

1 年开发+android

笔试没过。

数据结构生疏，链表反转实现错误、堆排不会，编码能力一般。

Java 2.0

引用机制不懂;

容器类相关：不懂结构原理，arraylist 较长使用，hashmap/treeset 没用过；

static final 与 final static 差别，不了解；

string/stringbuilder/stringbuffer，了解 string 不可变表象，不了解实质，builder/buffer

区分错误；

synchronized、lock 能够简单区分差别;

atomic 语义不了解；

thread、runnable 简单用过，不了解区别；

匿名内部类使用规范：使用过，简单了解；

异常 try-catch 机制简单了解（db 连接，关闭资源）；

泛型：偶尔用，基本不了解；

线程池，没了解，印象中用过；

Android 2.0-

listview 多种 cell 不了解；

launchmode 启动模式，知道 4 种，不了解具体；

view 创建过程不了解；

inflate 仅用过一种形式，parent = null，其余不知;

view 绘制过程了解(仅知道 onmeasure/onlayout/ondraw，具体不知);

view 的 touch 事件传递机制不了解;

ansynctask 原理没概念;

viewstub 不了解 ;

1.5 年 android

笔试 3.5

堆排序、链表反转实现正确

Java 基础 2.0+

引用机制 (可以区分 soft\weak , 虚引用不了解);

容器类(arraylist 数组实现 ;linkedList 双链表 ;treeset 没用过 ,hashset 实现 hashMap ,
value=null ; hashMap , 初始大小 , 扩容方式*2+1,entry 结构链表+数组 ; hashtable 线程安全 , 继承旧版 dict , 1.7 源代码二次 hashCode)比较了解 ;

synchronized、lock 不了解区别;

Atomic 语义不了解 (原子概念不了解);

thread、Runnable 关系与区别不了解 ;

线程池 newCachedThreadPool (封装 ExecutorsService , 封装 task 执行回调给 unthread) ;

泛型不懂;

static final 与 final static 区别 (实际没差别) ;

如何扩展功能（继承、switch、接口（单例管理））；

异常 try-catch 仅简单试用，具体不了解；

exception, throwable, error 不了解；

匿名内部类不了解；

性能优化、编程习惯（自我总结和意识不足）；

Android 基础 3.0

launchmode 可以描述清楚 task 栈的变化；

fragment、activity（引用方式、内存泄露风险、生命周期、互调方式）了解；

addview 操作过程（new view 过程，layoutparamter 与 parent 相同，
layoutinflate.inflate 接口不了解）

gravity/layout_gravity 能够区分；

view 创建过程不了解；

viewstub（view 占位、使用过程中 inflate 初始化，具体为什么要用不清楚）；

view 绘制过程了解(知道 onmeasure(扩充模式不记得)/onlayout/ondraw ,viewgroup
绘制不了解，dispatchdraw 不了解);

view 的 touch 事件传递机制（dispatch 不记得需要 return，但是能够大致描述清楚）；

layer-list 不知道；

shape 效果了解（圆形、线条、渐变、描边、padding）；

view 状态（focus、enable、press、select）；

文本 span 可实现效果（变色、加粗、字体、链接）；

硬件加速（manifest 中定义，定义范围不知），具体硬件加速原理不清楚；

android 版本兼容 (lint 扫 api、2.3tabhost、viewpager ,判断版本号 ,support-v4/7);

android 性能优化 (view 方面 , drawable , 9-patch 代替大图、不同分辨率区分 , 平铺当背景图片 no dpi , 扁平化、顶层 layout 采用 merge , hierarchyviewer 查看 , include 代码复用 , 图片加缓存 , list view 重用、异步加载) ;

list view (知道类型扩展、view holder、tag、根据 tag 到 view 中取 holder、重用需要重新刷新内部 view 组件) ;

ansynctask 使用 , 原理了解 (3.0 前后实现不同 , background、线程池、结束之后需要重新 new , 不能重新启动) ;

1 年开发+android

笔试没过。

数据结构生疏 , 链表反转实现错误、堆排不会 , 编码能力一般。

Java 2.0

引用机制不懂;

