

# 高频面试题

根据超过 1500 篇真实面经整理的腾讯，阿里，字节跳动，Shopee，美团，滴滴高频面试题

## 目录

- [腾讯](#)
  - Java
  - C++
  - 操作系统
  - 网络协议
  - 算法
  - 数据库
  - 系统设计
  - 非技术
  - 前端
- [阿里](#)
  - Java
  - C++
  - 操作系统
  - 网络协议
  - 算法
  - 数据库
  - 系统设计
  - 非技术
  - 前端
- [字节跳动](#)
  - Java
  - C++
  - 操作系统
  - 网络协议
  - 算法
  - 数据库
  - 系统设计
  - 非技术
  - 前端
- [Shopee](#)
  - Java
  - C++

- 操作系统
- 网络协议
- 算法
- 数据库
- 系统设计
- 非技术
- 前端

#### ▪ 美团

- Java
- C++
- 操作系统
- 网络协议
- 算法
- 数据库
- 系统设计
- 非技术
- 前端

#### ▪ 滴滴

- Java
- C++
- 操作系统
- 网络协议
- 算法
- 数据库
- 系统设计
- 非技术
- 前端

## 腾讯

### # Java

- ▲ 38 HashMap 与 ConcurrentHashMap 的实现原理是怎样的？ConcurrentHashMap 是如何保证线程安全的？
- ▲ 27 volatile 关键字解决了什么问题，它的实现原理是什么？
- ▲ 26 Java 中垃圾回收机制中如何判断对象需要回收？常见的 GC 回收算法有哪些？
- ▲ 26 synchronized 关键字底层是如何实现的？它与 Lock 相比优缺点分别是什么？
- ▲ 24 简述 JVM 的内存模型 JVM 内存是如何对应到操作系统内存的？
- ▲ 20 集合类中的 List 和 Map 的线程安全版本是什么，如何保证线程安全的？
- ▲ 15 String 类能不能被继承？为什么？
- ▲ 14 Java 线程和操作系统的线程是怎么对应的？Java线程是怎样进行调度的？
- ▲ 11 简述 BIO, NIO, AIO 的区别
- ▲ 11 实现单例设计模式（懒汉，饿汉）
- ▲ 8 == 和 equals() 的区别？

- ▲ 8 简述 Spring AOP 的原理
- ▲ 6 简述 Synchronized, Volatile, 可重入锁的不同使用场景及优缺点
- ▲ 2 简述 Java 的 happen before 原则
- ▲ 1 SpringBoot 是如何进行自动配置的?

## # C++

- ▲ 16 C++ 中智能指针和指针的区别是什么?
- ▲ 15 简述 C++ 右值引用与转移语义
- ▲ 14 简述 vector 的实现原理
- ▲ 14 简述 C++ 中智能指针的特点, 简述 new 与 malloc 的区别
- ▲ 13 C++ 11 有什么新特性
- ▲ 11 STL 中 vector 与 list 具体是怎么实现的? 常见操作的时间复杂度是多少?
- ▲ 7 编译时链接有几种方式? 静态链接和动态链接的区别是什么?
- ▲ 5 深拷贝与浅拷贝区别是什么?
- ▲ 3 类默认的构造函数是什么?
- ▲ 2 const、static 关键字有什么区别
- ▲ 2 只定义析构函数, 会自动生成哪些构造函数?
- ▲ 1 简述 C++ 中内存对齐的使用场景
- ▲ 1 简述 C++ 从代码到可执行二进制文件的过程
- ▲ 1 简述 STL 中的 map 的实现原理

## # 操作系统

- ▲ 34 进程和线程之间有什么区别?
- ▲ 29 进程间有哪些通信方式?
- ▲ 27 简述 socket 中 select 与 epoll 的使用场景以及区别, epoll 中水平触发以及边缘触发有什么不同?
- ▲ 26 Linux 进程调度中有哪些常见算法以及策略?
- ▲ 18 操作系统如何申请以及管理内存的?
- ▲ 12 简单介绍进程调度的算法
- ▲ 11 简述 Linux 系统态与用户态, 什么时候会进入系统态?
- ▲ 11 简述 LRU 算法及其实现方式
- ▲ 11 线程间有哪些通信方式?
- ▲ 8 简述同步与异步的区别, 阻塞与非阻塞的区别
- ▲ 8 简述操作系统如何进行内存管理
- ▲ 7 简述操作系统中的缺页中断
- ▲ 3 简述操作系统中 malloc 的实现原理
- ▲ 2 BIO、NIO 有什么区别? 怎么判断写文件时 Buffer 已经写满? 简述 Linux 的 IO 模型
- ▲ 1 进程空间从高位到低位都有些什么?

