面试题:NSOperationQueue与GCD的使用原则和场景

首先，我们要明确NSOperationQueue与GCD之间的关系

NSOpertaionQueue用GCD构建封装的，是GCD的高级抽象。

其次，我们要区别两者的不同

GCD仅仅支持FIFO队列，而NSOperationQueue中的队列可以被重新设置优先级，从而实现不同操作的执行顺序调整。

GCD不支持异步操作之间的依赖关系设置。如果某个操作的依赖另一个操作的数据（生产者-消费者模型是其中之一），使用NSOperationQueue能够按照正确的顺序执行操作。GCD则没有内建的依赖关系支持。

NSOperationQueue支持KVO，意味着我们可以观察任务的执行状态。

了解以上不同，我们可以从以下角度来定义原则

1. 性能

GCD更接近底层，而NSOperationQueue则更高级抽象，所以GCD在追求性能的底层操作来说，是速度最快的。这取决于使用Instruments进行代码性能分析，如有必要的话

2. 从异步操作之间的事务性，顺序行，依赖关系。GCD需要自己写更多的代码来实现，而NSOperationQueue已经内建了这些支持

3. 如果异步操作的过程需要更多的被交互和UI呈现出来，NSOperationQueue会是一个更好的选择。底层代码中，任务之间不太互相依赖，而需要更高的并发能力，GCD则更有优势

最后的一句话

别忘了高德纳的教诲：“在大概97%的时间里，我们应该忘记微小的性能提升。过早优化是万恶之源。”只有Instruments显示有真正的性能提升时才有必要用低级的GCD。