JVM 调优主要针对的是堆内存。JVM 的内存结构：方法区、堆内存、程序技术器、虚拟机栈（局部变量表、操作数栈、动态链接、返回地址）、本地方法栈，其中方法区和堆是线程共享的。Java 的对象内分配中堆内存，eden : survivor : survivor = 8:1:1，或者新生代 new:老年代 old=1:3 或者 3:8。当第一次 new 出来的对象时，特别大的对象会直接放在老年代，其他的普通对象放在新生代。再经过一次 GC 之后，eden 中还存活的对象，会到 survivor1 中，又经过一次 GC，eden 中加 survivor1 中的对象复制到 survivor2 中，再经过一次 GC，eden 中存活的加 survivor2 中的对象复制到 survivor1 中。内存的复制效率非常快。

New 出来的对象特别小的时候，直接放在栈上，前提是提前开了栈上分配的优化，默认是开启的。优点是栈上分配的不用进行垃圾回收，所以在栈上分配的效率比较高。栈上分配的无逃逸是指在一个方法内的对象，在方法外没有引用指向。TLAB，每个线程去 eden 申请内存，不用加锁，提高对象分配效率。一个对象 new 出来，首先放在栈上，如果栈上放不下，放线程本地，线程本地放不下，放 eden，如果特别大，直接放在老年代。

Minor GC 针对的是新生代的 GC。Major GC 针对的是老年代的 GC。Full GC 针对的是新生代和老年代的 GC。

HTTP1.1 和 HTTP1.0：（1）HTTP1.1 支持长连接，在一个 TCP 连接中可以传送 HTTP 请求和响应，减少了不断建立连接的消耗。（2）HTTP1.1 增加了 host 字段，一台物理机上存在多个虚拟机，可以共享一个 IP 地址。（3）增加了 100 状态码，客户端先发送一个只有请求头的请求，目的是试探服务器接收还是拒绝，如果接收了再发送请求体。

二叉树除了用递归，还可以使用非递归的栈来实现。

创建对象的方式：（1）new（2）Class 类的 newInstance()方法（3）clone 方法（4）反序列化

Linux 下常用命令：（1）ls，显示文件或目录（2）mkdir，创建目录（3）rmdir，删除空目录（4）rm，删除文件（5）cat，查看文件内容（6）cp，拷贝（7）touch，创建空文件（8）pwd，显示当前行数（8）wc，统计次数（9）tar，打包压缩。（10）chmod，权限管理（11）ipconfig，查看网络状况（12）ping，测试网络连通（13）netstat，显示网络状态信息（14）ps，查看进程状态（15）kill，杀死进程

批处理操作系统特点：成批处理，系统吞吐量高，资源利用率高，用户不能直接干预作业的执行。分时操作系统：多路性、独立性、及时性、交互性。实时操作系统：及时响应、快速处理、高可靠性和安全性、不要求系统资源利用率。

volatile 关键字保证在一个线程中对某一共享变量的更改，在其他线程中能够立即可见，并且禁止指令重排序。其保证可见性，但是不保证原子性，是一种轻量级的同步机制。Java 的 serialization 提供了一种持久化对象实例的机制，当持久化对象的时候，可能有一个特殊的对象成员，不想用 serialization 机制保存，为了在一个特定对象的一个域上关闭 serialization，可以在这个域上加上关键字 transient。

Synchronized 会自动释放锁，但是 Lock 必须在 finally 中手动释放锁。Lock 的特性，可重入、可中断、可限时、公平锁等。

一个文件中有 1000000 个数，找出最大的 50 个数。分治法、hash、最小堆。如果这 100 万个数有很多的重复数字，先用 hash 去重。然后将剩下的数字分组，比如分为 1000 组。建立最小堆，找出每一组中最大的 50 个，然后将找出来的所有数字放在一起，找出最大的 50 个数字。

内存泄漏是指程序动态分配了内存，但是在程序结束的时候没有释放这部分内存，从而造成内存不可用的情况。常用避免内存泄漏的方式：（1）少用 String，尽量用 StringBuffer 代替。（2）尽量少用静态变量，静态变量属于全局的，GC 不会回收。（3）避免集中创建大的对象。（4）使用缓存技术缓存经常使用的对象。（5）不要再循环中创建对象。

不管有没有出现异常，finally 块中代码都会执行，当 try 和 catch 中有 return 时，finally 仍然会执行。Finally 是在 return 后面的表达式运算后执行的，函数返回值是在 finally 执行前确定的。Finally 中最好不要包含 return，否则程序会提前退出，返回值不是 try 或 catch 中保存的返回值。

hashCode 的作用主要是用来配合基于散列的集合，如 HashMap、HashTable、HashSet，计算散列值，然后计算位置。假如在 hash 表中，如果将每一个对象都 equals，如果是比较一个对象，里面有很多字段，需要一一比较，消耗比较大，有了 hashCode 能减小比较的消耗，hashCode 默认是地址，重写的时候一般是地址和字段配合。hashCode 相等，equals 可能不相等，因为 hashCode 可能不是一个地址生成的。Equals 相等则 hashCode 一定相等。如果只用 hashCode 可以用来判断一个对象不相等，不能用来判断一个对象相等，可以提高查找的效率。

