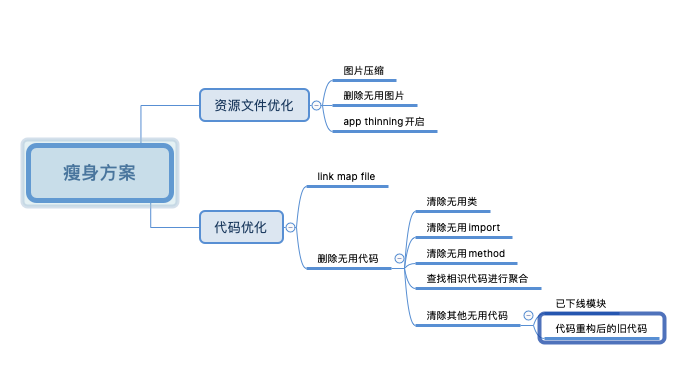
iOS瘦身方案

一、问题分析。

随着版本迭代的增多，经过多轮开发经手后ipa包越来越大。

二、方案：

  
三、具体过程：

3.1、资源文件优化：资源文件优化你可以从图片资源压缩、无用资源检查、资源文件上传中入手，或者使用官方提供的App Thinning。  
 3.1.1图片压缩：图片资源压缩也有很多公司使用webp格式图片，这种格式对比png、jpg要小很多，但客户端性能对解码webp图片会有一定的影响。在一些老的机型上，表现不是很好。所以需要做一些取舍。这里有webp和png格式转换的工具 **isparta**其实我觉得这两种方案可以共存，团队内部设定一个资源文件的最大值，比如超过 100KB 就选择使用 webp格式的图片，小于 100KB 就使用压缩工具。这样就可以极大的减少资源文件的大小，同时减少包的体积。  
 3.1.2 删除无用资源。

通过LSUnusedResourc（<https://github.com/tinymind/LSUnusedResources>）删除无用图片。需要注意的是在使用的过程中，你需要检查有些资源文件后缀是否有拼接上的，如“home\_btn\_normal”“home\_btn\_select” 或者 “bg\_1”，“bg\_2”这些需要人工进行查看是否有用到  
 3.1.3 部分较大且可动态修改的图片可考虑由服务器通过接口获取然后缓存。  
 3.1.4 app thinning 开启由苹果进行优化。他会针对不同架构进行图片资源等的下载。  
 3.2 代码优化  
 3.2.1 link map file：我们编写的源码需要经过编译、链接，最终生成一个可执行文件。在编译阶段，每个类会生成对应的.o文件（目标文件）。在链接阶段，会把.o文件和动态库链接在一起。Link Map File就是这样一个记录链接相关信息的纯文本文件，里面记录了可执行文件的路径、CPU架构、目标文件、符号等信息。理解Link Map File，可以帮助我们：1、理解链接过程。2、理解Crash时，通过Symbols还原出源码的机制。3、理解内存分段、分区。4、分析可执行文件中哪个类或库占用比较大，进行安装包瘦身。Link Map File配置：点击工程，选择Build Setting选项，搜索map，可以看到如下界面。将Write Link Map File设置为Yes后，Build结束后，会在默认路径下生成一个Link Map File文件，该文件是txt格式的。点击Path to Link Map File，可以设置Debug或Release模式下的生成路径。关于分析工具<https://github.com/huanxsd/LinkMap>  


可图形化查看工程中代码类的大小。

3.2.2 代码静态检查：可以使用AppCode（https://www.cnblogs.com/zhangxiaomeng1991/p/8350339.html）