个？（如果你因为生活太平淡了想选第一个，那这一条当我没说。。）

这些软实力其实要在生活中慢慢锻炼，比如多参加些活动，多和别人沟通，发表意见前好好组织自己的语言等等。（每个人都有每个人的性格，这些都因人而异，但有一点我们要记住的是**在面试中，不论什么情况，都要保持冷静和清晰的头脑，和一个谦卑的态度，交流要坦诚**<尤其作为应届生>,这样起

码印象会好很多，要知道虽然面试有各种各样花式的打分项，但是印象好往往是隐藏的决定性因素）

- **关于面试的准备和技巧**

面试的技巧首先就是刚刚说到的态度。 一个谦卑（注意不是自卑，也不是把自己放的很低的样子）和礼貌的态度和表达方式往往可以让面试官的印象分提高很多，印象分很像之后要考察的实力分的系数。留个好印象，面试就成功了一半。（当然你一定也听过那种聊得很嗨，或者偶遇校友之类的，毛都没问就聊通过了的场景。这种情况还是不算在技巧中了吧，应该属于运气和缘分的范畴。而大多时候聊得开心和舒服会让你感觉到通过率会比较高，这一点很多人还是深有体会的）

-
  - **然后说一下面试之前可以做足的准备。**

首先从简历开始说，简历怎么调格式，做几页，排版啥的就不想多说了，感觉很多文章都分享过。我只说说技术方面的内容怎么写。

专业技能的描述谓词无非就那么几种：**精通、熟练、熟悉、了解**。（还有一级叫**听说过**，这个级别的可千万别往上写啊兄弟）

精通感觉一般还是不要写，除非你在某个技术点上真的有足够的把握，比如源码看的很透彻的同时还能深刻理解原理并能灵活处理各种 case 场景，如果还有相关的开源贡献，那就自信地写精通吧。

我们把自己掌握的大多数技术点叫做熟练掌握，这个需要我们**在之前对各个技术点进行横向纵向的复习和总结**，并不只是用的多有经验就行，有的时候我们觉得熟，但真让你说的时候却不知道从何说起。

至于熟悉和了解，可以写一些自己理解原理但是不常用的技术点，尤其是比较流行的，各大公司都在用的技术（比如 MQ，分布式缓存等等），这些你在学校不一定用过，但是你可以通过看博客，写 demo 去理解他的设计和原理，面试的时候可以讲得清楚。

这里还有个技巧，更细心的同学可以针对每个公司岗位的 job detail 不同，熟悉和了解这块就针对 jd 中 cover 的技术点去写。这个做法是很聪明的，毕竟熟悉和了解这个层面是可以提前学习和准备的，有针对性的去写对方需要的，是提升通过率很好的一种方式。（如果嫌麻烦就算了，比如我就是）

下面再说说项目经历这块：叙述一定要精炼到位，细化到每一个亮点上。我现在再看我二三月份的简历简直是有种想撕了的冲动，当时就是项目描述两三行，然后概括下我大概做了哪几个模块。完了。
事实上，不能讲得这么泛泛，就从中找 2-3 个亮点，一句话高度概括，突出亮点。

比如后来我就把我一个普通的 web 项目挖出来三个点（爬虫，通信控制方面，安全加密方面），分别用一句话叙述，这一句话最好包括这个技术点的思路，解决了什么，有没有做什么优化。比如一个爬虫工具可以写成这样：

我负责这个系统中爬虫的开发，终端控制管理，店铺管理。。。。

是的，上面这个爬虫就占用了五个字，等于没说，面试官看到压根不知道怎么去问。这里还可以这样写：

负责系统中爬虫模块的开发，用于爬取影片的各种信息，包括年份，介绍，主演信息等。

这个起码告诉面试官爬虫做了什么，但是你并不能保证他会问，因为你的叙述不一定能引起他的兴趣。那还可以这么写：

独立开发多线程影片信息爬虫工具， 并针对线程池性能、网络异常以及反爬虫措施进行多次优化，容错性良好并达到并发请求 30+的 QPS。

你觉得面试官看到这句话，他不想问点什么吗（除非他真的就不想要你）。

所以说，其实面试官呢都是希望在面试的过程中发现我们的亮点和优势，从而展开更深层次的交流，但是往往在简历中没有一个让他询问的入口，这样就可能导致他会随意问（比如你了解什么什么吗？一般情况下都不太乐观），或者就说那你来讲讲吧（这种问法其实已经表达出他不知道怎么问的情况了，你在讲的时候一定要突出重点，否则会让人感觉没什么亮点，普普通通的项目而已（当然就算是普通的项目，我们也要挖掘它的价值和自己的价值不是））

我们在面试的过程中最重要的是自己掌握主动权，如果面试官问的都是我们熟悉的范围和准备过的领域那我们答起来也会更加得心应手。而让面试官随着我们简历中埋下的这些亮点（他就算知道你有意想说这一点，往往也会去问，因为他就是想在短时间的面试中了解你处理问题的能力），去进行更深层次的交流，而这个更深层次的交流，还需要我们针对简历上的每一句话，都准备多个层次和维度的扩展。