Finalize 方法跟 JVM 垃圾回收相关。两次标记：即使经过可达性分析之后该对象不可达，不代表该对象非死不可，是否死亡要经过两次标记。如果对象在进行可达性分析以后和 GC Roots 没有直接或者间接的连接，此时会进行第一次标记，筛选的条件是，当 finalize 方法没有被覆盖或者 finalize 方法被调用过的，则这次标记的对象被回收。第二次标记，如果一个对象覆盖了 finalize 方法，该对象将被放在一个 F-queue 队列中，之后虚拟机会自动触发该队列中的 finalize 方法，如果该对象没有重新与 GC Roots 建立连接（将 this 赋值给某个静态变量就是建立连接了），就会被垃圾回收器回收，收对象可以在 finalize 方法中拯救自己。

四种会话跟踪技术，保存用户和服务器之间的会话状态及信息技术叫会话跟踪技术。跟踪技术有四种：cookie、URL 重定向、隐藏表单、session。

Object 类中的方法：（1）clone 方法，保护方法，实现对象的浅复制，只有实现了 Cloneable 接口才可以调用 clone 方法，否则抛异常 CloneNotSupportedException。（2）getClass 方法，final 方法，获取运行时类型。（3）toString 方法。（4）finalize 方法，用于释放资源，很少使

用。（5）equals 方法，在 Object 中 equals 等于==，子类重写。（6）hashCode 方法，用于哈希查找，重写了 equals 方法一般都要重写 hashCode 方法。（7）wait 方法使得当前线程等待该对象的锁，当前线程必须是该对象的拥有者，即具有该对象的锁。wait()方法一直等待，直到获得锁或者被中断,wait(long timeout)设定一个超时间隔，如果在规定时间内没有获得锁就返回。调用该方法后当前线程进入睡眠状态，直到其他线程调用了该对象的 notify 方法或者 notifyAll 方法或者 interrupt 方法或者时间间隔到了。此时该线程可以被调度，如果使用的是 interrupt 的，则抛出一个 InterruptedException。（8）notify 方法，唤醒在该对象上等待的某个线程。（9）notifyAll，唤醒在该对象上等待的所有线程。

Static：（1）修饰变量，静态变量随着类加载时完成初始化，内存中只有一个，且 JVM 也只会为它分配一次内存，所有类共享静态变量。（2）修饰方法，在类加载的时候就存在，不依赖任何实例，static 方法必须实现，不能用 abstract 修饰。（3）修饰代码块，在类加载完之后就会执行代码块中的内容。（4）父类静态代码块---子类静态代码块---父类非静态代码块---父类构造方法---子类非静态代码块---子类构造方法。

Final：（1）修饰变量，编译器常量，类加载的过程完成初始化，只能是基本类型。运行时常量，基本数据类型或者引用数据类型。引用不可变，但引用的内容可变。（2）修饰方法，不能被重写，不能被子类修改。（3）修饰类，不能被继承。（4）修饰形参，形参不可变。

HashMap 中的 key 可以为 null，但不能是可变对象，如果是可变对象，对象中的属性改变，则对象中的 hashCode 也会改变，导致下次无法查找到 Map 中已经存在的数据。如果可变对象在 HashMap 中被用作键，在改变对象状态的时候，不要改变哈希值。HashMap 是非线程安全的，ConcurrentHashMap 是线程安全的，将整个 Hash 桶进行了分段，将大的数组分成了几个小的片段，每个小的片段上都有锁存在，在插入元素时需要先找到应该插入到哪一个段，然后再插入，需要获得锁。

排序算法时间复杂度

排序算法	平均	最差	稳定性	空间	备注
冒泡排序	O(n^2)	O(n^2)	稳定	O(1)	N 小时较好
交换排序	O(n^2)	O(n^2)	不稳定	O(1)	N 小时较好
选择排序	O(n^2)	O(n^2)	不稳定	O(1)	N 小时较好
插入排序	O(n^2)	O(n^2)	稳定	O(1)	大部分已排序较好
基数排序	O(logRB)	O(logRB)	稳定	O(n)	B 是真数（0-9）R 是基数
希尔排序	O(nlogn)	O(n^s)1<s<2	不稳定	O(1)	S 是所选分组
快速排序	O(nlogn)	O(n^2)	不稳定	O(nlogn)	N 大时较好
归并排序	O(nlogn)	O(nlogn)	稳定	O(n)	N 大时较好
堆排序	O(nlogn)	O(nlogn)	不稳定	O(1)	N 大时较好

Java 四种引用：（1）强引用，只要引用存在，垃圾回收器永远不会回收。（2）软引用，

用来描述一些还有用，但不是必须的对象。在系统将要发生内存溢出异常之前，将会把这些对象列入回收范围进行第二次回收。（3）弱引用，被弱引用关联的对象只能存活到下一次垃圾回收前。（4）虚引用，一个对象是否有虚引用的存在，对其生存时间不构成影响。为一个对象设置虚引用的目的是这个对象被垃圾回收器回收时收到一个通知。