

NLP 转后端开发，顺利拿到字节实习 offer！！

简单介绍一下我的基本情况，上海双非本 + 985 硕，目前在读研二。本科 + 研究生大部分时间都在搞算法，NLP 相关，今年终于认识到了理想和现实的差距，遂在 5 月份下定决心转后端开发。之后处理一些事情，暑假 7 月底的时候正式开始学习 Java，历时三个月，9 月底开始投日常实习，10 月 17 结束。面试过的公司包括：B 站，蔚来，百度以及字节，B 站挂在二面，其他均拿到日常实习的 Offer，最终选择了字节跳动。

从以上我的基本情况大家能看出，我的战线比较短。但是在这些时间里，我的作息基本上都是早 7 晚 11，很辛苦。

这篇文章首先分享我的面试经历，后面的部分与大家分享我的学习经验。

面经部分

下面的部分总结了面试过程中被问到的知识点(还记得的部分)，以及我个人的一些心得体会，供大家参考。

百度

百度给我的面试体验还是蛮好的，是我面试的所有公司中唯一一家对八股考察非常详细的。一面是非常详细的八股面试，涉及到 Java 基础知识、数据库、spring、jvm、多线程、场景设计等等，基本上准备的都被问过了，但是整体来说难度不高，知识面比较广但是不会深挖。二面来说区别就非常大了，注重实践能力的考查，而且会深入到底层原理。

面试之前会有自我介绍的环节，包括项目介绍。百度两面对我的项目提问都不多，可能是我介绍项目的时候就比较详细的原因。

接下来的部分是知识点整理：

百度一面(1h+)：

- 常用 GC 算法，常用的垃圾收集器，G1 了解吗
- 场景题：cpu 打满且频繁 full GC，怎么解决？
- 有 jvm 调优的经验吗？实际工作中遇到过内存相关的问题吗？用过哪些堆栈工具调试？
- Mysql 索引，数据结构为什么使用 B+ 树
- 索引覆盖了解吗
- 索引失效的场景
- 简单描述一下数据库的四种隔离级别以及对应的三种相关问题
- MVCC + 锁 保证隔离性
- 造成幻读的原因了解吗，快照读、当前读。
- 数据库自增 ID 和 UUID 对比
- HashMap 源码，数据结构，如何避免哈希冲突，对比 HashTable
- HashMap 源码中，计算 hash 值为什么有一个高 16 位和低 16 位异或的过程？
- 为什么重写 equals 还要重写 hashCode，不重写会有什么后果
- ConcurrentHashMap 底层实现，扩容问题。
- 如果让你自己实现哈希表，你会考虑什么问题？
- 场景题：亿级黑名单、短链接，你考虑使用什么数据结构？布隆过滤器、前缀树。其中布隆过滤器问了基本的原理和实现方式
- Java 引用类型，强软弱虚
- Java 是引用传递还是值传递
- Object 类你了解哪些方法
- 接口和抽象类的区别
- 线程池核心参数，以及工作原理
- ReentrantLock 对比 sync 锁
- lockInterruptibly()、acquire()、tryAcquire() 方法
- CAS 机制了解吗，存在什么问题
- 对象锁和类锁的区别
- 如果让你自己实现阻塞队列，如何实现？阻塞唤醒这一部分，如何实现？
- ThreadLocal，Volatile
- 看你项目中用到了 Netty，简单介绍下吧。这里还有个问题是问到 Netty 和 SpringBoot 整合的，但我一直都没理解她想问什么
- 粘包拆包问题，Netty 解决粘包拆包的 Decoder
- Spring 事务了解吗，Spring 事务的注解不生效，是什么原因
- 算法题：手写快速排序，时间复杂度，稳定性

整理感觉不错，基本都答上来了，按照 Guide 哥星球里的内容，认真准备就好。

百度二面(45 min):