比如还是这个爬虫，你可以充分扩展线程池的相关优化（有可能项目中没有怎么优化甚至就是个单线程，但是在准备面试的过程中还是可以专门去做一些 code 重构的），优化网络的 NIO 相关扩展，以及反爬虫的各种各样的措施，以及爬虫方分别如何应对。这里只要你去想，能准备的东西太多了，面试多了自然也就越来越熟，好像项目就是做得这么完美一样。

这里我就不怎么扩展去说了，再讲下去这篇文章就写不完了。。。

最后是面试中的技巧和经验。

1. 好好对着自己写的简历一行一行看一遍，这都是你挖的坑，是准备给面试官作为切入点交流的，并不是自己往里跳的。（对每一行都要有足够了解和把握）
2. 面试过程不要紧张，尤其是前几次，建议先从小公司入手锻炼下面试经验（参考我之后自身的反面教材
3. 面试方式不同，侧重点不同（无非是电话、视频、现场三种）。

    电话面试建议找个人少安静的地方坐着回答，并且建议拿纸笔多做记录多画多写。（当然如果你觉得身边很多朋友可以让你越聊越嗨那也可以，坐着是让你整个节奏慢下来，说话明显更加沉稳，亲身体会过站着走来走去和坐着的区别）

    视频面试其实和电话类似，只是可以实时写代码，面试官能看到你的表情。这里还是要放松，如果你比较紧张，可以不直视镜头，好好想问题就是了，因为很多面试官你答得好也会面无表情（因为他们也不常视频，表情都很尴尬），然后你看到他们没表情的表情肯定会受影响。

    现场面呢，最重要的是和面试官互动了，说几个点：语气要轻松点，多点肢体动作有助表达，多笑；不太好说清的就用笔在纸上画，一遍画一边讲，面试官也会更容易和你交流；如果你可以时不时幽默一下开开玩笑是更好了；见面和离开记得礼貌地握个手说声谢谢。

4. 学会平等交流，别把自己身段放的太低。其实有一点你要清楚，面试是个双选的过程，他可以拒绝你，你也可以拒绝他。千万不要太上赶着，反而会影响自己正常的表达和逻辑。（就跟你见了喜欢的姑娘就不会说话了一个道理）
5. 回答问题的时候不要一口气把知道的全部说完，然后还毫无条理。学会一个知识点由浅入深讲解给面试官，并且留有余地给他进一步去问。

    举个例子：

    就说最简单和普遍的 HashMap，让你讲讲，你就可以先说说 hashMap 的设计原理，底层结构（链表+数组）扩容方式等，从这你就可以说说这种设计好在哪里（比如讲一讲 put 是如何做 hash 的），这时候你可以说这种 hash 可能会有冲突，hashMap 也是做了相应设计的。

    然后面试官会问题你怎么解决冲突？你可以再给他讲讲解决 hash 冲突的三种通常方式，而 hashMap 用的是链式法，然后可以说到这样会有隐患就是 hash 链过长。

    面试官再问，你会给他讲解决复杂度高的长链用了红黑树的结构，这里还可以延伸到红黑树的特点或者 jdk7 和 jdk8 的

不同实现，这时候你可以说解决 hash 冲突，但 hashMap 还会有并发和同步的问题。

面试官会让你再讲讲，你可以说说 hashtable 是线程安全的，怎么实现的（sync 函数），并不好，从而引出更好的 juc 包，说说 concurrentHashMap，之后又可以说道锁分段原理，弱一致性迭代器，concurrentHashMap 的锁粒度（java7 和 java8 不同），同包的 CopyOnWriteArray 等等。

你还可以延伸说到锁（重量、轻量、悲观乐观各自实现、底层源码等等）、缓存（因为很多时候 Map 的结构可以作为缓存，从而可以说到缓存系统的设计，kv 原理，分布式缓存 redis、memcashed 等等）

举这个例子就是想说，一个简单的基础问题可以一步一步有条理有层次的回答，每一层表达完抛个引子，让面试官可以继续问下去，从而让面试官真正了解你的掌握的深度。

6. 如果真的不巧聊到不擅长的地方，学会转移话题，从一个点中聊自己感兴趣或是有把握的方面（比如你对消息队列不太熟但是 redis 用的熟，你就可以在问到消息队列的时候说，因为之前都是自己做的项目嘛，性能方面没有考虑到最优，一些异步的方式还是靠 redis list 去实现的，虽然 redis 的消息机制并不常见，但当时还是满足了需求，之后可以考虑性能方面的提升和技术评估；又比如问你 http 请求细节，rest 的设计实现细节，你可以说 http restapi 服务接口性能的一些不足，后来使用了 rpc 的方式，当然你这么说一定是要对 rpc 很了解）其实有的时候面试官是知道你是有意转移的，但是往往他们也不会抓着你不会的去问，非让你自己承认自己的盲区，他们也许根本不在意这些。
7. 如果真的被问到不会的，就直接说你不会（说你不会、说你不会，我再补充两遍），或者礼貌地说这方面可能我还要多学习。（对一个拿不准的问题千万不要猜，即使是二选一的那种问题，猜错了直接完蛋，猜对了被人看出来，再往深问还是完蛋）另外，像可能，大概是，我觉得这种表达最好不要，一听就是对一个点没把握，有可能会让面试官觉得学习太浮躁不喜欢寻求原理。
   那对于自己知道原理（确实是理解了的）但是没用过的东西，就讲讲原理，并承认自己实践不足，表现出好学的态度。面试一定要真诚。
8. 问到有什么 offer 就直接说，不要藏着掖着，也不要把更好的 offer（比如 bat 的）讲的非常诱人，一副 bat 我都拿到了的样子（面试官会心想，那你还来面试我们干什么）。**再强调面试过程一定要真诚。**除了直接说，诚实点之外，也要真的做些思考：对方公司跟之前的 offer 比优势在哪，比如平台更大？专业技能栈更 match？工作更有挑战力？地点更合适？有机会留用？随便一条符合的都可以讲出来，起码让对方觉得你想来面是有原因的并且真的有可能加入。（如果你还提前了解对方公司的文化，可以讲出这个文化自己很认同那就更可以了）