## # 网络协议

- ▲ 41 简述 TCP 三次握手以及四次挥手的流程。为什么需要三次握手以及四次挥手?
- ▲ 32 RestFul 与 RPC 的区别是什么? RestFul 的优点在哪里?
- ▲ 29 HTTP 与 HTTPS 有哪些区别?
- ▲ 26 RestFul 是什么? RestFul 请求的 URL 有什么特点?
- ▲ 23 一次 HTTP 的请求过程中发生了什么?
- ▲ 19 TCP 与 UDP 在网络协议中的哪一层, 他们之间有什么区别?
- ▲ 18 TCP 中常见的拥塞控制算法有哪些?
- ▲ 17 TCP 怎么保证可靠传输?
- ▲ 17 从系统层面上, UDP如何保证尽量可靠?
- ▲ 8 TCP 的 keepalive 了解吗? 说一说它和 http 的 keepalive 的区别?
- ▲ 8 简述 TCP 滑动窗口以及重传机制
- ▲ 8 简述 HTTP 1.0, 1.1, 2.0 的主要区别
- ▲ 7 简述 TCP 的 TIME\_WAIT
- ▲ 5 HTTP 的方法有哪些?
- ▲ 4 简述 TCP 协议的延迟 ACK 和累计应答
- ▲ 1 简述 TCP 的报文头部结构
- ▲ 1 简述 TCP 半连接发生场景
- ▲ 1 什么是 SYN flood, 如何防止这类攻击?

## # 算法

- ▲ 29 10亿个数中如何高效地找到最大的一个数以及最大的第 K 个数
- ▲ 21 最大子序和 (Leetcode)
- ▲ 20 爬楼梯 (Leetcode)
- ▲ 18 用 Rand7() 实现 Rand10() (Leetcode)
- ▲ 15 AVL 树和红黑树有什么区别?
- ▲ 15 给定一个包含 40亿 个无符号整数的大型文件, 使用最多 1G 内存, 对此文件进行排序
- ▲ 14 合并两个有序链表 (Leetcode)
- ▲ 14 路径总和 (Leetcode)
- ▲ 14 如果通过一个不均匀的硬币得到公平的结果?
- ▲ 13 实现快速排序
- ▲ 13 数组中的第 K 个最大元素 (Leetcode)
- ▲ 13 10亿条数据包括 id, 上线时间, 下线时间, 请绘制每一秒在线人数的曲线图
- ▲ 13 删除排序链表中的重复元素 (Leetcode)
- ▲ 11 有序链表插入的时间复杂度是多少?
- ▲ 10 Hash 表常见操作的时间复杂度是多少? 遇到 Hash 冲突是如何解决的?
- ▲ 9 环形链表 (Leetcode)
- ▲ 6 常用的限流算法有哪些? 简述令牌桶算法原理
- ▲ 4 简述常见的负载均衡算法
- ▲ 1 反转链表 (Leetcode)
- ▲ 1 第一个只出现一次的字符 (Leetcode)

## # 数据库

- ▲ 58 MySQL 为什么使用 B+ 树来作索引，对比 B 树它的优点和缺点是什么？
- ▲ 31 数据库的事务隔离级别有哪些？各有哪些优缺点？
- ▲ 24 什么是数据库事务，MySQL 为什么会使用 InnoDB 作为默认选项
- ▲ 21 什么情况下会发生死锁，如何解决死锁？
- ▲ 21 简述乐观锁以及悲观锁的区别以及使用场景
- ▲ 19 聚簇索引和非聚簇索引有什么区别？什么情况用聚集索引？
- ▲ 18 简述脏读和幻读的发生场景，InnoDB 是如何解决幻读的？
- ▲ 17 唯一索引与普通索引的区别是什么？使用索引会有哪些优缺点？
- ▲ 11 Redis 如何实现延时队列，分布式锁的实现原理
- ▲ 9 简述 Redis 中如何防止缓存雪崩和缓存击穿
- ▲ 9 简述 Redis 持久化中 rdb 以及 aof 方案的优缺点
- ▲ 6 MySQL 的索引什么情况下会失效？
- ▲ 5 简述 MySQL 的主从同步机制，如果同步失败会怎么样？
- ▲ 5 简述数据库中的 ACID 分别是什么？
- ▲ 4 Kafka 发送消息是如何保证可靠性的？
- ▲ 3 简述 Redis 中跳表的应用以及优缺点
- ▲ 1 假设 Redis 的 master 节点宕机了，你会怎么进行数据恢复？
- ▲ 1 假设建立联合索引 (a, b, c) 如果对字段 a 和 c 查询，会用到这个联合索引吗？

## # 系统设计

- ▲ 9 电商系统中，如何实现秒杀功能？如何解决商品的超卖问题？
- ▲ 1 假如明天是活动高峰？QPS 预计会翻10倍，你要怎么做？

## # 非技术

- ▲ 14 对加班有什么看法？
- ▲ 10 你的优势和劣势是什么？
- ▲ 7 与同事沟通的时候，如果遇到冲突了如何解决？
- ▲ 1 最近一年内遇到的最有挑战的事情是什么？

## # 前端

- ▲ 20 Vue 中双向数据绑定的实现原理是怎样的？
- ▲ 11 简述 Javascript 原型以及原型链
- ▲ 11 简述 Vue 的生命周期
- ▲ 11 简述浏览器的缓存机制
- ▲ 9 简述 diff 算法的实现机制和使用场景
- ▲ 9 简述虚拟 dom 实现原理
- ▲ 7 简述图片的懒加载原理
- ▲ 6 简述 Javascript 中的防抖与节流的原理并尝试实现
- ▲ 5 什么是闭包，什么是立即执行函数，它的作用是什么？简单说一下闭包的使用场景
- ▲ 4 简述浏览题事件循环机制