- 看你项目中用了一致性哈希做负载均衡，简单介绍一下
- 项目中 CompletableFuture 如何使用的
- 算法题：给定一个字符串，找到其中最长回文串
- 计网和组成原理学过吧，你认为哪个掌握的好？ 我选了计网。这一部分问的很深入，我没来得及记录，以下部分只是一些零散片段，但是整体问的时候是有逻辑的。
- OSI 七层模型
- TCP 三次握手，四次挥手整个过程包括状态的转换。为什么是三次握手、四次挥手。发送 Fin，实际的意义代表什么？（发送方没有数据要发送了，可以断开连接）
- 四次挥手，为什么等待 2 MSL
- 流量控制、拥塞控制
- 后面关于网络就更深入了，TCP 底层是怎么实现的，如果让你用 Java 模拟 TCP 的过程，做一个仿真，你有什么想法。大学学习计网的时候，协议栈之类的了解过吗（这部分我都不懂，认栽了）
- 看你项目中用到了 Spring，自动装配的过程了解吗。
- Spring 启动类的注解，介绍一下
- 因为我项目中用到了，所以被提问了 Spring 二次开发常用的扩展点，还涉及到了 Bean 的生命周期。BeanPostProcessor，在你项目中如何使用的
- Spring 中你常用哪些注解？Autowired 实现原理

计网仿真 TCP 以及后面深入的部分我不懂，Autowired 实现原理 没说清除，其余的都答上来了。

字节跳动

在我整个的面试过程中，字节给我的体验是最好的。一面二面的面试官都非常好，面试的问题、要求都说的很清楚，需要注意的点都提前告诉了我，甚至二面的面试官会提醒我，“在回答问题的时候这边会有敲键盘的声音，是我在记录，不要影响你回答问题。”对于初次求职面试，体验感拉满。

除此之外，字节的面试和百度思路不一样。百度是从八股出发，引出一些实际场景遇到的问题。字节几乎没有八股，是从项目出发，结合工程经验，主要考察思考的过程，关键点答出来之后，结果对错可能不是很重要（这里是我主观臆断的）。

字节一面(1h):

- 自我介绍，项目部分主要介绍了 rpc 项目，后续的问题都是基于这个项目
- rpc 远程调用的整个流程
- 项目中的 SPI 机制，介绍一下原理以及你做了哪些改进
- 项目中用到了负载均衡算法，详细介绍一下
- 一致性哈希的原理，虚拟结点
- 项目中的序列化方案，为什么序列化，你都了解哪些常用的序列化方法。
- 你项目中使用了 Kryo 序列化，优点你提到了，缺点了解吗
- 通信协议是你自己设计的，假如后面需要变更，比如添加新的字段，你项目中如何处理的？
- 服务的灰度发布介绍一下，如何实现的？
- Zookeeper 作为注册中心，假如崩溃了怎么办？这里开始连环问了
- 你提到了 Zookeeper 的一致性，它是如何保证的？
- ZAB 协议，选举的过程，这里问的很详细
- Zookeeper 是强一致性吗？
- 网络分区了解吗，CAP 理论
- Zookeeper 如何应对网络分区的，脑裂问题了解吗，如何解决？
- 假如我同一时间有大量服务发布，你提到了 Zookeeper 只有主节点负责写，怎么解决？假如主节点崩溃了，新选举出的主节点仍然没办法面对我的大流量，也崩溃了，如何解决？
- MQ 的原理，你知道哪些 MQ，各自有什么特点，什么时候需要用 MQ
- 你刚才提到了服务端保护机制，如何实现的？这里我答了限制连接数以及接口限流，基于责任链模式。之后问了用到的令牌桶以外的常用限流算法。
- 算法题：链表反转，你知道的所有实现方式。这里我写了递归和非递归两种。

整个面试的过程中大脑都是高速思考的，甚至从面试官的问题中得到了好多启发，是背八股掌握不到的，体验非常好。面试的问题几乎都答上来了，有一些不太熟悉的在面试官的提醒下也都回忆起来了（这点非常 nice，其他面试不会就直接过了，而字节会认真引导你，看你究竟掌握到什么程度）。事后 hr 小姐姐还告诉我面评非常好，鼓励我认真准备二面。

字节二面(1h):

字节二面的经历比较魔幻了，面试官在伦敦有时差，因此是晚上九点开始面试的，由于面试官比较忙，整个面试过程比较简单，自我介绍 + 项目介绍之后简单提问了几个问题（没有印象深刻的技术问题，这里就不重复整理了），沟通了一下实习时间，直接做算法题了。