4. 缘分和运气

关于这一点只有一句：平时多做好事，热爱生活。

其实都知道面试要讲缘分，讲运气，但人往往可以在很顺利地通过面试之后说句运气好运气好，却很难在努力准备却失败的经历之后保持平静。

但不管能不能转运是不是本命年有没有缘分，努力和收获的关系总是多年不变的真理。

所以，讲心态，讲实力，讲方法。足矣。

下面是面试的详细过程，包含面经：

按时间顺序，详细叙述一下我面试的过程，包括面经和心态的转变：
（括号里是对问题的补充，如果感觉有知识点的盲区，大家正好可以去深入学习一下。这里说一下我投的大多是 Java 研发岗，所以其他语言的可以忽略 java 问题）

首先说一点，**复习准备一定要早**（当然这是说给 19 届师弟师妹们听的，嗯你们看到这里已经可以开始复习了），有同学去年暑假剑指 offer 都刷了一遍，然后我竟然今年 3 月才买这本书。。还有同学前一年冬天就已经去实习了，这种机会也不错。但寒假如果不实习的，一定要进入到学习状态。我准备的就有点晚，寒假完全没看书，真正开始准备，大概是 2 月中旬号玩了一晚上狼人杀之后。

就是那个时候发现校招就要开始了，然后开始慌了。大概看了一周的书（基本上都是 java 基础）,然后师兄说阿里内推了，心想赶紧投吧要不人家招够了就不招了（后来发现都 5 月 6 月了还在招。。），然后就慌慌忙忙投了简历。

这里我要说一点最重要的事情： 一定不要在没有面试经验的情况下先面大厂，或者是你想去的公司。
我是 3 月 1 日下午三点半在阿里的官网完善的简历，5 点电话就过来了。作为一个 java coder，阿里是个很好的平台，（当然 C++的岗可以好好准备腾讯），一定还是准备充分了再投，**你先面的结果很可能是焦急地等一两个月然后被后来准备充分再来面试的人取代**。

阿里 3.1，3.2（两面）

两天各面了一面（投的蚂蚁金服，第一天面的就不太好，第二天又把我捞起来面，并且面完感觉就走远了，但当时却没有 reject，这就导致之后一个月一直在流程中，阿里其他事业部的师兄师姐没办法把我的简历提走，最终到要笔试了然后变成了 rejected）

那个时候就是处于我所说的无知状态，知识点掌握的不够牢，简历写的乱七八糟，面试经验为零（这应该是我上学以来的第一次面试），面试技巧就更不懂了。

然后当时两次都感觉聊得很差（尤其是聊算法模型的时候都想自爆了），也根本没有记录面经的想法。所以有些问题我都记不起来了，大概说一下吧。

- hashMap 的扩容原理，初始有 13 个，要怎么 new？(达到了负载因子，直接手动>>1)
- Integer 的常量缓存池的问题(-127~128 范围有个 cache)
- ConcurrentHashMap 的 size()怎么做的（并没有完全加锁，而是先乐观的认为不会有写，通过 modCount 判断是否更改，这个我当时记不清就用了很多可能、大概、应该这种词，事实证明直接说不清楚会更好）
- Spring 的 AOP 关于拦截 private 方法一些问题.(细节忘记了，当时答得也不好)
- 项目中数据字典怎么做的缓存，如何做的通信，有没有用什么模块。（说了自己的做法，用的全局的 HashMap，然后他会延伸到高并发的场景，分布式缓存怎么做等等，由于没实际操作经验提前也没准备，并且还没有直接说不会，又是用很多模棱两可的语句答得）
- 讲讲你的论文相关的模型吧（这个其实在简历上根本没写，只是写了数学建模的奖，然后面试官就开始问机器学习的算法，很多都是我没准备的，并且我心想我投的也不是算法岗啊。。。**所以说对于简历上的每一条一定要熟悉，做足准备，并且遇到简历上没有都扯到的方面，要想办法转移，不要在这耗着**）
- 讲完算法的问题，面试官很尴尬的说了句，你这自己的研究方向你都搞不清楚吗？我当时预感就差不多走远了。。。