- ▲ 4 localStorage 与 cookie 的区别是什么？

## 阿里

### # Java

- ▲ 49 HashMap 与 ConcurrentHashMap 的实现原理是怎样的？ConcurrentHashMap 是如何保证线程安全的？
- ▲ 41 Java 中垃圾回收机制中如何判断对象需要回收？常见的 GC 回收算法有哪些？
- ▲ 34 简述 Synchronized，Volatile，可重入锁的不同使用场景及优缺点
- ▲ 30 synchronized 关键字底层是如何实现的？它与 Lock 相比优缺点分别是什么？
- ▲ 26 JVM 中内存模型是怎样的，简述新生代与年老代的区别？
- ▲ 20 简述 Spring AOP 的原理
- ▲ 19 实现单例设计模式（懒汉，饿汉）
- ▲ 17 Java 是如何实现线程安全的
- ▲ 16 简述 ArrayList 与 LinkedList 的底层实现以及常见操作的时间复杂度
- ▲ 16 Java 类的加载流程是怎样的？什么是双亲委派机制？
- ▲ 15 简述 Java 的反射机制
- ▲ 14 Java 中 sleep() 与 wait() 的区别
- ▲ 13 Java 线程池里的 arrayblockingqueue 与 linkedblockingqueue 的使用场景和区别
- ▲ 13 String 类能不能被继承？为什么？
- ▲ 11 Java 中接口和抽象类的区别
- ▲ 11 JVM 是怎么去调优的？简述过程和调优的结果
- ▲ 10 hashCode 和 equals 方法的联系
- ▲ 6 Spring MVC 的原理和流程
- ▲ 6 手写生产者消费者模型
- ▲ 5 Java 缓冲流 buffer 的用途和原理是什么？

### # C++

- ▲ 16 简述 vector 的实现原理
- ▲ 11 简述 C++ 中智能指针的特点，简述 new 与 malloc 的区别
- ▲ 10 简述 C++ 编译的过程
- ▲ 3 C++ 中虚函数与纯虚函数的区别
- ▲ 2 C++ 中什么是菱形继承问题？
- ▲ 1 C++ 的 vector 和 list 中，如果删除末尾的元素，其指针和迭代器如何变化？若删除的是中间的元素呢？
- ▲ 1 C++ 的重载和重写是如何实现的？

### # 操作系统

- ▲ 36 进程和线程之间有什么区别？
- ▲ 18 进程间有哪些通信方式？
- ▲ 17 简述几个常用的 Linux 命令以及他们的功能。
- ▲ 13 线程有多少种状态，状态之间如何转换
- ▲ 11 进程有多少种状态？

- ▲ 7 简述 socket 中 select 与 epoll 的使用场景以及区别，epoll 中水平触发以及边缘触发有什么不同？
- ▲ 5 Linux 下如何排查 CPU 以及 内存占用过多？
- ▲ 3 进程通信中的管道实现原理是什么？
- ▲ 3 Linux 下如何查看 CPU 荷载，正在运行的进程，某个端口对应的进程？
- ▲ 2 如何调试服务器内存占用过高的问题？
- ▲ 2 Linux 如何查看实时的滚动日志？

## # 网络协议

- ▲ 35 简述 TCP 三次握手以及四次挥手的流程。为什么需要三次握手以及四次挥手？
- ▲ 34 HTTP 与 HTTPS 有哪些区别？
- ▲ 24 TCP 与 UDP 在网络协议中的哪一层，他们之间有什么区别？
- ▲ 21 一次 HTTP 的请求过程中发生了什么？
- ▲ 19 TCP 中常见的拥塞控制算法有哪些？
- ▲ 15 DNS 查询服务器的基本流程是什么？DNS 劫持是什么？
- ▲ 11 Cookie和Session的关系和区别是什么？
- ▲ 7 TCP 怎么保证可靠传输？
- ▲ 4 TCP 中 SYN 攻击是什么？如何防止？
- ▲ 2 TCP 四次挥手的时候 CLOSE\_WAIT 的话怎么处理？
- ▲ 2 简述 WebSocket 是如何进行传输的

## # 算法

- ▲ 33 10亿个数中如何高效地找到最大的一个数以及最大的第 K 个数
- ▲ 19 两个 10G 大小包含 URL 数据的文件，最多使用 1G 内存，将这两个文件合并，并找到相同的 URL
- ▲ 15 二叉树的层序遍历
- ▲ 12 常用的排序方式有哪些，时间复杂度是多少？
- ▲ 12 用 Rand7() 实现 Rand10() (Leetcode)
- ▲ 11 使用递归及非递归两种方式实现快速排序
- ▲ 10 实现 LRU 算法，实现带有过期时间的 LRU 算法
- ▲ 10 如何随机生成不重复的 10个100 以内的数字？
- ▲ 9 二叉树的最近公共祖先 (Leetcode 236)
- ▲ 7 如何实现大数运算
- ▲ 4 反转链表 (Leetcode)
- ▲ 3 用栈实现队列 (Leetcode)
- ▲ 2 1000台 机器，每台机器 1000个 文件，每个文件存储了 10亿个 整数，如何找到其中最小的 1000个 值？
- ▲ 1 两个文件包含无序的数字，数字的大小范围是0-500w左右。如何求两个文件中的重复的数据？
- ▲ 1 最长公共子序列 (Leetcode)
- ▲ 1 链表倒数第K个数 (Leetcode)
- ▲ 1 按序打印 (Leetcode)