容器类相关 : 不懂结构原理 , arraylist 较长使用 , hashmap/treeset 没用过 ;

static final 与 final static 差别 , 不了解 ;

string/stringbuilder/stringbuffer , 了解 string 不可变表象 , 不了解实质 , builder/buffer

区分错误 ;

synchronized、lock 能够简单区分差别;

atomic 语义不了解；

thread、runnable 简单用过，不了解区别；

匿名内部类使用规范：使用过，简单了解；

异常 try-catch 机制简单了解（db 连接，关闭资源）；

泛型：偶尔用，基本不了解；

线程池，没了解，印象中用过；

Android 2.0-

listview 多种 cell 不了解；

launchmode 启动模式，知道 4 种，不了解具体；

view 创建过程不了解；

inflate 仅用过一种形式，parent = null，其余不知；

view 绘制过程了解(仅知道 onmeasure/onlayout/ondraw，具体不知)；

view 的 touch 事件传递机制不了解；

ansynctask 原理没概念；

viewstub 不了解；

1 年 android+应届

小孩比较聪明，动手能力可以。

对所接触的知识了解及掌握程度不深，钻研程度不够。

操作系统，自己说不了解，没考察。

研究生网络方向，考察了一下网络基础，基本概念已经扔的差不多了。

tcp 两个小问题没有思路。

#1000 个字节从 a 点传输给 b 点，需要多少次（mtu、拥塞控制）；

#a 点、b 点通过网线链接，握手成功建立连接之后，突然物理断开，接下来会发生什么（keep-alive）；

tcp/udp 了解不够深（仅知道长（无）连接、三次握手）；

Java 基础 2.5

引用机制（弱、软无法区分）不了解；

各种常见容器类了解使用，都不了解实现原理（仅知道链表、数组、哈希实现，再具体不知）；

synchronized、lock 能够简单区分差别；

Atomic 不了解；

thread、runnable 关系与区别简单了解；

线程池 newcachedthreadpool 不了解；

泛型不懂;

java 继承方式了解 (接口方式代替多重继承);

static final 与 final static 不清楚差别 (实际没差别);

异常 try-catch 机制简单了解 (场景: 文件读取, final 关闭 stream 操作);

能够确定递归全部能够非递归方式替代;

如何扩展功能 (多用接口方式);

性能优化编程习惯 (慎用 static 变量, 避免使用大对象, hashmap 替代 list) 有一些;

匿名内部类使用规范讲述不清;

Android 基础 2.5

android 版本演进 (3.0fragment) 了解很少;

android 版本兼容没经验;

surfaceview 特别之处不清楚 (有游戏开发经验);

硬件加速不了解;

view 创建过程不了解;

view 绘制过程了解(仅了解 onmeasure/onlayout/ondraw , viewgroup 相关不了解);

view 的 touch 事件传递机制不够了解, 无法准确描述;

ansyncntask 简单使用过, 原理不了解;

android 性能优化没概念 (布局优化、listview);

文本 span 相关不了解;

viewstub 不了解;

笔试没过，后两题没思路、写不出来。

4 年开发&android

Java 基础 2.5

引用机制不了解;

string/stringbuilder/stringbuffer 了解 (线程安全、创建方式、native 实现、final) ;

容器类比较了解 ;

hashset/treeset 不了解 ;

hashmap 实现原理简单了解(hashCode,entry 数组以及链表、扩容、与 hashtable 差别) ;

synchronized、lock 差别不了解;

volatile 不了解;

Atomic 不了解;

thread、runnable 关系与区别简单了解 ;

线程池 newcachedthreadpool 不了解 ;

泛型不了解;

java 继承方式了解 (接口方式代替多重继承) ;

异常 try-catch 机制简单了解 (final 关闭 db/stream 操作) ;

finalize 关键字简单了解 ;

Android 基础 2.5

launchmode 比较了解；

application、task、process 清晰了解；

view 创建过程不了解；

layoutinflater 了解如何获取实例，无法区分 inflate 两种参数形式接口差别；

view 绘制过程了解；

view 的 touch 事件传递机制简单了解（dispatch\intercepte...），无法准确描述；

asynctask 使用过，内部实现简单了解到线程池和 ui 线程；

硬件加速不了解；

layer-list 没了解；

listview 性能优化（扁平化布局、异步加载、viewholder 复用 view）；

文本 span 相关不了解；

笔试没过，编码能力不够，写不出来。

5 年开发，3 年 android

Java 基础 3

hashset 实现原理简单了解（value 指向同一对象，元素不能重复），

treeset 实现原理简单了解（可排序、接口 compare 自定义）

hashmap 实现原理了解 (hashcode,entry 数组以及链表、扩容,与 hashtable 差别)

lock 不了解 , synchronized 关键字了解使用;

volatile 不了解;

