Java集合

HashMap底层结构是什么？如何解决哈希冲突？

**底层结构：**

JDK1.8之前：数组+链表结构

JDK1.8之后：数组+链表+红黑树。

* 当链表长度≥8且数组容量≥64时，链表会转换为红黑树，以优化查询效率（时间复杂度从O(n)降至O(logn)）
* 红黑树退化条件：当链表长度≤6时，红黑树会退化为链表。

**解决哈希冲突的机制：**

1. 拉链法：哈希冲突发生时，以链表形式存储在同一个数组索引
2. 红黑树优化（JDK1.8后）：当链表长度≥8且数组容量≥64时，链表转换为红黑树
   1. 扩容：通过增加数组容量，重新分配节点，减少哈希冲突，适合小容量场景。
   2. 树化：通过将链表转换为红黑树，优化查询性能，适合大容量场景。
3. 哈希函数优化：
   1. 二次扰动函数：混合高位和低位、优化索引分布、减少哈希冲突  
      由于先计算哈希值、再计算索引，所以如果哈希值的高位变化较大，但低位变化较小，直接用 (n-1) & hash 计算索引时，低位相同的键会映射到同一位置。通过二次扰动函数，将哈希值的高位和低位混合，使哈希值的每一位都参与索引计算，减少冲突概率  
      步骤1：获取 key.hashCode()，记为 h。

步骤2：将 h 右移16位（h >>> 16），得到 h 的高16位。

步骤3：将 h 与 h >>> 16 进行 异或运算（^），得到最终的哈希值。

* 1. 计算索引的公式为(n-1) & hash，n是数组容量，确保索引分布均匀

1. 动态扩容机制：当元素数量超过阈值（容量×负载因子，默认0.75）时，数组扩容为原来的2倍，重新分配节点以减少冲突
   1. 负载因子过高会增加哈希冲突，过低则导致频繁扩容
   2. 数组容量为2的幂次：保证(n-1) & hash等价于取模运算，同时扩容时只需判断高位是否为1来分配新位置（无需重新计算所有哈希）

详细描述HashMap的扩容机制（触发条件、rehash过程）。

为什么HashMap线程不安全？举例说明并发问题场景。

**HashMap线程不安全的原因**

1. **数据覆盖**

多个线程同时进行插入、删除或修改操作时，可能会导致数据覆盖或丢失。

例如，两个线程同时插入不同的键值对，可能会覆盖对方的操作。

1. **扩容死循环**

在JDK 1.7中，HashMap的扩容是通过头插法将链表节点迁移到新数组中的。

在并发扩容时，多个线程同时操作链表，可能会导致链表形成环形结构，从而引发死循环。

1. **树化异常**

在JDK 1.8中，链表转红黑树的操作是线程不安全的。

多个线程同时进行树化操作，可能会导致红黑树结构异常。

**如何解决HashMap线程不安全的问题？**

1. **使用 Collections.synchronizedMap**：在方法级别加锁，性能低。
2. **使用 ConcurrentHashMap**：采用分段锁（JDK 1.7）或CAS+ synchronized（JDK 1.8），性能高。