## # 数据库

- ▲ 52 MySQL 为什么使用 B+ 树来作索引，对比 B 树它的优点和缺点是什么？
- ▲ 33 数据库的事务隔离级别有哪些？各有哪些优缺点？
- ▲ 26 简述乐观锁以及悲观锁的区别以及使用场景
- ▲ 20 简述一致性哈希算法的实现方式及原理
- ▲ 18 简述脏读和幻读的发生场景，InnoDB 是如何解决幻读的？
- ▲ 16 简述 Redis 持久化中 rdb 以及 aof 方案的优缺点
- ▲ 14 什么情况下会发生死锁，如何解决死锁？
- ▲ 14 SQL 优化的方案有哪些，如何定位问题并解决问题？
- ▲ 9 Redis 的缓存淘汰策略有哪些？
- ▲ 6 简述数据库中什么情况下进行分库，什么情况下进行分表？
- ▲ 5 数据库索引的实现原理是什么？
- ▲ 4 为什么 Redis 在单线程下能如此快？
- ▲ 3 数据库查询中左外连接和内连接的区别是什么？
- ▲ 3 简述什么是最左匹配原则
- ▲ 2 数据库的读写分离的作用是什么？如何实现？
- ▲ 2 简述 Redis 中跳表的应用以及优缺点
- ▲ 1 Redis 中，sentinel 和 cluster 的区别和适用场景是什么？

## # 系统设计

- ▲ 16 什么是 CAP？什么是最终一致性？什么是幂等操作？
- ▲ 14 电商系统中，如何实现秒杀功能？如何解决商品的超卖问题？

## # 非技术

- ▲ 21 最近阅读哪些技术书籍，遇到技术问题是怎么去解决？
- ▲ 9 与同事沟通的时候，如果遇到冲突了如何解决？
- ▲ 8 你的优势和劣势是什么？
- ▲ 2 团队合作沟通中遇到过什么问题？
- ▲ 1 简单描述一下自己是怎么样的一个人？
- ▲ 1 为什么要离开现在的公司？
- ▲ 1 目前为止，坚持得最久的一件事情是什么？

## # 前端

- ▲ 11 promise 有哪些状态？简述 promise.all 的实现原理
- ▲ 8 简述 CORS 的用途以及基本设置
- ▲ 8 简述 JWT 的原理和校验机制
- ▲ 4 localStorage 与 cookie 的区别是什么？



# 字节跳动

## # Java

- ▲ 20 Java 中垃圾回收机制中如何判断对象需要回收？常见的 GC 回收算法有哪些？
- ▲ 18 synchronized 关键字底层是如何实现的？它与 Lock 相比优缺点分别是什么？
- ▲ 17 hashmap 和 hashtable 的区别是什么？
- ▲ 15 HashMap 与 ConcurrentHashMap 的实现原理是怎样的？ConcurrentHashMap 是如何保证线程安全的？
- ▲ 11 HashMap 实现原理，为什么使用红黑树？
- ▲ 9 hashMap 1.7 / 1.8 的实现区别
- ▲ 9 简述 Java 的反射机制
- ▲ 6 Java 线程间有多少通信方式？
- ▲ 6 简述 Synchronized，volatile，可重入锁的不同使用场景及优缺点
- ▲ 6 Java 类的加载流程是怎样的？什么是双亲委派机制？
- ▲ 5 简述常见的工厂模式以及单例模式的使用场景
- ▲ 5 JVM 中内存模型是怎样的，简述新生代与年老代的区别？
- ▲ 3 Java 常见锁有哪些？ReentrantLock 是怎么实现的？
- ▲ 3 ThreadLocal 实现原理是什么？
- ▲ 3 简述 Spring 的初始化流程
- ▲ 3 简述生产者消费者模型
- ▲ 1 Java 如何高效进行数组拷贝
- ▲ 1 CAS 实现原理是什么？
- ▲ 1 成员变量和方法的区别？
- ▲ 1 Java 中接口和抽象类的区别

## # C++

- ▲ 11 简述 C++ 右值引用与转移语义
- ▲ 11 const、static 关键字有什么区别
- ▲ 11 C++ 的多态是如何实现的？
- ▲ 11 C++ 中解释类模板和模板类的区别
- ▲ 8 C++ 中哪些函数不能被声明为虚函数？
- ▲ 4 C++ 中虚函数与纯虚函数的区别
- ▲ 3 简述 C++ 中内存对齐的使用场景
- ▲ 1 构造函数为什么不能被声明为虚函数？
- ▲ 1 什么是内存泄漏，怎么确定内存泄漏？

## # 操作系统

- ▲ 23 进程间有哪些通信方式？
- ▲ 15 操作系统如何申请以及管理内存的？
- ▲ 13 简述 socket 中 select 与 epoll 的使用场景以及区别，epoll 中水平触发以及边缘触发有什么不同？
- ▲ 9 操作系统中，虚拟地址与物理地址之间如何映射？
- ▲ 8 I/O 多路复用中 select, poll, epoll 之间有什么区别，各自支持的最大描述符上限以及原因是什么？