其他问题真的太久远了，我当时也没有记面经的习惯，所以就没有了。但是最大的感受是面阿里的时候整个人都是紧张的状态，语速特别快，恨不得把知道的都说出来，没有条理，并且把自己姿态放得特别低，还在楼道不停走来走去。（对，以上说的这几点全都是不应该的，但主要原因还是当时准备太不充分，简历方面的准备以及知识点的积累都不够；另外一点，还是要强调不会的就是不会，千万别说大概是，我觉得吧这种东西，说的不好很容易让面试官认作不懂装懂，虽然你只是很想向面试官表达点什么，哪怕只是积极的态度）

从阿里面完试开始我的心态基本上就崩掉了，对自己极度怀疑，加上今年诸事不顺，心情直接跌到了低谷。然后整个三月基本上都是黑暗的，整个一个月都没再投内推，每天从早晨起来，大多时间就在看书刷题，晚上十点回

宿舍躺在床上，还要刷两个小时牛客的面经。一个月下来很少说话，提升肯定是有，但是这个过程，其实完全可以用更好的心态去经历。

然后这段极其黑暗痛苦的日子持续到三月底，一个师兄想帮我内推百度，因为之前因为没信心也错过了腾讯的内推，就心想要不试一下吧。

百度（123 面）

一面 3.30 下午两点 45mins

- 自我介绍，印象最深最费心的项目（这个一套可以提前准备，在某些亮点可以估计抛出等面试官来问）
- 讲讲项目中的爬虫和优化怎么做的，为什么选用 jsoup 而没有用 python 的 urllib
- 说说你了解的反爬虫措施，和针对异常的处理。
- 那你觉的你来做一个网站要从哪些方面考虑反爬虫。

  这里可以提前和学习，即使你真正使用的只是一点，也可以在相关问题上做更深的了解。

  最简单只分析请求，拦截所有非浏览器的直接请求（可以通过添加伪报头解决）;查看 refer 页做防盗链接（可以改 refer 项）；基于用户行为的策略，同一 ip 相似请求判定（代理或 ip 池，或间隔请求解决）；基于用户 session 的策略（模拟多用户登录解决）；封装前端数据，用 js 渲染生成（通过探取和模拟异步 ajax 请求解决）；对 ajax 请求进行加密等等方法。

- 讲讲项目中怎么实现的充值，锁的机制和事务注解，如何保证了事务的一致性。
- spring 层面做事务和数据库层面做的区别，各自实现方式。
- 聊了事务的传播性和隔离级别，问了 mysql 的默认隔离级别（可重复读）
- spring 中事务传播性怎么配置（xml 方式和注解方式，还有关于 savepoint 的使用）
- 算法：O(1)删除执行链表结点，做分析（其实是要指出剑指 offer 中那个直接 copy 值的方法的缺陷和隐患）
- 算法：二叉树的最长距离（递归的思想）

二面 3.30 下午五点 50mins

- 聊项目（这次是针对项目中的加密算法和安全性做了阐述，大概 20 分钟吧，之前爬虫那个例子已经说了项目亮点要怎么准备了，这里我就不多说了）
- 详细聊了聊 spring 的 IOC 和 AOP 思想

- 关于 AOP 在 spring 的应用（比如事务，通知，aspectJ，slf4j 的原理，和 log4j 的对比）
- 关于 jdk 代理和 cglib 第三方代理（说出对接口代理和子类继承的区别）
- 用的什么数据库，Mysql。
- 最大的数据量多大，用了索引没有，怎么用的（聊了前缀索引，对于 varchar 类型的值，又聊了聊 char，varchar，text，blob 的关系和区别）。
- 为什么索引不能随便用，什么时候用（什么时候失效，什么时候最高效）。
- 如何达到索引开销和性能的平衡，用了一个表去举例。（方法就是，根据情境看经常做的查询是哪些，然后依次是什么查询条件，保证最高效索引的同时，也保证索引不失效，避免无效开销，并且根据 show profile 和 explain 功能进行对比分析）
- 数据库用了缓存没有，讲讲 redis 的理解（用作缓存，队列，也可做存储）。
- redis 是单线程还是多线程的，举个例子（做计数器，rank 排行榜）
- 讲讲 hbase 的原理，CloumnFamily 包含哪些，region 什么情况做分割，对于版本号这个第四维度的使用方式（一般默认三个版本）等等
- 让我等消息，说经理会联系我。

经理面 4.12 中午 40mins

- 经理面其实更加放松，不只是技术方面，还有生活，性格多方面，感觉是个技术+hr 的综合面。
- 聊家庭，家乡，工作意愿，爱好等等（聊了十几分钟，感觉都很不错，然后之后的面试也就很轻松了，基本是我在讲他在听）
- 讲了讲项目的设计，包括异常的处理，数据库设计，通信模型的设计。
- 讲讲你理解的 JVM 吧（从内存划分说到了 GC 算法、分代思想，CMS 和 G1 collector，到类加载模型，tomcat 的非双亲委派、线程上下文加载器，到 JVM 调优的策略，gc 参数设置策略，如何找死锁，读快照，发现内存泄漏等等吧）
- 然后说了下部门的技术栈和部门介绍，说了部门可能没有留用的 hc，问我介不介意（当时还没 offer 当然说不介意只是希望去学习）
- 然后就说把我简历锁了，之后会给我发 offer。

