作为一个双非渣硕，历经两个月的时间，面试了大大小小公司的 Android 实习生岗位，最近终于结束了面试状态，决定好好把面试问题以及相关经验整理下来，顺便附带自己的学习经验与准备过程，攒攒人品，为秋招再战。

## 前言

2016 年开始接触 Android，从刚开始接触就不断地听到 Android 市场饱和，工作难找等消息。虽然当时也非常迷茫，不过由于第一次深入接触编程语言，再加上自己的一点兴趣，就一直坚持下来了。

通过两个月的面试经历，确实发现 Android 岗位比较少，而且通常要求比较高，不仅需要 Android 开发经验，往往还需要会 React Native,JavaScript 等，甚至还期望你能具有 IOS 开发经验。

不过作为应届生还是有些优势的，那就是一些一线的互联网公司还是比较看中个人基础 以及发展潜力的，所以如果能在自己的专业方向上具有扎实的基础，1-2个实际开发项目以及个人的兴趣，还是能够找到一个满意的 Android 岗位的工作的。目前这些素质，自己也很欠缺，通过下面的面试经历就可以看出来，不过最起码有个努力的目标，可以好好准备为秋招做准备。

## 面试经验

自己大大小小投了也有 20 多家公司，不过经历简历筛选以及笔试淘汰，最终就经历了7家公司的面试。下面我就把自己面试中问到的问题贴出来供大家参考，一些具体项目相关的就不贴了。

### 阿里巴巴

阿里是 3 月初开始投的，是自己第一次面试大型的互联网公司，当时自己的准备也不够充分，表现不是很好，经历了三次技术面，最后挂了。

#### 阿里一面

* 排序，快速排序的实现
* 树：B+树的介绍
* 图：有向无环图的解释
* TCP/UDP 的区别，滑动窗口，如何确保有效性
* volatile
* synchronized 与 Lock 的区别
* Java 线程池
* Java 中对象的生命周期
* 类加载机制
* 双亲委派模型
* Android 事件分发机制
* MVP 模式
* RxJava

#### 阿里二面

* 抽象类和接口的区别
* synchronized 与 lock
* 集合 Set 实现 Hash 怎么防止碰撞
* JVM 内存区域 开线程影响哪块内存
* 垃圾收集机制 对象创建，新生代与老年代
* 二叉树 深度遍历与广度遍历
* B树、B+树
* 消息机制

#### 阿里三面

* 项目介绍
* 项目中做了些什么？主要解决的问题
* 为什么选择 Retrofit，RxJava
* RxJava 的特点
* 进程调度
* 进程与线程
* 死锁
* 进程状态
* JVM 内存模型
* 并发集合了解哪些
* ConCurrentHashMap 实现
* CAS 介绍
* 锁 synchronized，lock
* 开启线程的三种方式，run() 和 start() 方法区别
* 线程池
* 常用数据结构简介
* 判断环
* 排序，堆排序实现
* 链表反转
* 海量数据 字典查找
* 平时看什么书

### 网易游戏

网易游戏当时投的时候就没抱希望，招聘信息上明确指定只招固定的那几所985高校，就随便投了，没想到笔试都没做就直接打电话面试了，不过问的问题确实很深入，结果显然，一面就挂了。

#### 网易游戏一面

* 集合
* concurrenthashmap
* volatile
* synchronized 与 Lock
* Java 线程池
* wait/notify
* NIO
* 垃圾收集器
* Activity 生命周期
* AlertDialog,popupWindow,Activity 区别

### 远景能源

远景能源是一家互联网能源公司，给出的实习待遇是相当好，经室友推荐就投了简历。最后流程走完，得知挂在了二面上，大概原因就是没有拿得出手的项目，实际项目经验不足。

#### 远景电话面

* 四大组件
* Android 中数据存储方式
* 微信主页面的实现方式
* 微信上消息小红点的原理
* 做的项目，一个印象深刻的问题
* 看的技术博客，印象深刻的

#### 远景现场一面

* 两个不重复的数组集合中，求共同的元素。
* 上一问扩展，海量数据，内存中放不下，怎么求出。
* Java 中 String 的了解。
* ArrayList 与 LinkedList 区别
* 堆排序过程，时间复杂度，空间复杂度
* 快速排序的时间复杂度，空间复杂度

