虎牙一面面经

1，Volley与OkHttpClient的区别

2，HTTP各版本之间的区别

3，Tcp与Udp的区别

4，Java特性（封装、继承、多态）

5，内存泄漏

6，类加载机制

7，事件分发机制

8，MVC，MVP，MVVM的区别

虎牙二面

1，插件化，组件化

2，设计模式（观察者模式，单例模式）

3，单例模式中两个判空的作用，如果去掉里面的判空会出现什么

4，Https与Http的区别

5，tcp的三次握手

6，对一段序列进行排序的会选择什么算法（选择，插入，冒泡，堆排，快排，归并等进行了分析）

7，RN以及大前端的潜趋势

阿里一面

1，ListView的Item复用机制，MVVM的在ListView上的体现（数据驱动）

2，图片加载相关（估计是想问Glide，picasso框架等），LruCache等。

3，线程，Service的生命周期，Binder的原理，AIDL与Messenger的区别等，跨进程通信方式等

4，WebView，移动端的技术趋势

5，HashMap的原理，红黑树的构造，java7和8的区别，为什么java8中链表长度大于等于8才转换成红黑树？以及红黑树与链表的区别。

6，反射

手撸：1，实现数组的反转，以及打印

          2，实现三个线程A，B，C分别打印1，2，3，.....，100的数，要求A打印模3为0的数字，B打印模3为1的数字，C打印模3为2的数字，且按照递增顺序打印。思路：根据线程名或线程id进行判别，然后加锁。

360一面

1，public，private，和不加关键字修饰符的区别

2，匿名内部类的使用上

3，安卓本地存储的方式

4，View机制（）

5，事件分发机制

算法题：找出一个数组中出现次数最多的数字

360二面

1，Binder的机制

2，View的机制，如何实现一个时钟（秒针，分针，时针的绘制，我的思路是，上层分针转得快用动画实现，底层时针用绘制的方式实现）

3，hashMap的底层实现原理，例如，扩容后的index怎么变之类的

4，RSA，对称加密的，大数相乘算法思路

5，技术趋势，底层，双端

斗鱼三面（大多聊的项目）

1，事件分发机制

2，Lru算法

3，OkHttp的复用连接池大小是否能改变

4，多任务断点下载

字节跳动一面

1，自动拆箱装箱

2，HashMap的原理，链表数据是如何判断相等的

3，栈，队列的区别

4，hashtable

5，Synchronize

6，Activity生命周期 A 打开 B，然后按下back键，生命周期变化

    A->B，A:onPasue - B: onCreate - B:onStart - B:onResume - A:onStop

    按下Back，B:onPause - A:onRestart - A:onResume - B:onStop - B: onDestroy

7,布局优化，OverDraw的优化

8，ListView的复用错乱解决方式（图片显示错误）

算法，1，四则运算表达式的计算

            2，找出无序数组中两个数相加为K的元素的index

字节跳动二面

1，手写单例DCL，静态内部类的方式有点写错了

静态内部类的方式

public class Single{

    public static Single getInstance(){

        return Holder.instance;

    }

    public static class Holder{

        private static Single instance = new Single();

    }

}

2，TCP的三次握手

3，用C写socket服务端的实现

4，输入网址的过程（DNS，Http，Tcp，OSPF）

5，算法，判断是否是一个有效的IPV4的ip地址（有效：以.号分割,判断每段如果有长度，首位不能为0且长度不大于1，四位长度的小于255,即为真）

搜狐一面

1，Activity的生命周期，场景 A -> B ,然后返回的生命周期

2,四种启动模式

singleTask和SingleInsTance的区别

    1，SingleInstance会自动分配Affinify属性，创建新栈,注意

场景：A - Standerd

            B - SingleInstance

            C - Standerd

       A -> B -> C,再按下Back键，显示的是谁，其实主要就是看C加到拿个任务栈了

3,关于Activty的状态保存，onSaveInstanceState的调用

三种场景：1，A->B

                2, A -> Home键

                3，A -> 锁屏

4,ListView的底层原理

    图片错位如何解决

    RecyclerView的底层原理

ListView与RecyclerView的区别：

    1，缓存的机制不同，ListView二级缓存，RecyclerView四级缓存

    2,RecyclerView可以局部刷新

    3，ViewHolder是RecyclerView的必须，而ListView为推荐

    4,RecyclerView的LayoutManager提供了更多的布局

    5，RecyclerView支持item的动画添加和删除

    6，ListView默认间隔符，RecyclerView默认非间隔符

    7,RecyclerView未实现列表的点击，LitView默认实现列表的点击

5,图片压缩

6，Service的启动方式，bind与start两种方式的混用会如何？Service与Activity的通信

        混用：先start，则先stop再unbind；先bind，则先unbind再stop

7，IntentService是否会自己停

8，Service与Thread的区别

    区别：Thread，CPU调度的最小单位，

                Service，Android的四大组件之一

                Service的优先级相对较高

9，广播的分类，静态广播与动态广播的优先级（普通，有序，本地）

10，contentProvider

11，Android的几种本地化存储方式

12,SharedPreference的commit和apply的区别

    区别：SharedPreference 相关修改使用 apply 方法进行提交会先写入内存，然后异步写入磁盘，commit

            方法是直接写入磁盘。如果频繁操作的话 apply 的性能会优于 commit，apply会将最后修改内容写入磁盘。

            但是如果希望立刻获取存储操作的结果，并据此做相应的其他操作，应当使用 commit。

13,动画的分类,(帧动画，View动画，属性动画)

14,自定义View的过程

15,Handler的机制，handler采用定时是否准确，不准确（消息队列的阻塞）

16,HashMap的原理

17,sleep和wait的区别

    区别：1,sleep方法是Thread类的静态方法,wait()是Object超类的成员方法

              2,sleep()方法导致了程序暂停执行指定的时间，让出cpu该其他线程，但是他的监控状态依然保持者，当指定的时间到了又会自动恢复运行状态。在调用sleep()方法的过程中，线程不会释放对象锁。

而当调用wait()方法的时候，线程会放弃对象锁，进入等待此对象的等待锁定池，只有针对此对象调用notify()方法后本线程才进入对象锁定池准备

              3,sleep方法需要抛异常,wait方法不需要

              4,sleep方法可以在任何地方使用,wait方法只能在同步方法和同步代码块中使用