其实到现在我都很感谢百度，虽然最后因为部门和留用 hc 的原因没有去，但是这次面试收获最大的就是信心。并且经理电话里就给了口头 offer，这个让我悬了一个多月的心一下子就放下来了，接下来的几天乃至之后的面试整个人都是放松的状态。（真的很难形容，这种转变就在这一个电话的前后）

所以，其实大家也早晚会有这一天，没必要一直那么紧张的状态，太痛苦了。（虽然我知道这种话说得简单），另外一点就是三月份确实有了很大的提高，这个告诉我们，该逼自己的时候还是要狠心一点。

360（123 面）

有了信心或者说有一个 offer 之后，你就会越来越顺利，从这之后的每次面试都会有不同程度的收获。360 的流程是走得最顺利的，笔试+面试，三次面试一天完成，但是 360 的面经可能写得稍微简单一点，因为很多知识点在后面的面试中也出现了，就没有做过多解释，腾讯滴滴美团的面经要详细一些。

360 一面 4.12

- 自我介绍加项目
- 线程池如何优化的爬虫，数据规模
- 网络时延如何处理
- 同名影片如何选取，有没有更好的方式。
- 反爬虫的原理，从低级到高级说一下，分别如何应对
- 收获了什么
- 线程进程区别
- 说下资源方面的区别，共享，不共享
- 共享的具体哪些资源
- jvm 内存模型
- 堆区的特点
- 数据库左连接右连接，场景
- 给 200 个 200 个数的数组，找到最大的 200 个
- git 常用的操作，git rebase 和 git merge 区别
- 分布式数据库怎么调用
- linux 常用命令，查看内存，查看磁盘使用率

360 二面 4.12

- 聊项目，介绍下背景，怎么谈的
- 印象最深的模块及解决，其他项目呢，跟着老师做的，还有简历上没写的项目
- 项目经验还可以那基础怎么样自己觉得，
- 说下 jvm 吧你知道的，中间会问
- jvm 详细如下
- 内存模型
- 垃圾回收
- 分代及回收算法
- 哪些作为 gc root
- 收集器的特点分类
- 类加载机制和双亲委派模型
- 几个加载器
- tomcat 类加载有什么不同，说加载顺序并不是双亲模型，具体顺序说一下
- 并发注意什么，线程实现同步的方式，通信

- 几种同步的区别
- 悲观锁乐观锁，底层怎么实现的，越详细越好
- 单例模式的特点，几种实现，容易引发的问题
- 如何防止内存泄露，哪些会容易造成
- jvm 调优如何检查内存泄露，如何优化 gc 参数
- 写 sql 查询带日期多次考试成绩表中，每个学生的每门课最高成绩，日期要准确
- 分别用 having 子句写和用子查询写
- 写代码 旋转数组中查找某一个值

360 三面 hr 4.12

这是唯一一篇 hr 面经，因为我只面过这一次 hr，其他的三面要么是技术，要么只是打个电话说说情况，还没问问题就挂了电话（比如腾讯的）。

- 讲一下项目怎么接的，怎么跟甲方沟通，遇到的最大问题，怎么克服的
- 自己的项目和老师的项目和安排怎么协调。
- 平时有什么爱好，怎么安排自己的时间
- 摄影都去哪拍，喜欢什么运动，什么时候运动
- 为什么选我们部门，其他部门你怎么看
- 我们是做移动端后台的，喜欢玩手机吗
- 说说常用的 app 平时怎么用，频率
- 以后会不会创业，为什么
- 以后的规划，职业技术和生活两方面

hr 面没有太多经验，个人生活爱好这类我感觉就是放轻松聊，规划方向这些可以自己提前准备准备，但是比如创业，offer（之前说过了），价值观一类的问题，其实你也不知道对方想要什么样的回答，干脆随缘吧。

腾讯 （12 面）

腾讯的面试有一点印象很深，很考察思维能力，经常会有一些意想不到的问题，或者智力题。挺有意思。

一面 4.23

- 上来看了我的简历问我会不会 C++，我心想虽然学过，但是好几年不写，还是说不会吧，然后面试官很好，就不问了。
- hashmap 底层结构画一下，手写代码做一个 url 解析器，用正则方式和 hashMap 的数据结构。
- 识别 2 的 n 次方，写个函数。（最快的是用位操作，大家应该都知道 n&(n-1)可以去掉二进制最右的 1，那 2 的 n 次幂&之后便为 0）
- 自己实现 http response 响应头的结构及解析，用 buffer（写个伪代码）
- resp 头中都有什么（主要考察 http 相关知识）

- 海量数据找到出现次数最多的 100 个（内存不足的时候可以先做 hash 分片，最后多路 merge，每次操作可以用 hashMap 计数，也可以自己做 hash 函数计数）
- redis 底层实现，zset 数据结构（问到了 SkipList 跳表这种结构）
- jvm 内存模型，分代，cpu100% 怎么排查（我以为问 Jconsole 的使用，其实是想问 linux 性能监测和调优）
- 用 int 值表示 ip 如何做（刚好 32 位 bit 一对一映射），写个伪代码做 transfer
- nio 模型说一下
- selector 中的 wakeup 什么含义（这个答得不是很好）
- select poll epoll （linux 内核相关的知识）
- arraylist.sort 怎么实现的（这个可以看看 TimSort 的思想）
- 怎么看待 java 跟 c++（说下区别和自己的感受）