Atomic 简单了解;

引用机制简单了解;

string/stringbuilder/stringbuffer 了解 (线程安全、创建方式) ;

泛型不了解;

异常 try-catch 机制 (注意事项、系统开销较大不易大范围捕获、多个 exception 区分处理、final 关闭 db/stream 操作、return 执行顺序) ;

final、finally、finalize 关键字各自使用场景能够大致描述 ;

thread、runnable 关系与区别 ;

线程池 newcachedthreadpool 用过但执行逻辑不清楚 ;

Android 基础 2.5

launchmode 比较了解 ;

application、task、process 之间的关系比较模糊 ;

view 创建过程不了解 ;

layoutinflater.inflate 能够描述三个参数版本的接口 , 无法区分差别 ;

view 绘制过程不了解;

view 的 touch 事件传递机制简单了解 (dispatch\intercepte...) , 无法能够正确描述;

ansynctask 使用过 , 内部实现简单了解到线程池和 ui 线程;

硬件加速仅了解针对 window/view/activiy/application 进行开关;

layer-list 没了解；

《这是 qiuliang 的一个面试反馈》java 3-

android 3+

技术沟通细节

java:

hashmap: 知道 equals 为 true 时, hashCode 须一致; 知道大概实现原理

ConcurrentModificationException: 多线程修改 / 访问导致; 认为单线程不会;

WeakHashMap: 看名字有点概念, 但认为是 value 用 WeakReference 管理;

ReferenceQueue 没用过;

generics: wildcard 没怎么用过; 见过 ? extends

finalize(): 知道基本原理; 知道不建议用的原因;

android:

应用和进程: 知道一个 app 可以有多个进程; 不太确定 Application 是否有多个;

task stack: 知道会恢复; 知道 ActivityManager 可以看 RunningTask, RecentTask;

ui 性能优化:

减少层次 (比如使用 RelativeLayout, 使用 ViewStub) ;

避免耗时操作;

listview: 重用 convertView, 使用 ViewHolder 避免 findViewById;

Bitmap 要 sample size;

用过 traceview, lint, hierarchyviewer, 知道 overdraw

sysdump

知道有这个东西, 有一点印象

第一轮面试反馈—候选人杜家杰

中级 2 分 (5 分制)

笔试 1 分

思路方面较快、正确、能够进一步优化。

第一题, 思路, 取差比较, C 作答, 代码不正确;

第二题, 思路: 排序。代码未写;

第三题, 思路: 链表倒置。代码未写。

Java 基础 2 分

synchronized 关键字用法了解;

volatile/AtomicXXX 关键字不了解;

死锁问题与 block 操作混淆;

string/stringbuilder/stringbuffer 无法区分;

happens-before 原则没听说过;

泛型操作概念了解;

reference 机制不了解;

集合框架简单了解;

异常机制不太了解;

throw/try-catch-finally 不了解 (return 执行流程不确定)、throws 了解;

Android 基础 2 分

view 的 touch 事件传递机制不了解;

ansynctask 简单了解使用, 底层实现不了解;

activity 的 lanchmode 简单了解, 结合实际分析不正确;

listview 使用及性能优化不了解;

硬件加速不了解;

handler/message/looper, 知道简单使用及对应关系;

aidl 进程通信简单了解原理, 具体使用细节掌握不够;

布局掌握不够, linearlayout 三等分不会;

其它基础 (数据结构、网络、C/C++/Python、linux 等) 2 分

C/C++ 基础: 较差, 无法书写;

数据结构: 较差, 基本排序算法不会;

网络：

socket 原理描述不清楚;

滑动窗口不了解;

糊涂窗口问题不了解;

拥塞控制不了解;

https 协议实现原理简单了解，具体握手过程不了解;