- ▲ 6 简述操作系统中的缺页中断
- ▲ 5 进程和线程之间有什么区别？
- ▲ 4 两个线程交替打印一个共享变量
- ▲ 4 简述 mmap 的使用场景以及原理
- ▲ 3 Linux 中虚拟内存和物理内存有什么区别？有什么优点？
- ▲ 2 什么时候会由用户态陷入内核态？
- ▲ 1 简述 traceroute 命令的原理
- ▲ 1 简述 Linux 零拷贝的原理
- ▲ 1 简单介绍进程调度的算法
- ▲ 1 什么情况下，进程会进行切换？
- ▲ 1 简述自旋锁与互斥锁的使用场景

## # 网络协议

- ▲ 13 TCP 中常见的拥塞控制算法有哪些？
- ▲ 11 TCP 怎么保证可靠传输？
- ▲ 9 一次 HTTP 的请求过程中发生了什么？
- ▲ 9 简述常见的 HTTP 状态码的含义（301，304，401，403）
- ▲ 9 简述 TCP 三次握手以及四次挥手的流程。为什么需要三次握手以及四次挥手？
- ▲ 8 TCP的拥塞控制具体是怎么实现的？UDP有拥塞控制吗？
- ▲ 8 简述 HTTPS 的加密与认证过程
- ▲ 8 什么是跨域，什么情况下会发生跨域请求？
- ▲ 5 DNS 查询服务器的基本流程是什么？DNS 劫持是什么？
- ▲ 4 简述对称与非对称加密的概念
- ▲ 4 简述 OSI 七层模型，TCP，IP 属于哪一层？
- ▲ 4 HTTP 的方法有哪些？
- ▲ 3 简述 TCP 滑动窗口以及重传机制
- ▲ 1 TCP四次挥手过程以及所处状态，为什么还需要有 time\_wait？
- ▲ 1 TCP 在什么情况下服务端会出现大量 CLOSE\_WAIT ？

## # 算法

- ▲ 22 给定 100G 的 URL 磁盘数据，使用最多 1G 内存，统计出现频率最高的 Top K 个 URL
- ▲ 19 10亿个数中如何高效地找到最大的一个数以及最大的第 K 个数
- ▲ 17 合并两个有序链表 (Leetcode)
- ▲ 17 64 匹马，8 个赛道，找出前 4 匹马最少需要比几次；
- ▲ 14 实现快速排序
- ▲ 14 两数相加 II (Leetcode 445)
- ▲ 12 搜索旋转排序数组 (Leetcode)
- ▲ 12 判断有环链表的环长度 (Leetcode)
- ▲ 12 旋转数组的最小数字 (Leetcode)
- ▲ 11 最大子序和 (Leetcode)
- ▲ 11 给定一个 foo 函数，60%的概率返回0，40%的概率返回1，如何利用 foo 函数实现一个 50% 返回 0 的函数？
- ▲ 11 搜索旋转排序数组 II (Leetcode)

- ▲ 10 K 个一组翻转链表 (Leetcode 25)
- ▲ 9 最长连续子序列 (Leetcode)
- ▲ 9 二叉树的最近公共祖先 (Leetcode 236)
- ▲ 8 最小栈 (Leetcode)
- ▲ 8 丑数 II (Leetcode 264)
- ▲ 8 路径总和 (Leetcode)
- ▲ 8 二叉树的锯齿形层次遍历 (Leetcode)
- ▲ 7 旋转数组 (Leetcode)

## # 数据库

- ▲ 22 MySQL 为什么使用 B+ 树来作索引，对比 B 树它的优点和缺点是什么？
- ▲ 15 数据库的事务隔离级别有哪些？各有哪些优缺点？
- ▲ 13 Redis 序列化有哪些方式？
- ▲ 12 简述 Redis 持久化中 rdb 以及 aof 方案的优缺点
- ▲ 12 简述 Redis 的哨兵机制
- ▲ 12 Redis 如何实现分布式锁？
- ▲ 11 简述 Redis 中如何防止缓存雪崩和缓存击穿
- ▲ 7 简述乐观锁以及悲观锁的区别以及使用场景
- ▲ 6 数据库有哪些常见索引？数据库设计的范式是什么？
- ▲ 4 Redis 有几种数据结构？Zset 是如何实现的？
- ▲ 4 Cookie和Session的关系和区别是什么？
- ▲ 4 Redis 中 key 的过期策略有哪些？
- ▲ 3 什么情况下会发生死锁，如何解决死锁？
- ▲ 3 并发事务会引发哪些问题？如何解决？
- ▲ 2 简述数据库中的 ACID 分别是什么？
- ▲ 2 MySQL 中 join 与 left join 的区别是什么？
- ▲ 1 简述 MySQL 三种日志的使用场景
- ▲ 1 模糊查询是如何实现的？

## # 系统设计

- ▲ 14 简述 CAP 理论
- ▲ 2 简述生产者消费者模式的流程
- ▲ 1 设计一个阻塞队列
- ▲ 1 简述 MapReduce 的原理
- ▲ 1 停车场有有限个车位，有多个车来抢车位，设计一个系统需要根据车辆进入和离开停车场的时间进行计费