- 能实习多久（这个好说）
- 去深圳工作怎么看（这个真没想过，不过当时说也可以吧）
- 有女朋友吗（...）

二面 4.24

二面其实就两个大问题，但一直往很深的地方问。

- 100 亿个数找最大 1000 个（说了分片，用堆，再归并）
- 问你确定吗？（我一想是最大的 1000 个不是出现次数最多的，其实是可以顺序读取，还是用堆实现）
- 有什么缺点，分布不均匀（说一下堆的复杂度由来）
- 有没有其他思路（用 hash 散列，计数排序）
- 这个更慢，还有更快的吗（我心想我平时就都是答得堆啊，怎么这次还有更快的？）
- 然后讲了基于 partition 的划分思想（找到第 k 个开始 partition，在左边就在右面递归，在右面就在左边递归，最后确定 partition 出最大的 k 个）
- 这种思想了解了，但最坏情况太差，不稳定，还有更快的吗
- 是不是要用概率统计学，抽样估计？
- 说下思想。。
- 不够精确，还有又精确又稳定的吗？
- 。。。
- 那又给你一个数，你怎么快速告诉我是不是在这 100 亿个数中？
- 这个我知道有可能是想问 Bloom Filter，但是具体到 hash 函数去几个怎么算，怎么判断误差等细节，我也记不太清了，就说了说思想。

然后进入第二个问题：

- 一个进程最多申请多大空间（看机器 cpu 的处理位数看情况）‘
- 怎么保证进程间数据的安全？线程呢？

- 安全方面有没有做过一些研究？
- 登录验证怎么做的，为什么用 md5，有没有改进（+salt 使 md5 库难解出），微信用的什么方式你知道吗？你想想应该用什么方式？（这里可能是问 SSO 单点登录的原理吧，可以讲讲 SSO JWT token 等技术的原理，这个也是我实习之后了解到的，当时答得一般）
- 那说到通信安全，怎么保证 http 的安全性，幂等性，回调同一个会话怎么标识不同请求，不同会话怎么区分（这个每个问题都画图叙述了下）
- TCP 3 次握手和 timewait 讲一下原理
- 讲一下滑动窗口，饱和了怎么处理
- http 安全吗?https 说一下？
- get 和 post 请求
- linux 怎么查看网络状态（vmstat）
- 查看 udp 的性能，udp 端口多少，什么时候用 udp？
- 为什么 tcp 不行？
- qq 里哪些用的 tcp 哪些用 udp？分别针对每种情况说一下为什么？

可以看到腾讯还是很爱问网络通信、大数据处理的（当然 C、C++也很爱问，只不过我说了别问，他们就真没问，当然你做 java 的也不要期望他们会问你多少 java 问题）

然后当时并没有 hr 面，我心想应该是挂了，但是微信的状态又迟迟没挂，结果到了一个月之后五月底我都回家准备入职了（绝对又是补招的备胎），然后打电话问问个人情况（也没说是不是准备给 offer），问可不可以去深圳，转 C++方向。

当时也有比较好的机会了，并且实习转岗，如果不确定能留下，绝对是不建议做的一件事,所以就实话实说了。

滴滴新锐（123 面）

一面 5.12

- 说说你对现有 Web 开发框架的理解（从各个层入手横向对比优缺点，印象中说了 SpringMVC 和 Struts，mybatis 和 hibernate 及 jpa）
- mybatis 和 hibernate 各自的缓存原理和比较，hibernate 的一级二级和查询缓存，还有针对缓存的 miss 率，置换策略，容量设置和性能的平衡问了自己的理解。
- 要你设计的话，如何实现一个线程池(就讲线程池的原理，从初始线程数，核心线程数，然后到任务队列，满了继续到最大线程数，再满了到饱和策略 handler，饱和策略一般有哪几种，基本上要理解 ThreadPoolExcuter 的构造方法那几个参数)
- synchronized 关键字，实现原理（和 Lock 对比着说，说到各自的优缺点，synchronized 从最初性能差到 jdk 高版本后的锁膨胀机制，大大提高性能，再说底层实现，Lock 的乐观锁机制，通过 AQS 队列同步器，调用了 unsafe 的 CAS 操作，CAS 函数的参数及意义；同时可以说说

synchronized 底层原理，jvm 层的 moniter 监视器，对于方法级和代码块级，互斥原理的不同，+1-1 可重入的原理等）

- 算法：手写一个 ArrayList 类，实现 add，remove，等基本的方法（主要考扩容的原理和实现，重点写出扩容机制以及扩容时的 copy 过程）
- 然后让对这个 ArrayList 进行改进，使之可以应对并发的场景
- 算法：手写字符串的正则匹配，实现*和.的功能，用的递归（写了一半他说时间差不多了，思想大概了解了）。