工程经验 2 分

代码/性能优化，没做过；

综合评价：

编码能力不足，专业基础较差;

android 项目较少，实际工程经验不够；

Bitmap

知道内存大小占用；

对存放位置有概念；

第一轮面试反馈—候选人刘丁瑜

初级 3 分（5 分制）

笔试 4 分

思路清晰、正确，能够通过引导完善作答。

第一题，求和取差。C 正确作答。错误之处：输入不对；

第二题，小根堆。C 作答，基本正确。错误之处：混用 C++ 的 new 关键字、堆插入、删除方法中缺少 i 参数声明、堆删除最后节点交换执行顺序不对；

第三题，链表倒置。C 作答，不够完整。错误之处：缺少 struct 定义、节点类型不对、输入输出链表未实际调用 reverse 操作、输出不对，不应求和；

Java 基础 2.5 分

synchronized 关键字用法较了解；

volatile/AtomicXXX 关键字不了解；

string/stringbuilder/stringbuffer 能够从内存角度理解、从线程安全、性能方面无法区分；

泛型操作比较了解；

4 种 reference 了解；

集合框架简单了解；

异常机制及相应处理不了解；

throw/try-catch-finally 比较了解（return 执行流程不清晰）；

Android 基础 2 分

view 的 touch 事件传递机制不了解；

ansynctask 比较了解;

activity 的 lanchmode 不了解, 结合实际分析不正确;

webview 知道简单使用;

硬件加速不了解 (知道有 manifest/java 配置方式、具体不了解);

handler/message/looper 懂得使用, 不甚了解;

其它基础 (数据结构、网络、C/C++/Python、linux 操作系统等) 2 分

C/C++ 基础 : 较好;

数据结构 : 链表、树、大小根堆排序了解;

操作系统/linux :

线程/进程(基本概念简单了解、分段/页地址映射管理、进程/线程上下文、任务栈等简单理解);

linux 系统调用 fork、进程启动流程不了解;

线程/进程通信与同步 (同进程、跨进程、pipeline (单双向区别、本质)、信号量、共享内存、socket) 不了解;

网络 :

socket 原理及使用不了解;

心跳机制不了解;

糊涂窗口问题不了解;

拥塞控制简单了解;

TCP/UDP 区别, 不了解;

https 协议实现原理不了解;

http 常用 header 不了解;

能够设计实现多线程下载;

工程经验 2 分

代码/性能优化 (布局层级尽量要少、switch 优于 if-else (原理不了解) 等) ;

综合评价 :

优点 :

解决问题思路比较敏捷、比较有 sense;

根据笔试表现, 编码能力不错;

不足之处 :

专业基础知识面广、但不够夯实;

工作时间短、实际工程经验及 android 经验不够;

《qiuliang 的面试反馈》java 和 android 都是开始工作这一年内学习的, 掌握还可以

java:

可以解释清楚 HashMap 的基本实现

对 WeakReference, SoftReference 有基本了解和使用, ReferenceQueue 有一点概念, 不

了解具体使用

对 c++ 模版和 java 范型了解基本的实现区别

android:

对 back stack 及 task 有基本理解, 知道进程被 kill 后 stack 会被恢复

ContentProvider, 了解使用, 知道用一个实例来处理并发访问

service 没怎么用过

之前工作中用过 webview, 基本的控制、回调等都比较熟悉, 知道 jsbridge 的漏洞

ui 优化 知道避免 io 操作 控制 view 层级 ,listview 要复用 convertView 及使用 ViewHolder,

用过 hierarchyviewer 和 traceview

没做过内存方面的分析

android 的源码没有专门看过, 偶尔在 eclipse 中看下

业余会用 shell, python 写点小东西

《技术 boss 梁汝波面试反馈》由于一, 二面问得比较多, 我只简单聊一下:

对编码和设计有简单体会

对 UI, 动画感兴趣

换工作的原因是在 360 频繁响应需求, 没有节奏, 希望有 lead 能统一协调工作

希望深入熟悉 android 机制和实现，理解 view 绘制原理，实现流畅的 ui，不过业余精力投入到了了解 shell, python, 正则等技术上了

沟通流畅

再发一个我们组刘成的面试反馈。android 有一定的知识广度，但是深度相对欠缺。3-

看过的书：设计模式/Java 看过一些，都比较久远，不记得书名/作者，只记得是机械工业出版社出版，(貌似是 CoreJava, GoF)

使用的 IDE: eclipse 为主, android studio 用过早期版本，感觉差不多，后者补全功能略强

ant/maven/gradle: 项目中使用 ant, 其他有听过，但不清楚各自的特点.