- 算法题：有一个 $n * n$ 的棋盘，每个格子有 RB@ 三个状态，R 表示红色，B 表示蓝色，@ 表示此路不通。机器人从左上角走到右下角，每次只有上下左右四个方向选择，相同颜色之间没有代价，跨越不同颜色代价为 1，求解机器人从左上角走到右下角，最少的代价。
- 这里我用回溯求解的，很快就写出来了，思路也没问题。但是复杂度计算卡住了，在面试官多次且反复的提示下，算出来了。最后提问环节面试官跟我说后续优化可以加一些剪枝操作

二面没有遇到难度比较大的问题，大部分时间都被我卡在了算时间复杂度(很菜勿喷)....

B 站

综合来说，B 站面试给我的体验是非常差的，一面的时候我感觉还没进行比较深入的交流，问了一些八股，很快就结束了。二面上来之后，问了你觉得自己项目有什么亮点吗，然后就来了一个 Hard 算法，又结束了.....

B 站一面(30 min)

- Rpc 远程调用的流程
- 一致性哈希算法详细介绍
- 为什么选用 Zookeeper 作为注册中心，注册中心作用是什么
- 动态代理
- Redis 在你项目中如何使用的，穿透、雪崩、击穿了解吗
- 你项目中用的是 RabbitMQ，为什么，和其他 MQ 对比如何？
- RabbitMQ 的原理
- 你项目中的 灰度发布、分组管理如何实现的
- 无算法题

之后在我以为他准备深入提问的时候，面试官告诉我面试结束了，整个过程不到 30 min，而且没有算法题。

B 站二面(30 min)

- 简单介绍下你的项目，是工程项目、学校项目还是自己学习的
- 你认为项目中有什么亮点？
- 算法题：K 个有序数组，输出最终排序后的数组（K merge）。

B 站是我第一个走面试流程的公司，二面在字节一面的前一天。B 站二面是我第一次在面试过程中写算法题，结果就遇到个这，当时心态是崩掉的。因为第一次确实有些紧张，思路不清晰。虽然是力扣 Hard 难度吧，但是事后觉得也没有很难，做不出来还是大多归因于自己。但是复盘的过程中，我发现在面试的过程中，虽然太紧张了没实现出来，我把两种解题思路都思考到了，并且面试官提问时间复杂度，在提示下也求解出来了(很菜勿喷)，整体表现自我感觉也算可圈可点吧。

给大家的经验就是，平时刷题的时候时间复杂度求解一定要重视！

参考答案

由于面试题目太多，这里就不一一列举参考答案了。

你可以在下面两份参考资料中找到上面大部分问题的参考答案：

- JavaGuide 在线阅读版：<https://javaguide.cn/> <<https://javaguide.cn/>> 。
- 《Java 面试指北》<https://mp.weixin.qq.com/s?__biz=Mzg2OTA0Njk0OA==&mid=2247533385&idx=1&sn=105fdab847e4cf93d2603e57461658be&chksm=cea10c82f9d685944dc09010e5354566f8a8a40adce7cdcd3049444f7f83d4d084c0ef072542&token=766741944&lang=zh_CN&scene=21#wechat_redirect> 。

备战部分

下面是我从七月底写出第一行 Java HelloWorld 到现在的大致时间表：

- 七月底，正式开始投入时间学 Java，在师兄、师姐的推荐下选择了 JavaGuide 作为主线的学习资料，之后加入了[知识星球](https://mp.weixin.qq.com/s?__biz=Mzg2OTA0Njk0OA==&mid=2247530558&idx=1&sn=cb2ed58af806ee3d3d174ce5ae09c7e3&scene=21#wechat_redirect) <https://mp.weixin.qq.com/s?__biz=Mzg2OTA0Njk0OA==&mid=2247530558&idx=1&sn=cb2ed58af806ee3d3d174ce5ae09c7e3&scene=21#wechat_redirect>，认真阅读了关于学习路线的内容。
- 七月份用了一周多的时间熟悉 Java 语法。前期主要跟随 小码哥 恋上数据结构课程，一边复习算法数据结构，一遍熟悉 Java 语法。
- 八月份开始，选择了 Guide 哥推荐的千峰商城项目作为入门，大概用了两周时间，全程跟随视频敲完代码，收获非常大。这一过程中，恋上数据结构这门课程是同步学习的，基本就是早晚做项目，下午学算法。
- 多线程、JVM 方面的知识，我选择了马士兵的课程。这两部分是同时学习的，理解为主。选择马士兵课程的原因之一，是因为马老师讲课是以面试为导向的，一边理解一边掌握八股文了，效率比较高，总耗时两周左右。
- 网络 IO 部分的知识 以及 Zookeeper、MQ 等中间件，这三部分是一起学习的，参考的资料包括马士兵课程、稀土掘金的 Zookeeper 课程、慕课网的 MQ 课程以及尚硅谷的一些资料，耗时大概一周左右吧。
- 以上打基础大概花费了一个月的时间。