## # 非技术

- ▲ 6 项目中最难的地方是哪里？你学习到了什么？

## # 前端

- ▲ 25 手写题库 <https://github.com/Mayandev/fe-interview-handwrite>
- ▲ 12 简述浏览器的渲染过程，重绘和重排在渲染过程中的哪一部分？
- ▲ 11 简述 diff 算法的实现机制和使用场景
- ▲ 9 什么是闭包，什么是立即执行函数，它的作用是什么？简单说一下闭包的使用场景
- ▲ 9 promise 有哪些状态？简述 promise.all 的实现原理
- ▲ 9 简述什么是 XSS 攻击以及 CSRF 攻击？
- ▲ 9 localStorage 与 cookie 的区别是什么？
- ▲ 8 Vue 中双向数据绑定的实现原理是怎样的？
- ▲ 8 简述 CSS 有哪些上下文类型？
- ▲ 5 简述 ES6 的新特性
- ▲ 5 了解过 Gulp Grunt 吗？简述他们的优势以及劣势
- ▲ 4 Javascript 可以保存的最大数值是多少？
- ▲ 4 优化首屏渲染的方式有哪几种？
- ▲ 3 JavaScript 中的严格模式是什么，有什么作用？

## Shopee

## # Java

- ▲ 14 Java 中垃圾回收机制中如何判断对象需要回收？常见的 GC 回收算法有哪些？
- ▲ 12 简述 ArrayList 与 LinkedList 的底层实现以及常见操作的时间复杂度
- ▲ 9 实现单例设计模式（懒汉，饿汉）
- ▲ 9 Java 类的加载流程是怎样的？什么是双亲委派机制？
- ▲ 8 HashMap 与 ConcurrentHashMap 的实现原理是怎样的？ConcurrentHashMap 是如何保证线程安全的？
- ▲ 8 HashMap 实现原理，为什么使用红黑树？
- ▲ 6 常用的排序方式有哪些，时间复杂度是多少？
- ▲ 3 volatile 关键字解决了什么问题，它的实现原理是什么？
- ▲ 2 JVM 中内存模型是怎样的，简述新生代与年老代的区别？
- ▲ 2 简述 SortedSet 实现原理
- ▲ 2 简述使用协程的优点
- ▲ 1 Java 编译后的 .class 文件包含了什么内容？
- ▲ 1 什么是公平锁？什么是非公平锁？
- ▲ 1 如何判断一个 Hash 函数好不好？

## # 操作系统

- ▲ 10 进程和线程之间有什么区别？
- ▲ 7 进程通信中的管道实现原理是什么？
- ▲ 5 进程间有哪些通信方式？
- ▲ 5 Linux 下如何排查 CPU 以及内存占用过多？
- ▲ 3 简述 Linux 虚拟内存的页面置换算法
- ▲ 3 简单介绍进程调度的算法
- ▲ 2 LVS 的 NAT、TUN、DR 原理及区别

- ▲ 1 系统调用的过程是怎样的？操作系统是通过什么机制触发系统调用的？

## # 网络协议

- ▲ 19 HTTP 与 HTTPS 有哪些区别？
- ▲ 12 TCP 与 UDP 在网络协议中的哪一层，他们之间有什么区别？
- ▲ 8 简述什么是 XSS 攻击以及 CSRF 攻击？
- ▲ 6 简述 TCP 中的拥塞控制与滑动窗口机制
- ▲ 5 简述 TCP 三次握手以及四次挥手的流程。为什么需要三次握手以及四次挥手？
- ▲ 5 简述 HTTP 1.0, 1.1, 2.0 的主要区别
- ▲ 4 SSL握手流程为什么要使用对称密钥？
- ▲ 3 简述 JWT 的原理和校验机制
- ▲ 3 TCP 的 keepalive 了解吗？说一说它和 http 的 keepalive 的区别？
- ▲ 2 一次 HTTP 的请求过程中发生了什么？
- ▲ 2 DNS 查询服务器的基本流程是什么？DNS 劫持是什么？

## # 算法

- ▲ 8 快速排序的空间复杂度是多少？时间复杂度的最好最坏的情况是多少，有哪些优化方案？
- ▲ 5 环形链表 (Leetcode)
- ▲ 5 用栈实现队列 (Leetcode)
- ▲ 4 AVL 树和红黑树有什么区别？
- ▲ 4 10亿个数中如何高效地找到最大的一个数以及最大的第 K 个数
- ▲ 3 数组中的逆序对 (Leetcode)
- ▲ 2 如何从一个数组输出随机数组
- ▲ 2 寻找旋转排序数组中的最小值 (leetcode)
- ▲ 2 红黑树是怎么实现平衡的？它的优点是什么？
- ▲ 1 和为 K 的子数组 (Leetcode)