hierarchyview: 很久以前用过，记得有些设备可用，有些不可用，不清楚为什么

pulltorefresh 原理: 看过某个库的实现，但是不记得具体名称/作者. 实现了解也不够深入

硬件加速/layerType: 不了解

PropertyAnimation: 了解基本使用，对于自定义属性动画，以及实现原理不清楚

了解蓝牙/NFC 在 Android 新版本中的增强特性

另外，很重要一点，把相关工具都熟悉一下，lint、hierarchyviewer、mat、findbugs、

traceview 等等，用一用，要知道都是干什么的，怎么用的

还有一点，proguard 也仔细看一下

qiuliang 的最新一个面试反馈。java 略差 2.5，android 还可以 3

java 没怎么系统地深入学过

范型：wildcards

finalize()理解：知道可能被 gc 调用

反射：unittest 时用过反射，知道可以用于访问 private 方法

听过 WeakHashMap

android

应用与进程关系：知道可以多进程；Application 对象是否都有单独实例：不是很确定

service: startService/bindService; 交互: binder;

多进程交互: broadcast, binder (aidl), socket, ContentProvider

ContentProvider: 单个实例

api 版本兼容: 直接判断 sdk_int, 知道 support 包中 ViewCompat

ddms 中工具: debug, 截图, viewanimator, mocklocation, heap, MAT, update Thread

偶尔看下 android 源码

学习:

dev 文档

csdn, 51cto, google, stackoverflow

开源项目:

PullToRefresh, ActionSherlock, ViewPager, SlidingMenu, HorizontalListView, ZXing

用过 HttpClient

之前的项目: android 5-6 人, (ios 也是 5-6, server 20~30), 测试 4-5 个

节奏: 一个星期一个版本

对工作节奏有心里准备

之前团队最长经验的 2.5 年, 觉得自己无成长, 比较关心团队是否能帮助自己成长

嗯 版本演进、兼容 (why) 也看一下 各版本的差异

support-v4/7 包, 还有里面的 compat

把 android 项目 build 过程直至打包完成 也了解一下

笔试：

每个题单独打分后平均

1. 二分查找实现

2. 从长度为 100 万的整数数组中找出最大的 100 个 (100 万的数组在内存中，数组中数可能重复)

3. 用单向链表表示十进制整数，求两个整数的和 (如给出链表 A、B，返回 C)

打分规则:

每个题按 5 分打 (权重 2 : 4 : 4)

第 1 题实现 5 分

第 2 题思路 1.5 分，实现 3.5 分

第 3 题思路 1.5 分，实现 3.5 分

面试知识点：

java 相关：

Q: ArrayList 和 Vector 的主要区别

A: Vector 是同步的，线程安全

Q: Object 类 hashCode(), equals()方法的理解 (a == b 与 a.equals(b)的区别), 在 HashMap 等类库中如何利用这两个方法的

A: == 判断两个变量是否指向同一个对象，equals 判断两个对象“内容”是否相同；equals()为 true 的两个对象的 hashCode()必须相同，equals 为 false 的两个对象的 hashCode()可以相同；

HashMap 用 key 的 hashCode()作为散列值，求余后找到链表，再在链表中对 key 作 equals()比较.

Q: 反射机制理解

Q: finalize()：做什么，为什么不建议用

Q: WeakReference, SoftReference, ReferenceQueue 理解, WeakHashMap 的原理

Android:

Q: Task stack , Activity, 进程，及应用的理解

A: task stack 记录启动的 activity 历史，启动新的 activity 时 push, 返回的时候 pop;

Activity 的 launchMode 理解:

standard: 每次启动都创建新实例

singleTop: 如果 stack 顶部有要启动啊@的 activity 的实例, 则直接用此实例
(onNewIntent())

singleTask: 此 activity 只能放在 stack 的最底部

singleInstance: 此 activity 只能是 stack 中唯一一个 activity 实例

常用的 Intent flags:

FLAG_ACTIVITY_SINGLE_TOP: 效果等同于 singleTop

FLAG_ACTIVITY_NEW_TASK: 将 activity 放到“新的” stack 中(默认放当前 stack 中); 用非 activity 的 context 来启动 activity 需要加此 flag

FLAG_ACTIVITY_CLEAR_TOP: 如果要启动的 activity 已经有实例在当前 stack 中, 不启动新的实例, 而是将其上面的 activity 都关闭

task stack 由系统管理, 进程被系统杀掉或意外 crash 后, stack 仍然保留, 再启动改应用, 会恢复 stack 顶部的 activity;

