内存溢出与内存泄露

一、定义

1. 内存溢出: 即为 out of memory, 当你要求分配的内存超过了系统给你的内存时, 系统就会抛出 out of memory 的异常(每个 Android 能用的内存是有限的)
   比如: 当前应用只剩下 4M 的空间可用, 但你却加载得到一个需要占用 5M 空间的图片 Bitmap 对象, 就会抛出溢出的异常
2. 内存泄露: 即为 memory leak, 一个对象被创建后, 你不再使用它了, 但因为某种原因它又没有成为垃圾对象, 这块内存不能再被分配置使用.
   比如: 查询数据库得到的 cursor 对象在使用完后没有关闭, Activity 中使用 Handler 发延迟消息, 但退出前不移除未处理的消息
3. 内存泄露不多时没有太大影响, 但积累得多了就会导致应用运动缓慢, 到最后就会内存溢出.

二、内存泄漏的分类

1. 常发性内存泄漏: 发生内存泄漏的代码会被多次执行到，每次被执行的时候都会导致一块内存泄漏
2. 偶发性内存泄漏: 发生内存泄漏的代码只有在某些特定环境或操作过程下才会发生。常发性和偶发性是相对的。对于特定的环境，偶发性的也许就变成了常发性的
3. 一次性内存泄漏: 发生内存泄漏的代码只会被执行一次，或者由于算法上的缺陷，导致总会有一块仅且一块内存发生泄漏

   说明: 危害性大小顺序为: 1)>2)>3)

三、造成内存泄露的几种场景

1. 长生命周期的对象持有短生命周期对象的引用: Activity 中使用 Handler
2. 资源数据连接相关对象不关闭: cusor, stream, connection
3. HashSet 中的对象或 HashMap 中的 Key 对象, 基内部与 hash 值相关的属性被修改
4. 一些对象产生后不会自动释放或需要完全执行完了才释放. 比如: Bitmap, Thread, AsyncTask

四、避免内存泄露

1. 尽早释放无用对象的引用

2. 使用字符串处理，避免使用 String，应大量使用 StringBuffer，每一个 String 对象都得独立占用内存一块区域
3. 尽量少用静态变量，因为静态变量存放在永久代（方法区），永久代基本不参与垃圾回收
4. 避免在循环中创建对象

五、造成内存溢出的的场景

1. 申请了太多的对象. 比如: 使用 ListView 时，不复用 convertView，当数据项多时就会出现内存溢出
2. 创建的对象内存太大. 比如: 不经过压缩直接加载大图片文件
3. 内存泄露积累一定的时间后就可能出现

六、避免内存溢出

1. 通过复用对象的方式，减少产生的对象
2. 大对象需要先压缩后创建
3. 避免或减少内存泄露的情况