## # 数据库

- ▲ 13 简述乐观锁以及悲观锁的区别以及使用场景
- ▲ 12 数据库的事务隔离级别有哪些？各有哪些优缺点？
- ▲ 11 为什么 Redis 在单线程下能如此快？
- ▲ 9 MySQL 为什么使用 B+ 树来作索引，对比 B 树它的优点和缺点是什么？
- ▲ 5 什么情况下会发生死锁，如何解决死锁？
- ▲ 3 数据库有哪些常见索引？数据库设计的范式是什么？
- ▲ 3 简述 SQL 中左连接和右连接的区别
- ▲ 3 数据库的读写分离的作用是什么？如何实现？
- ▲ 2 简述数据库中的 ACID 分别是什么？
- ▲ 2 什么是数据库事务，MySQL 为什么会使用 InnoDB 作为默认选项
- ▲ 2 Redis 中，sentinel 和 cluster 的区别和适用场景是什么？

## # 系统设计

### ▲ 8 简述 CAP 理论

## # 非技术

- ▲ 7 你对 Shopee 了解多少？你选择 Shopee 的原因是什么？
- ▲ 1 你的老师和同学是如何评价你的？
- ▲ 1 最近在看什么书以及技术文章？

## 美团

## # Java

- ▲ 22 HashMap 与 ConcurrentHashMap 的实现原理是怎样的？ConcurrentHashMap 是如何保证线程安全的？
- ▲ 22 Java 中垃圾回收机制中如何判断对象需要回收？常见的 GC 回收算法有哪些？
- ▲ 19 Java 的线程有哪些状态，转换关系是怎么样？
- ▲ 18 synchronized 关键字底层是如何实现的？它与 Lock 相比优缺点分别是什么？
- ▲ 15 Java 怎么防止内存溢出
- ▲ 13 简述常见的工厂模式以及单例模式的使用场景
- ▲ 12 简述 BIO, NIO, AIO 的区别
- ▲ 11 Java 类的加载流程是怎样的？什么是双亲委派机制？
- ▲ 11 简述 ArrayList 与 LinkedList 的底层实现以及常见操作的时间复杂度
- ▲ 11 实现单例模式
- ▲ 9 volatile 关键字解决了什么问题，它的实现原理是什么？
- ▲ 9 hashCode 和 equals 方法的联系
- ▲ 9 什么是重写和重载？
- ▲ 7 简述 JVM 的内存模型 JVM 内存是如何对应到操作系统内存的？
- ▲ 7 JVM 中内存模型是怎样的，简述新生代与年老代的区别？
- ▲ 5 简述装饰者模式以及适配器模式
- ▲ 5 什么情况下会发生死锁，如何解决死锁？
- ▲ 4 简述 Java 中 final 关键字的作用
- ▲ 4 ThreadLocal 实现原理是什么？

## # 操作系统

- ▲ 19 进程和线程之间有什么区别？
- ▲ 12 进程间有哪些通信方式？
- ▲ 4 多线程和多进程的区别是什么？
- ▲ 3 为什么进程切换慢，线程切换快？

## # 网络协议

- ▲ 18 TCP 怎么保证可靠传输?
- ▲ 17 HTTP 与 HTTPS 有哪些区别?
- ▲ 15 TCP 与 UDP 在网络协议中的哪一层, 他们之间有什么区别?
- ▲ 14 简述 TCP 三次握手以及四次挥手的流程。为什么需要三次握手以及四次挥手?
- ▲ 13 TCP长连接和短连接有那么不同的使用场景?
- ▲ 10 HTTP 中 GET 和 POST 区别
- ▲ 8 TCP 与 UDP 在网络协议中的哪一层, 他们之间有什么区别?
- ▲ 4 什么是 ARP 协议?

## # 算法

- ▲ 15 实现快速排序
- ▲ 15 简述 LRU 算法及其实现方式
- ▲ 14 按序打印 (Leetcode)
- ▲ 12 二叉树的前序遍历 (Leetcode)
- ▲ 11 数组中的第 K 个最大元素 (Leetcode)
- ▲ 11 旋转图像 (Leetcode)
- ▲ 11 剑指 Offer 10- II. 青蛙跳台阶问题
- ▲ 11 链表倒数第K个数 (Leetcode)
- ▲ 9 多数元素 (Leetcode)
- ▲ 7 用栈实现队列 (Leetcode)
- ▲ 6 搜索旋转排序数组 (Leetcode)
- ▲ 6 两数相加

## # 数据库

- ▲ 19 简述乐观锁以及悲观锁的区别以及使用场景
- ▲ 17 简述 MySQL 常见索引数据, 介绍一下覆盖索引
- ▲ 17 简述事务的四大特性
- ▲ 15 Redis 如何实现分布式锁?
- ▲ 14 简述 Redis 的线程模型以及底层架构设计
- ▲ 14 简述 Redis 持久化中 rdb 以及 aof 方案的优缺点
- ▲ 14 为什么 Redis 在单线程下能如此快?
- ▲ 13 MySQL 为什么使用 B+ 树来作索引, 对比 B 树它的优点和缺点是什么?
- ▲ 13 简述脏读和幻读的发生场景, InnoDB 是如何解决幻读的?
- ▲ 11 数据库的事务隔离级别有哪些? 各有哪些优缺点?
- ▲ 11 简述 Redis 的过期机制和内存淘汰策略
- ▲ 10 聚簇索引和非聚簇索引有什么区别? 什么情况用聚集索引?
- ▲ 10 简述 Redis 的哨兵机制
- ▲ 10 简述 Redis 如何处理热点 key 访问
- ▲ 9 什么是数据库事务, MySQL 为什么会使用 InnoDB 作为默认选项
- ▲ 8 MySQL 有哪些常见的存储引擎?
- ▲ 4 简述 Redis 中常见类型的底层数据结构