二面 5.12

- 说说你认为项目中技术最薄弱的一个地方（答了 IO 网络监控和通信模块，短连接性能太差）
- 举例说说在什么情况下会出现性能瓶颈，如何优化（答了用 NIO 的方式）
- NIO 的原理，jdk 中有哪些工具和类去实现，如何实现（selector 和 channel 的用法）,真的好用吗？还可以用什么？（面试官应该是想问 netty，因为没有实际用过，只能给他讲了 netty 的原理）
- 那来说说 AIO 吧，和 NIO 什么区别（对异步的理解）,AIO 在工程中如何实现的？（大概说了下 ajax 的回调函数），又问回调函数具体是怎么实现的（传递函数指针）。
- 然后借着异步 IO 想问消息队列，讲了一下几种模型和原理。（面试中没有用过没关系，只要你懂原理还是可以跟面试官讲，起码可以证明你是爱学习的）
- 项目中非技术上的困难（和甲方沟通需求，没有规范化的项目设计，需求变更太频繁等），问了我解决的方法还有以后希望怎么改进。（变相问互联网公司里面各个 team 以及需求方是如何合作和分工的）
- 讲讲 Spring 中怎么对初始化的 bean 做其他操作。（这里有三种方式，@PostConstruct 注解方式，init-method 的 XML 配置方式，InitializingBean 接口方式）
- 三种实现上有什么区别（还好看过点源码，其实前两种是一个意思，都是通过反射的方式用 aop 思想实现，可以消除对 spring 的依赖；接口方式是直接调用 afterPropertiesSet 方法，效率更高点。spring 加载 bean 时先判断接口方式，再执行配置注解方式）
- 算法题，一个先减后增的数组，查找目标值。（这里并不是查找最值，也不是剑指 offer 上的旋转数组，但是思想上也可以用二分的方式）
- 算法题，两个大数求和，要按高到低位的输入，实时输出结果的对应位，空间 0，时间 O(n)，不借助工具类。（要考虑实时的进位标识，以及多个 9 之后的连续进位标识）

两面完了电话让去参加新锐的现场终面，很有诚意地报销了所有的花销。新锐的三面还是有难度，基本上围着算法在问。

三面 5.12

- 算法：int 范围的随机数的阶乘编码实现。
  （这个题如果直接按最简单的算法题肯定是不行的）

  1.首先考虑要用字符串做运算(因为中间数太大了，只有 String 能保存，当然你可以借助 BigInteger 或 BigDecimal 类去辅助实现)。
  2.阶乘直接计算代价太大，循环太多，考虑设计中间缓存。（正常算复杂度太高，本身就是阶乘级的，所以正常想到用时间换空间）
  3.只用空间换时间的话缓存也不能覆盖全部，如果把所有的中间值保存，空间是 eb 级别，不现实。（这里就要达到一个空间和时间复杂度的平衡点）
  4.存部分中间值用部分空间换取时间，达到空间复杂度和时间复杂度的最优平衡。（开始说的二分做分割存储之后改为等间隔做分隔存储，间隔选取多长为好？我觉得要首先确定空间复杂度的接受极限，然后尽可能减小时间复杂度，因为空间复杂度是可以有预估值的，而时间复杂度当然我们是希望约小越好的）

（这里说一下，我并不是一开始都想到了，只是面试官一直在提示我思路，给我时间思考，没有否定过我）

- 因为头一次手写白板，返回类型有错误，面试官说你这个编译器会提示什么？
- 又问了异常体系，checked unchecked 虚拟机原理怎么做。
- 解释下 iaas.paas.saas 和之间的关系，外呼接口和服务怎么调用的。
- 数据库主从备和读写分离原理，ibatis 怎么配置。（这个只讲了数据库层面的原理，比如监听线程，主机和从机的同步方式等，但是具体代码层面的配置，由于没亲自做过，就说不太知道。）
- 算法，股票最长增长区间，优化
- 算法，最长递增子序列，一个 dp 数组一个 max 数组，最优情况

ps：这个面试官应该是面试过程中遇到最 nice 的一个，也是我现在的老大。其实面试除了自身的因素也有面试官的因素，一个好的面试官不会随便地否定和质疑你（当然有专门压力面的），而是可以让你在放松的环境下，挖掘你真正对于一个方面的深度和理解。最后的十几分钟他并没有问我问题，只是在跟我聊天，他跟我说不管是哪个公司，真正的发展还是跟部门的方向和氛围有关系，选择的时候不要只看公司，做的业务部门方向和 leader 才是该去了解和考虑的。作为应届生很多时候不那么了解，这就要靠我们（指面试官）多去了解你想发展的方向。然后聊了很多成长路径和规划的事。

真正实习到现在一个多月，深深觉得面试就是面试，很多知识和题目都是可以准备的，而工作中面对各种情况解决问题的能力和方式才是更重要的。为了面试准备了很多，工作了发现要学的东西更多，我们真的还有很长的路要走。