- 九月份返校之后，开始着手准备简历上的项目，花了一些时间在 Guide 哥的知识星球里翻看优秀开源项目介绍，选择了 Guide 哥的手写 Rpc 项目以及星球推荐的 IM 项目。
- 有了基础之后，项目做起来还是比较快的，加上有源码可以参考，各自用了一周就基本实现完成了。这里总共耗时两周。
- 后面的时间里，我针对这两个项目做了深入研究。在**极客时间**和**稀土掘金**里，我分别找到了 rpc 和 IM 的相关课程。由于都是文字的形式，加上自己实现过基本功能，读起来非常快，快速整理出了课程内作者对于项目深入思考的部分，之后融合到自己的项目中。这里我认为是非常关键的一步，在面试的时候我能够顶住面试官的连环问，和这些课程中的相关内容以及思考题的深度是分不开的。在这一过程中，我还有幸加了几个作者大大的微信，不停地和作者交流自己的思考，甚至发现了课程中的一些小瑕疵。非常幸运他们都很有耐心，给予了我很多指导，尤其是 crossoverjie 大佬，几乎是有问必答，甚至在我面试之前，还给予了我很大鼓励（相当感动）。
- 大概到九月中旬，我就开始整理简历并且投递了，之后一边复习八股，一边完善项目。
- 九月底，我的项目已经基本整理完成了。我花了一周的时间系统梳理八股文，制作了很长的脑图帮助我回忆知识点。
- 十月份，国庆节的假期里，我保持着每天 15+题的速度，快速找回了算法题的手感。由于时间真的太仓促了，时间复杂度这一块我没有重视，后面也付出了惨痛的代价。提醒各位读者，算法复杂度的计算一定要重视起来。
- 最终功夫不负有心人，我收获了百度、蔚来以及字节的日常实习 offer。JavaGuide 以及[知识星球](https://mp.weixin.qq.com/s?__biz=Mzg2OTA0Njk0OA==&mid=2247530558&idx=1&sn=cb2ed58af806ee3d3d174ce5ae09c7e3&scene=21#wechat_redirect) <https://mp.weixin.qq.com/s?__biz=Mzg2OTA0Njk0OA==&mid=2247530643&idx=2&sn=cf4855e15ea7c8a669c7896b8337292b&scene=21#wechat_redirect> 内部的 [《Java 面试指北》](#) <https://mp.weixin.qq.com/s?__biz=Mzg2OTA0Njk0OA==&mid=2247530643&idx=2&sn=cf4855e15ea7c8a669c7896b8337292b&scene=21#wechat_redirect> 在我整个备战的过程中起了很大作用，是我的指路明灯。

写在最后

由于从本科开始就一直做算法 NLP 相关的工作，对后端开发了解甚少，加上时间紧迫，我不得不采取一种囫圇吞枣的方式进行学习。对我而言，在不到三个月的时间里，从 Java 的入门阶段到通过日常实习面试实在付出了太多，每天早 7 晚 11 的作息时间对身体也产生了一些伤害，如果时间允许，我更希望节奏慢下来，把每个知识点都学扎实、学透彻。

本篇面经实际上没有太多东西可以分享给大家，因为笔者实际上也只是一个才学了不到三个月 Java 的新手小白。如果说文章里有什么是值得大家参考的话，我希望是面对目标绝对坚持的毅力以及面对困难永不退缩的决心，是它们支撑着我逐渐越过一个又一个的“不可能”。