红黑树在HashMap中的具体应用条件是什么？

为什么Java 8将链表转红黑树的阈值设为8？

HashMap的负载因子为什么默认是0.75？

如何设计一个自定义类作为HashMap的键？

HashMap的遍历方式有哪几种？哪种效率更高？

Java并发（ConcurrentHashMap、CAS）

ConcurrentHashMap如何实现线程安全？JDK7和JDK8的区别？

CAS的原理是什么？存在哪些缺陷？

AQS的核心机制是什么？举一个基于AQS实现的工具类。

为什么ConcurrentHashMap的size()方法需要特殊处理？

自旋锁和互斥锁在CAS中的协同作用？

如何用CAS实现一个无锁队列？

synchronized和ReentrantLock的性能对比？

线程池中的Worker类是如何利用AQS的？

SpringBoot/SSM框架

Spring AOP的实现原理？JDK动态代理和CGLIB的区别？

Spring Bean的生命周期和作用域有哪些？

SpringBoot自动配置的实现机制（@EnableAutoConfiguration）？

如何解决循环依赖问题？三级缓存机制详解。

Spring事务失效的常见场景有哪些？

MyBatis的#{}和${}的区别及SQL注入问题。

SpringMVC请求处理流程中的HandlerAdapter作用？

SpringBoot如何集成Tomcat容器？

MySQL

B+树索引的查询过程？为什么不用哈希索引？

事务隔离级别和MVCC的实现原理？

Redo Log和Undo Log的作用及写入机制？

覆盖索引和索引下推的区别？

死锁的产生条件及排查方法（show engine innodb status）。

Change Buffer对写操作的优化原理？

分库分表后全局ID生成方案有哪些？

主从同步延迟的解决方案？

Redis

Redis单线程模型为什么能高效处理请求？

缓存雪崩/穿透/击穿的解决方案？

RDB和AOF的优缺点及混合持久化机制？

Redis Cluster的slot分配算法？

大Key问题如何定位和解决？

布隆过滤器的实现原理及误差率计算？

Redis事务和Lua脚本的原子性区别？

热Key问题的解决方案（本地缓存/分片）？

网络协议（TCP/IP）

TCP三次握手的详细过程和状态变化？

HTTP/1.1到HTTP/2的核心改进有哪些？

TIME\_WAIT状态的作用及过多时的解决方案？

TCP快速重传和超时重传的区别？

HTTPS的TLS握手过程（ECDHE算法）？

QUIC协议如何解决队头阻塞问题？

TCP拥塞控制中的BBR算法原理？

WebSocket协议如何实现全双工通信？

操作系统

进程和线程的区别？协程的优势是什么？

虚拟内存的作用及页面置换算法（LRU实现）？

select/poll/epoll的区别及边缘触发模式？

用户态和内核态切换的开销来源？

自旋锁在单核CPU下是否有意义？

mmap系统调用的实现原理？

进程间通信共享内存和消息队列的对比？

零拷贝技术（sendfile）的实现原理？

Linux命令

如何用grep查找包含"error"的日志并统计次数？

awk如何实现按列求和（{sum+=$3} END{print sum}）？

如何用sed批量替换文件中的字符串？

strace和perf工具的作用及使用场景？

lsof命令如何查看被删除但未释放的文件？

如何用tcpdump抓取指定端口的SYN包？

使用dd命令测试磁盘IO性能的方法？

如何通过/proc文件系统查看进程内存映射？

Nginx/Docker

Nginx反向代理和负载均衡配置示例？

Docker镜像分层机制及构建优化方法？

Docker网络模式（bridge/host/none）的区别？

Nginx热更新配置文件的原理（信号机制）？

如何用Docker Compose编排多容器应用？

Kubernetes中Pod和Deployment的关系？

容器与虚拟机的性能差异来源？

如何通过cgroups限制容器资源使用？

Nginx

Nginx 的反向代理是如何工作的？

反向代理的基本概念

Nginx 通过监听客户端请求并将其转发到后端服务器，隐藏后端服务器的实际地址。

如何在 Nginx 中实现负载均衡？

负载均衡配置，如 upstream 指令

支持的算法：轮询（默认）、加权轮询、IP 哈希

Nginx 如何优化静态资源的加载速度？

使用缓存（proxy\_cache 或 fastcgi\_cache）

配置 gzip 压缩、HTTP/2、浏览器缓存（expires）

Docker

Docker 的核心组件有哪些？它们的作用是什么？

Docker Engine：运行容器的引擎

Docker Images：容器运行所需的文件和依赖

Docker Containers：运行的容器实例

Docker Hub：公共镜像仓库

什么是 Dockerfile？列出常见指令及其用途？

Dockerfile 是用于构建镜像的文件

常见指令

FROM：指定基础镜像

RUN：运行命令

CMD：容器启动时执行的命令

Docker 的数据卷是什么？如何使用？

数据卷是持久化存储的数据，用于在容器之间共享数据。

创建：docker volume create my\_volume

挂载：docker run -v my\_volume:/data my\_container

容器化技术

容器与虚拟机的区别是什么？

容器是轻量级的，直接共享宿主机的操作系统内核。

虚拟机需要完整的操作系统，开销较大。

Kubernetes 如何管理容器化应用？

Kubernetes 提供容器编排功能，包括自动化部署、扩展、负载均衡和服务发现。

容器化技术的优点有哪些？

提高开发效率

易于迁移

更好的资源利用和隔离性

Devops

什么是 CI/CD？它在 DevOps 中的作用是什么？

CI（持续集成）：自动化构建和测试代码

CD（持续部署/交付）：将通过测试的代码自动部署到生产环境

DevOps 常用工具有哪些？

CI/CD 工具：Jenkins、GitLab CI、CircleCI

配置管理：Ansible、Chef、Puppet

监控：Prometheus、Grafana

什么是基础设施即代码（IaC）？

使用代码（如 YAML、Terraform）定义和管理基础设施

优点：自动化、可重复性、可版本化