美团（123 面）

1 面 1hour 5.26

- java 基础，从头到尾问了个遍，都是大家准备的，但是也挺深的，包括：
- hashMap，红黑树处理冲突，jdk7 和 jdk8 有什么区别
- JUC 相关的集合，ConcurrentHashMap jdk7 和 jdk8 的区别，Collections.sort 函数 jdk7 和 jdk8 分别怎么实现的。（总感觉这个面试官在某段时间肯定纠结过两个版本）
- CopyOnWriteList 底层是什么，适用的情况，vector 的特点，实现的是 List 接口吗。
- 并发的问题，线程间通信三种方式
- 锁的膨胀过程，Synchronized 和 Lock 的区别，底层的 monitor 实现和 unsafe 类的 CAS 函数，参数表示什么，寄存器 cpu 如何做）
- volatile cpu 和寄存器层面是怎么实现的。
- 线程池构造函数参数，各种类型的预设池各自的特点，ForkJoinPool 是怎么实现的，多线程等等问了一个遍。
- 为什么匿名内部类的变量必须用 final 修饰，编译器为什么要这么做，否则会出现什么问题

数据库：

- 索引的分类。
- 主键索引和普通索引的区别，组合索引怎么用会失效。
- 索引的前缀匹配的原理，从 B 树的结构上具体分析一下。
- 聚集索引在底层怎么实现的，数据和关键字是怎么存的。
- 组合索引和唯一性索引在底层实现上的区别（这个是整个一面感觉答得不好的一个问题，不太明白面试官想问啥）
- sql 的优化策略，慢查询日志怎么操作，参数含义。
- explain 每个列代表什么含义（关于优化级别 ref 和 all，什么时候应该用到 index 却没用到，关于 extra 列出现了 usetempory 和 filesort 分别的原因和如何着手优化等）
- show profile 怎么使用。

2 面 1hour 5.27 （因为这一面问得很深，所以到现在都记得很清楚）

- 一个 url 到页面全过程（让我能说多详细说多详细，最好从 OSI 七层的每一层去扩展）
- http 的请求头格式（这个真的记不太清了，只说了几个有印象的标志位）
- getpost 区别，post 可不可以用 url 的方式传参。
- 说到了 url 有最大长度，就问长度有限制是 get 的原因还是 url 的原因，为什么长度会有限制，是 http 数据包的头的字段原因还是内容字段的原因，详细说明。（在他一步步追问下答了个差不多）
- 关于幂等性的详细介绍。

- 幂等性是 http 层面的问题吗，还是服务器要处理和解决的内容。（就是看你对幂等性的定性是怎么理解的）
- 后台服务器对于一个请求是如何做负载均衡的，有哪些策略，会出现什么样的问题，怎么解决。（说了一致性 hash 算法，分布式 hash 的特性，具体的应用场景，又非要问我知不知道这个最早在哪个公司使用的...我说这个真不知道。好像是 amazon?）
- 说说 http 的缺点，无状态，明文传输。
- 那 https 是怎么做的，如何实现的？ca 认证机构。
- 然后问我 https ssl tcp 三者关系，其中哪些用到了对称加密，哪些用到了非对称加密，非对称加密密钥是如何实现的。（还好我项目中涉及到了一些加密）
- 关于加密的私钥和公钥各自如何分配（客户端拿公钥，服务器拿私钥）
- 那客户端是如何认证服务器的真实身份，详细说明一下过程，包括公钥如何申请，哪一层加密哪一层解密。
- java 的优先级队列，如果让你设计一个数据结构实现优先级队列如何做？
- 用 TreeMap 复杂度太高，有没有更好的方法。
- hash 方法，但是队列不是定长的，如果改变了大小要 rehash 代价太大，还有什么方法？
- 用堆实现，那每次 get put 复杂度是多少（lgN）
  （思想就是并不一定要按优先级排队列的所有对象，复杂度太高，但每次保证能取最大的就行，剩下的顺序不用保证，用堆调整最为合适）
- **在线编程题**：敲一个字串匹配问题，写了常规代码。问 kmp 的代码思想，最后问了下正则中用的改进后的 BM 算法。（还有个比较新奇的 Sunday 算法，有兴趣的同学也可以看一下）

3 面 hr

- 其实写了 3 面，感觉根本不算面试了，就是随便介绍了下部门，然后商量实习时间(大概补招都这样吧)。因为已经决定去滴滴新锐了，就跟她说可能暑期不能实习，然后说可以秋招再联系。
- 另外美团这家要跟师弟师妹们说一声，投简历一定还是要选事业群的，千万不要选都喜欢，否则就算过了笔试，也会像我这样等两个月大概是补招才会联系到你。

写在最后

其实大家可以从这个过程中看出来，随着时间的推进，自己的知识储备和各方面的经验和能力都是上升的。

总结几句：

- 对自己要有个定位
- 准备永远都不嫌早不嫌多
- 心态差了及时调整
- 面试挂了及时总结
- 这么多年也该逼自己一把了

一两次甚至一系列的失败并不可拍，成功之后反而没有失败总结得透彻，收获的多。**失败的经历会让你不断提升能力，成功的经历会让你不断提升信心。而不管成功失败都会提升你的经验，都会有收获。**

所以不要害怕失败，因为早晚会成功。
愿大家都能拥有满意的结局。

2017.07.21