#### 远景现场二面

问的都是一些项目问题，比较宽泛，没问具体技术点

### 今日头条

今日头条是在四月初投的，当时找了一个月，都没拿到拿得出手的 offer，有点心烦意乱，就又海投了一波。4.18 做了今日头条的面试，4.25 进行的视频面试。一共进行了 3 轮视频面试，头条的面试官很好，看的出来头条的技术是很强的，也很注重算法。三轮技术面后，hr 通知通过，给了个秋招终面直通卡，具体发不发 offer 两周内答复。这两天已经陆续开始发 offer 了，目前还没收到，可能备胎了吧。

#### 今日头条一面

* 数据结构中堆的概念，堆排序
* 死锁的概念，怎么避免死锁
* ReentrantLock
* synchronized
* volatile
* HashMap
* singleTask 启动模式
* 用到的一些开源框架，介绍一个看过源码的，内部实现过程。
* 消息机制实现

#### 今日头条二面

* synchronized 与 ReentrantLock
* ReentrantLock 的内部实现
* 用到的一些开源框架，介绍一个看过源码的，内部实现过程。
* Java 中异常
* App 启动崩溃异常捕捉
* 事件传递机制的介绍
* ListView 的优化
* 今日头条推荐新闻去重，推荐的时候去掉用户已经看过的新闻。
* 二叉树，给出根节点和目标节点，找出从根节点到目标节点的路径。手写算法
* 模式 MVP，MVC 介绍
* 断点续传的实现

#### 今日头条三面

* 集合的接口和具体实现类，介绍
* TreeMap 具体实现
* synchronized 与 ReentrantLock
* 手写生产者/消费者模式
* 逻辑地址与物理地址，为什么使用逻辑地址
* volatile
* 一个无序，不重复数组，输出 N 个元素，使得 N 个元素的和相加为 M，给出时间复杂度、空间复杂度。手写算法
* Android 进程分类
* 前台切换到后台，然后再回到前台，Activity 生命周期回调方法。弹出 Dialog，生命值周期回调方法。
* Activity 的启动模式

### 触宝科技

触宝科技是一家上海的互联网公司，是通过实习僧上简历投递获得的这次面试机会，一共进行了两轮电话面试，最终获得了 offer。触宝科技的 hr 人很好，时间观念很强，整个流程走的比较顺利。

#### 触宝科技一面

* RxJava 的作用，与平时使用的异步操作来比，优势
* Android 消息机制原理
* Binder 机制介绍
* 为什么不能在子线程更新 UI

#### 触宝科技二面

* JVM 内存模型
* Android 中进程内存的分配，能不能自己分配定额内存
* 垃圾回收机制与调用 System.gc() 区别
* Android 事件分发机制
* 断点续传的实现
* RxJava 的作用，优缺点

### 爱奇艺

爱奇艺也是通过实习僧上简历投递获得的机会，本来不抱希望，结果过了 10 天左右约我面试。面了大概一个小时，聊得还不错，最后第二天通知我挂了，有点不知所措，可能是实习时间达不到要求吧（只能这样安慰自己了）。

#### 爱奇艺一面

* RxJava 的功能与原理实现
* RecycleView 的使用，原理，RecycleView 优化
* ANR 的原因
* 四大组件
* Service 的开启方式
* Activity 与 Service 通信的方式
* Activity 之间的通信方式
* HashMap 的实现，与 HashSet 的区别
* JVM 内存模型，内存区域
* Java 中同步使用的关键字，死锁
* MVP 模式
* Java 设计模式，观察者模式
* Activity 与 Fragment 之间生命周期比较
* 广播的使用场景

### 携程

携程是 3 月份投的内推，结果挂掉了，后来通过笔试又获得的机会，笔试完快一个月才收到的通知，本来都快忘记了。既然通知了，就去面了。今天刚去面的，经过 2轮技术面，1 轮 hr 面，第二轮技术面是总监面，主要聊了聊项目上的问题。下周一会根据排名发 offer，现在暂时进入备胎池。

#### 携程一面

* Activity 启动模式
* 广播的使用方式，场景
* App 中唤醒其他进程的实现方式
* AndroidManifest 的作用与理解
* List,Set,Map 的区别
* HashSet 与 HashMap 怎么判断集合元素重复
* Java 中内存区域与垃圾回收机制

