

# 测试

2023年8月18日 11:35

base上海，研二

本科天坑专业，硕士电子信息专业 强化学习方向(根本没搞懂)

非科班

学习Java一年左右 基本上跟着网上教程学习的

投递暑期实习情况：

- 1 阿里巴巴 一面挂
- 2 美团 四面后挂
- 3 字节跳动 三面+HR面 已offer(转 go)
- 4 携程 测评挂(有点离谱)
- 5 拼多多 二面结束 拿到字节offer后放弃了主管面
- 6 华为 二面结束 目前在池子
- 7 京东 笔试挂(笔试完转推荐了)

给面试机会的基本上都到了三面(除了阿里)

阿里1面：

- 1 说一下什么是泛型，使用场景，泛型的好处
- 2 如何去除代码中的很长的if-else语句？
- 3 什么是反射，好处是什么，什么地方用了反射，项目里怎么用的
- 4 Spring中Bean的生命周期，什么时候销毁？
- 5 多个人抢一张票，如何防止超卖问题(给多个解决方案)？
- 6 做项目之前会做一些设计吗？
- 7 数据库和缓存一致性问题？

总结：问了很多Spring相关的，然而我Spring八股没背，确实答的不好，直接g

美团1面：

- 1 介绍项目 项目相关问了十几分钟，后面问题也和项目相关
- 2 数据库字段，如何设计
- 3 redis bigkey问题
- 4 redis和mysql数据一致性问题
- 5 mysql索引失效的情况
- 6 rabbitmq如何保证消息一定被消费一次
- 7 mq 消息积压如何处理
- 8 算法题：easy题，给定一个链表的某个节点，删除此节点

美团2 面：转部门了，相当于一面

基本上是结合项目和八股来问

- 1 为什么不找实验室项目相关的工作，而找互联网方向
  - 2 项目中redis缓存了文章id还是文章的内容？为什么？回答没有缓存文章信息，只是缓存的热门文章id
  - 3 如果缓存文章内容会出现什么问题？ 回答bigKey
  - 4 如何解决上述缓存文章的bigkey？ 回答文章分段
  - 5 如何确定一个文章有几段，并取出对应的文章段？ redis中用什么数据结构来存放？
  - 6 bigkey除了value很大的场景会出现，还有什么情况会造成bigkey？ 回答 列表或者hash中存放的数据个数较多
  - 7 网站统计的PV还是UV？ 回答UV，如何统计的？
  - 8 如果访问量非常多，如何统计UV？ 回答可以考虑bitmap或者hyperloglog
  - 9 hyperloglog底层原理知道吗？ 回答基于伯努利试验的概率统计，是一个约数，可能存在误差，具体细节没懂
  - 10 用了Mysql吧，mysql事务的完整性通过什么保证？ 回答基于undolog，mvcc，binlog，redolog，各种锁机制 没有细说
  - 11 mysql中char和varchar的区别？
  - 12 如果数据长度在10个以内，且经常改变，长度也可能改变，用char还varchar？ 从mysql底层数据存储结构考虑
  - 13 Java中的同步机制？ 回答synchronized和ReentrantLock
  - 14 详细介绍上面两个，锁的升级 和 ReentrantLock底层的AQS
  - 15 二者各有什么优缺点？
  - 16 volatile了解吗？ 从MESI + 总线嗅探机制 + lock前缀指令方面做了介绍
  - 17 Java里面的引用？ 强 软 弱 虚
  - 18 各在什么地方使用？ 只回答了强引用 和 弱引用(在ThreadLocal中使用，可以在一定程度预防内存泄漏)
  - 19 讲一讲ThreadLocal如何在一定程度防止内存泄漏的
  - 20 ThreadLocalMap中 key为什么设置为弱引用，value可以设置为弱引用吗？
  - 21 项目里用了前缀树，为什么要用前缀树，hash可以吗？ 回答：基于空间和时间的平衡做考虑，hash的空间占用大
  - 22 项目里面redis和数据可能会出现数据不一致的情况吗？如何解决
  - 23 先更新Redis缓存再修改数据库可以吗？和先修改数据库有什么区别？
  - 24 项目里面用了mq做通知，为什么用mq？用mq的场景有哪些？ 回答在数据一致性和用户体验上做平衡。
  - 25 算法题：简单题，判断镜像二叉树。
- 反问：部门业务，主要语言，是否有人带，实习生培养制度

美团 3，4 面：

3面忘记了，主要还是八股为主，面试官全程在听我背八股

4 面有点像技术面和HR面，问了项目难点，逻辑题，条件概率，未来规划

总结：美团面试体验还不错，难度适中，特别是算法题比较简单，最后面完10天还是挂了，其实感觉回答的还行

字节1面：

一面八股为主

1 自学平时如何学习

2 介绍一下项目

3 项目中表的结构

mysql和es数据同步问题

4 Mysql索引建立原则

5 聚簇索引和非聚簇索引

6 binlog, redolog, undolog作用和区别

7 数据库事务特性acid

8 数据库事务隔离级别

9 http get和post区别

10 tcp和udp区别

11 http2.0优化

12 leetcode 1129题 不会

13 leetcode 53 子数组最大和 (进阶: 如果可以选择删除数组中的一个元素或者不删除, 最大和是多少)

字节二面:

1 非科班的, 平时怎么学?