## # 系统设计

- ▲ 17 电商系统中，如何实现秒杀功能？如何解决商品的超卖问题？
- ▲ 14 简述 CAP 理论

## # 非技术

- ▲ 11 下一份工作希望学习到什么？
- ▲ 11 项目中最难的地方是哪里？你学习到了什么？
- ▲ 9 团队合作沟通中遇到过什么问题？
- ▲ 9 最近在看什么书吗，有没有接触过什么新技术？
- ▲ 5 成长过程中影响你最深的事件和人

## # 前端

- ▲ 12 简述 Vue 的生命周期
- ▲ 12 手写题库 <https://github.com/Mayandev/fe-interview-handwrite>
- ▲ 11 简述项目打包和发布的流程
- ▲ 11 简述 diff 算法的实现机制和使用场景
- ▲ 8 简述 react 的生命周期以及通信方式
- ▲ 8 什么是闭包，什么是立即执行函数，它的作用是什么？简单说一下闭包的使用场景
- ▲ 7 简述浏览器的垃圾回收机制
- ▲ 7 移动端适配有哪些方案？
- ▲ 7 简述浏览器的渲染过程，重绘和重排在渲染过程中的哪一部分？
- ▲ 6 MVC 模型和 MVVM 模型的区别
- ▲ 5 了解 xss 攻击吗？如何防止 xss 攻击？
- ▲ 4 简述 React 中的 Effect Hook 机制
- ▲ 4 简述 React setState 原理
- ▲ 3 简述强缓存与协商缓存的区别和使用场景

## 滴滴

## # Java

- ▲ 8 什么是重写和重载？
- ▲ 7 简述并实现工厂模式，工厂模式有什么常见问题？
- ▲ 5 Java 常见锁有哪些？ReentrantLock 是怎么实现的？
- ▲ 3 HTTP 中 GET 和 POST 区别
- ▲ 2 Java 有几种基本数据类型，分别占多少字节？
- ▲ 1 Java 中 int 的最大值是多少？
- ▲ 1 简述封装、继承、多态的特性及使用场景



## # 操作系统

- ▲ 1 malloc 创建的对象在堆还是栈中？

## # 网络协议

- ▲ 13 RestFul 是什么？RestFul 请求的 URL 有什么特点？
- ▲ 9 简述 HTTP 1.0, 1.1, 2.0 的主要区别
- ▲ 6 什么是跨域，什么情况下会发生跨域请求？
- ▲ 6 Cookie和Session的关系和区别是什么？
- ▲ 6 简述 JWT 的原理和校验机制
- ▲ 5 如何设计 API 接口使其实现幂等性？
- ▲ 3 简述 TCP 滑动窗口以及重传机制
- ▲ 3 TCP 中 SYN 攻击是什么？如何防止？

## # 算法

- ▲ 11 实现快速排序
- ▲ 8 反转链表 (Leetcode)
- ▲ 8 爬楼梯 (Leetcode)
- ▲ 8 剑指 Offer 10- II. 青蛙跳台阶问题
- ▲ 5 快速排序的空间复杂度是多少？时间复杂度的最好最坏的情况是多少，有哪些优化方案？
- ▲ 4 实现归并排序

## # 数据库

- ▲ 11 联合索引的存储结构是什么？
- ▲ 6 简述 Redis 中如何防止缓存雪崩和缓存击穿
- ▲ 3 聚簇索引和非聚簇索引有什么区别？什么情况用聚集索引？
- ▲ 3 Redis 如何实现分布式锁？

## # 前端

- ▲ 22 手写题库 <https://github.com/Mayandev/fe-interview-handwrite>
- ▲ 15 简述 Javascript 原型以及原型链
- ▲ 14 Vue 中双向数据绑定的实现原理是怎样的？
- ▲ 14 简述 Javascript 中的防抖与节流的原理并尝试实现
- ▲ 12 简述 diff 算法的实现机制和使用场景
- ▲ 12 简述 CORS 的用途以及基本设置
- ▲ 12 简述输入 URL 到浏览器显示的流程
- ▲ 12 如何减少页面渲染的时间？
- ▲ 10 简述浏览器的垃圾回收机制
- ▲ 10 简述浏览器的渲染过程，重绘和重排在渲染过程中的哪一部分？
- ▲ 10 简述 Javascript 中 this 的指向有哪些
- ▲ 9 简述 jsonp 的工作原理
- ▲ 9 简述 Javascript 事件冒泡和事件捕获原理

- ▲ 8 如何解决 CSS 类名重名?
- ▲ 8 简述发布订阅模式的实现方式以及原理
- ▲ 8 箭头函数和普通函数的区别是什么?
- ▲ 8 简述图片的懒加载原理
- ▲ 7 简述 Vue 的生命周期
- ▲ 7 简述常见异步编程方案 (promise, generator, async) 的原理
- ▲ 6 简述 Javascript 的柯里化与逆柯里化