#### 携程二面

* EventBus 作用，实现方式，代替 EventBus 的方式
* Android 中开启摄像头的主要步骤
* Github 上传了哪些东西，代码量

## 学习资料

从 Android 开发工程师的角度来讲，我自己主要准备了以下几个方面的内容：

### 1.Java

Java 基础，如集合，反射，注解，IO，NIO，其中集合很重要，面试常问。

JVM，如内存区域，内存模型，垃圾回收机制的算法，收集器，类加载机制。

Java 并发，如 sychronized,lock,volatile，生产者/消费者模式，线程池，死锁。

### 2.Android

Android 基础，如四大组件，Binder 机制，消息机制，事件分发机制，自定义 View 过程。

Android 开源库，如 Retrofit,RxJava 等原理实现，优缺点，以及使用。

### 3.数据结构

链表，栈，队列，排序，树，图，以及其中涉及到的一些算法题目。

### 4.设计模式

单例模式，观察者模式，建造者模式，工厂模式，装饰者模式等。

### 5.操作系统

进程与线程，进程通信，进程调度，分页存储，分段存储，虚拟内存等。

下面介绍以下我看过的一些书籍。

### Java

疯狂 Java 讲义(有些人说不好，自己看着还行吧，可以看核心卷，别人都推荐，我没看过)

Thinking in Java(看了一部分，没看完，建议有一定基础再看)

深入理解 Java 虚拟机(很好的一本书，必看)

Head First 设计模式(非常生动 的讲述设计模式)

Java 多线程变成核心技术(讲述 Java 中多线程的一些问题，比较基础)

Effective Java(78条开发中会用到的实际经验，很好，还没看完)

### Android

Android 群英传(很基础，通俗易懂)

Android 开发艺术探索(面试必备，内容都深入浅出)

### 数据结构

大话数据结构(讲述各种数据结构的概念，算法实现是 C,可以作为入门书籍看)

剑指offer(面试必备，面试的时候好多 上面的题目)

其他的没看了，不过可以推荐一个网上视频课程，讲的很好——[数据结构](http://www.icourse163.org/learn/ZJU-93001?tid=120001)

### 操作系统

现代操作系统(需要耐心的看，书也挺厚，暂时没看完)

推荐个网上课程——[操作系统](http://www.xuetangx.com/courses/course-v1:TsinghuaX+30240243X_2015_T2+2015_T2/courseware/be5b8d4fec0c4c329d19845020bc67b2/39384cfa48f44ee8a99561f69f4db4a5/)

## 总结

找工作是个很辛苦的事情，而且一般周期都比较长，有时候即看个人技术，也看运气。第一次找工作，最后的结果虽然不尽如人意，不过收获远比 offer 大。接下来就是针对自己的不足，好好努力了。

## 关于面经

[杭州找 Android 工作的点点滴滴](http://mp.weixin.qq.com/s?__biz=MzIxNzU1Nzk3OQ==&mid=2247484459&idx=1&sn=857366c0e2d20523c68c63d254b1ebaa&chksm=97f6ba9fa08133891e7ba8ea7e3dd90bbd7802496076dbb7b9c95e42d62f23e96ded7503b2e4&scene=21#wechat_redirect)

[那些IT培训出来的Android工程师，希望你面试时涨点记性](http://mp.weixin.qq.com/s?__biz=MzIxNzU1Nzk3OQ==&mid=2247484193&idx=1&sn=4738d11bc16d0428be1c96b0de827bc4&chksm=97f6bd95a081348393d505ae8a109319a376747f73fce028cf2b902879d3b1c026f86e8f9801&scene=21#wechat_redirect)

[为跳槽的你献计献策（Android）](http://mp.weixin.qq.com/s?__biz=MzIxNzU1Nzk3OQ==&mid=2247484063&idx=1&sn=f0140e180f0b7dc96e31cc427b6285f8&chksm=97f6bc2ba081353dbd924997284f0fde015f5a154c5ef27996d22a048a7871dbf4acb9241ea7&scene=21#wechat_redirect)

[想编程，是勤奋自学还是去培训班学习？](http://mp.weixin.qq.com/s?__biz=MzIxNzU1Nzk3OQ==&mid=2247484311&idx=1&sn=74cb556a455fabe8b6bdbb74c928329d&chksm=97f6bd23a0813435e62f0e0c80dfa7f72216744a103eec9bbd678a38cfca0a1cd969397ddd98&scene=21#wechat_redirect)

更多面试吐槽，面试题，请看这个专题

[《Android面试专辑》](http://www.jianshu.com/c/1009682ba4cb)