2 进程线程区别, 线程有自己的私有数据吗?

3 redis基本数据类型

4 redis淘汰策略 持久化策略(aof,rdb优缺点)

5 lru如何实现, redis中是如何实现的?

6 topk问题, 返回时无需按序返回, 代码, 要求O1空间复杂度, 时间复杂度尽可能低

7 代码的时间复杂度是多少

8 如果数组数据量非常大怎么办? 如果k也很大怎么办?

9 写两个SQL(好久没写sql差点全忘了, 最后也没运行) 网上应该可以找到原题

两张表

student表, 有id和name字段

course-score表, 有学生idsid, 课程courseid和score三个字段

1-返回总分最高的三个学生姓名 (每个学生课程数量可能不一样)

2-求所有课程都及格的学生数量

字节三面: leader面

果然还是三面压力最大

看别人的三面都是深挖项目, 或者设计题和编程题

我这三面应该算简单的了

自我介绍, 说你最擅长的知识, 如果只是简单使用过就不用说了

我说对计算机网络和操作系统这块了解还可以。

面试官问我自学计网和操作系统时遇到什么比较晦涩的知识, 让我自己说。然后就深入探讨了MSS,MTU,

子网掩码的作用 (为什么有子网掩码, 在哪一步起作用), 路由选择协议,

osi各层之间能不能反过来封装(如果反过来封装有什么问题),  
ip地址和mac地址什么的(如果缺少mac会怎么样, 缺少ip地址呢?)  
dns解析的流程, 用的什么协议,  
就是比八股问的更深一点, 如操作系统为什么要有虚拟内存,  
虚拟地址到物理地址的转换过程,  
内存分段和内存分页的好处和缺点  
为什么说线程是调度的基本单位(面试官说这其实挺难回答的, 我说了自己的理解, 面试官说你说的结论, 而不是原因, 说等我工作几年就理解了),  
还有volatile具体功能怎么实现的(要讲到cpu的mesi, 总线嗅探, storebuffer, lock指令等)。  
算法题也没怎么写, 可能面试官就看你是不是对于知识有更深层次的理解, 而不是只是简单的背八股文。  
如果对于底层了解比较深的同学三面可能就比较轻松了, 我对底层了解也就半瓶水。

总结: 可能是因为语言不一样, 字节很少问语言和项目。主要还是问的基础, 幸运的是算法题没给我出困难, 要不然就gg。

pdd总结:

pdd面试忘记做记录了。

主要问了Java语言, JVM, 垃圾回收器等。

算法题: leetcode 117 不能用迭代, 要用递归

算法题: 给定两个有序链表, 有重复数字, 将链表合并后任然有序, 并且每个数字重复次数不能超过两次

总结: 总的来说, 对于Spring等框架问的不太多, 除了阿里其他的都是简单问了一下, 个人感觉出题频率高的

MySQL, Redis, 计算机网络, 操作系统, 项目相关, 语言, 数据结构

功能测试:

当搜索框为空时, 点击搜索按钮或按下回车键, 应该返回搜索结果页面。

当输入一个有效的关键词并点击搜索按钮或按下回车键, 应该返回相关的搜索结果页面。

当输入一个无效的关键词并点击搜索按钮或按下回车键, 应该返回搜索结果为空的面或者相关提示信息。

边界测试:

输入搜索关键词的最大长度(例如512个字符), 确保搜索功能正常工作。

输入搜索关键词的最小长度(例如1个字符), 确保搜索功能正常工作。

输入包含特殊字符、数字和中文字符的关键词, 确保搜索功能正常处理不同类型的输入。

富输入测试:

尝试输入带有拼写错误的关键词, 查看是否提供拼写纠正的建议。

尝试输入一个常见的问题或问题描述，查看搜索结果是否能够提供相关答案或解决方案。  
尝试输入一个长尾关键词，查看搜索结果是否能够准确匹配用户的意图。

#### 兼容性测试：

在不同的浏览器（例如Chrome、Firefox、Safari、Internet Explorer等）上测试搜索功能，确保在各种浏览器中正常工作。

在不同的操作系统（例如Windows、Mac、Linux等）上测试搜索功能，确保在各种操作系统中正常工作。

在不同的设备（例如电脑、手机、平板电脑等）上测试搜索功能，确保在各种设备上正常工作。

#### 性能测试：

测试搜索框的响应时间，确保搜索框能够快速响应用户的输入。

测试搜索结果页面的加载时间，确保页面加载速度快且流畅。

#### 安全性测试：

尝试输入恶意代码或特殊字符，查看搜索框是否能够正确过滤和处理，以防止XSS攻击等安全漏洞。

#### UI测试：

测试搜索框在不同分辨率下的显示效果，确保搜索框能够自适应不同屏幕大小。

测试搜索框在不同浏览器下的显示效果，确保搜索框的样式、布局等在各种浏览器中一致。

#### 可用性测试：

模拟不同场景下的用户操作，如点击搜索框、输入关键词、选择关键词提示等，确保搜索框的交互体验流畅。

测试搜索框是否提供自动补全功能，以减少用户的输入量。

测试搜索框是否提供搜索历史记录功能，以方便用户查看之前的搜索记录。

功能、界面、兼容性、安全性、性能压力、易用性、弱网无网测试、场景组合测试