一般一个应用启动一个进程; 但 一个应用的 activity, service 等可以运行在不同的进程中; 不同的应用如果证书签名一致也可以设置运行在同一进程中;

Q: service 运行在哪个线程? 如何启动和停止

A: main thread

startService, bindService

stopService (stopSelf), unbindService,

Q: ContentProvider 理解

Q: BroadcastReceiver 理解

Q: touch event 分发及处理流程

A: dispatchTouchEvent, onIntercept, onTouchEvent , 父 view 及子 view 处理流程

Q: view layout 过程

A : measure -> layout, 自定义 view 需要覆盖 onMeasure, onLayout

Q: UI 优化

A: view tree 层级不要过深(<11);

AdapterView 不会对每个 item 用生成一个 view, 而是只对可见的 item 生成 view, view

会被重用,Adapter 的 getView 要注意重用 convertView; 同时用 viewHolder 避免重复

findViewById

分析工具 : hierarchy viewer, lint (layoutopt)看 UI

traceview 看函数调用时间

Q: 内存分析相关

A: ddms 中 heap, allocation tracker

MAT 分析引用情况

Q: 图片加载

A: 队列控制, 缓存控制(内存, 磁盘), bitmap 占用内存大小评估

=====

2014-11-23 讨论整理

java:

a 容器类使用: 类型, 差别

a vector, list

a - b map 原理

a string / stringbuilder

a 引用类型, 区分

b WeakHashMap, ReferenceQueue

a exception, throwable, error 继承关系, check/uncheck, runtime

a finally

并发：

a synchronized,

a AtomicXXX

b ExecutorService

b ConcurrentHashMap

a 基本类型，引用类型 大小

a 数组大小

a 内部类/匿名类：知道对外部的 this 引用

范型：

a 基本用法: 类，方法

b wildcards

b 是否知道是编译时处理

a 接口 / 抽象类区别

a - b 反射

b Annotation

b GC 原理

b finalize()

a for loop 与迭代器

a 如何从 list 中删除元素

OOP/设计模式

a-b 列出几种常见设计模式(单例，工厂，Builder)

a-b 用 java 写出一个单例的实现

b xml/json 解析方式

b csv 文件读取接口设计

Android

a activity 生命周期 (必须, 不清楚迅速 pass)

a-b 应用和进程关系

a-b task 和 activity launchMode, intent affinity(flags), task 被恢复情况(加分)

b activity.onSaveInstanceState()

a support 包 , fragment 使用

ANR

a 什么是 ANR

b 如何看 ANR log

service:

a 运行在那个线程 ?

a 如何启动 / 停止 ?

a-b activity 如何与 service 交互 (intent, ServiceConnection, aidl, Messenger)

b STICKY, NOT_STICKY (深入)

contentprovider:

a 有几个实例?

a 运行在那个线程 ? 注意线程安全 ?

Q: BroadcastReceiver 理解

a 运行在那个线程 ?

Q: api 兼容如何写

a-b if ,反射 , XXXCompat

Q: touch event 分发及处理流程

a-b (自顶向下, 涉及方法: dispatchTouchEvent, onInterceptTouchEvent, onTouchEvent)

深入 : 描述具体 case

Q: dpi 概念

a

Q: Handler, Message, Handler leak, obtainmessage

a

Q: Drawable

a 有哪些?

Q: style, Theme

a

Q : ImageView: scaleType

a

Q: 自定义 ViewGroup

a onMeasure, onLayout

Q: LayoutInflate.inflate attach=false 与 parent=null 差别

a

Q: UI 优化

b

A: view tree 层级不要过深(<11);

AdapterView 不会对每个 item 用生成一个 view, 而是只对可见的 item 生成 view, view

会被重用,Adapter 的 getView 要注意重用 convertView; 同时用 viewHolder 避免重复

findViewById

分析工具 : hierarchy viewer, lint (layoutopt)看 UI

traceview 看函数调用时间

Q: 硬件加速

b

Q: 内存分析相关

b

A: ddms 中 heap, allocation tracker

MAT 分析引用情况

Q: Bitmap 及图片加载

a-b Bitmap decode: inJustDecodeBounds, inSampleSize, InPurgable,

inPreferredConfig

b 占用大小

b 存储位置: native heap, java heap

b 异步加载设计, 磁盘缓存设计