常见面试问题整理（考研复试面试/计算机408+数据库基础概念）

一、数据结构与算法

Q：递归、迭代、分治、回溯、动规、贪心的概念

A：
递归的本质是将原问题拆分成具有相同性质的子问题，递归解法的特点有两个，分别是子问题拆分方程和终止条件。 递归的调用栈会有深度限制。

迭代和递归本质可以说是一样的， 只是我们模拟了递归的调用栈而已，因此迭代有时候会用到栈这样的数据结构。

分治算法的特征一般是先将原问题拆分成若干个子问题（分解），然后求解子问题（终止条件），最后将各个子问题的解合并，形成原问题的解（合并）。（在求解一些问题时，有很多重复的子问题，这样会导致算法非常低效，这个时候就可以加缓存，也就是带备忘录的递归解法）

动态规划的特征：
最优子结构：在自下而上的递推过程中，我们求得的每个子问题一定是全局最优解，既然它分解的子问题是全局最优解，那么依赖于它们解的原问题自然也是全局最优解。
重叠子问题：在求解原问题的时候，我们往往需要依赖其子问题，子问题依赖其子子问题，甚至可能同时依赖多个子问题，因此子问题之间是有重叠关系的。
状态初始化：因为动态规划是依赖子问题的，因此需要先初始化已知状态，从而才能自下而上的递推到原问题得解。
遍历状态集：首先根据需要求解的结果来确定状态集，然后遍历状态集，更新状态 dp
状态转移方程：也就是每一级子问题之间的关系式。

回溯算法是一种通过探索所有可能的候选解来找出所有的解的算法。如果候选解被确认不是一个解的话（或者至少不是最后一个解），回溯算法会通过在上一步进行一些变化抛弃该解，即回溯并且再次尝试。

贪心算法(代表算法有Prim, kruskal, Dijkstra, Huffman)是指，在对问题求解时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，它所做出的仅仅是在某种意义上的局部最优解。
贪心算法的特征：

最优子结构：贪心算法也是将原问题分解成多个性质相同的子问题，每个问题都是局部最优。
贪心选择性质：在做贪心选择时，我们直接做出当前子问题中看起来最优的解，这也是贪心算法和动态规划的不同之处。
遍历状态集：遍历状态集，做出局部最优选择，更新结果。
无后效性： 某个状态以后的过程不会影响以前的状态，只与当前状态有关

Q：DFS 的基本思想和 BFS 的基本思想

A：
DFS （深度优先搜索）类似于二叉树的先序遍历，它的基本思想是首先访问出发点并将其标记为已访问，然后选取与起始点邻接的未被访问的任意一个顶点标记并递归访问，再选择与上一标记顶点邻接的未被访问的顶点标记并递归访问， 以此重复进行。当一个顶点的所有邻接顶点都被访问过时，则依次退回到最近被访问过的顶点，若该顶点还有其他邻接顶点未被访问，则从这些未被访问的顶点中取一个并重复上述递归访问过程， 直到途中所有的顶点都被访问过为止。
BFS（广度优先搜索） 类似于树的层序遍历， 需要使用队列这一数据结构， 它的基本思想是首先访问起始顶点入队并标记为已访问，当队列不空时循环执行访问操作，即出队并依次检查出队顶点的所有邻接顶点，访问没有被访问过的邻接顶点并将其入队。当队列为空时跳出循环，广度优先搜索即完成。

Q：Kruskal 算法的基本思想

A：
首先构造一个只有 n 个顶点没有边的非连通图,给所有的边按值以从小到大的顺序排序,选择一个最小权值边,若该边的两个顶点在不同的连通分量上,加入到有效边中,否则舍去这条边,重新选择权值最小的边,以此重复下去,直到所有顶点在同一连通分量上。

Q：Prim算法的基本思想

A：
从某顶点 u0 出发，选择与它关联的具有最小权值的边(u0, v)，将其顶点加入到生成树的顶点集合U中。每一步从U中挑选一个顶点u，而另一个顶点不在U中的各个顶点中选择权值最小的边(u, v)，把它的顶点加入到U中，直到所有顶点都加入到生成树顶点集合U中为止。

Q：简述链表和数组的优缺点

A：
数组的优点： 随机访问性强， 查找速度快
数组的缺点： 插入和删除效率低， 可能浪费内存， 内存空间要求高，必须有足够的连续内存空间。数组大小固定，不能动态拓展
链表的优点： 插入删除速度快， 内存利用率高，不会浪费内存， 大小没有固定，拓展灵活。
链表的缺点： 不能随机查找，必须从第一个开始遍历，查找效率低

Q：Heap 和 Stack 的概念及区别

A：
一、堆栈空间分配区别：
1、栈（操作系统）：由操作系统自动分配释放 ，存放函数的参数值，局部变量的值等。其操作方式类似于数据结构中的栈；
2、堆（操作系统）： 一般由程序员分配释放， 若程序员不释放，程序结束时可能由 OS 回收，分配方式倒是类似于链表。
二、堆栈缓存方式区别：
1、栈使用的是一级缓存， 他们通常都是被调用时处于存储空间中，调用完毕立即释放；
2、堆是存放在二级缓存中，生命周期由虚拟机的垃圾回收算法来决定（并不是一旦成为孤儿对象就能被回收）。所以调用这些对象的速度要相对来得低一些。
三、堆栈数据结构区别：
堆（数据结构）：堆可以被看成是一棵树，如：堆排序；
栈（数据结构）：一种先进后出的数据结构。

Q：哈希表最好最坏情况下复杂度？

A：
O(1)和 O(n)， n 为表长。

Q：求二叉树的直径？

A：
两次 DFS，第一次找出距离 root 最远点，第二次以第一次结果为起点找出第二个点，这两点的距离即为直径。

Q：
A：

Q：
A：

二、操作系统

Q：进程和线程的区别

A：
根本区别：进程是资源分配最小单位，线程是程序执行的最小单位。
地址空间：进程有自己独立的地址空间，每启动一个进程，系统都会为其分配地址空间，建立数据表来维护代码段、堆栈段和数据段；线程没有独立的地址空间，同一进程的线程共享本进程的地址空间。
资源拥有：进程之间的资源是独立的；同一进程内的线程共享本进程的资源。
执行过程：每个独立的进程有一个程序运行的入口、顺序执行序列和程序入口。但是线程不能独立执行，必须依存在应用程序中，由应用程序提供多个线程执行控制。线程是处理机调度的基本单位，但是进程不是。由于程序执行的过程其实是执行具体的线程，那么处理机处理的也是程序相应的线程，所以处理机调度的基本单位是线程。
系统开销：进程执行开销大，线程执行开销小

Q：系统调用的定义？

A：
系统调用是 OS 与应用程序之间的接口，它是用户程序取得 OS 服务的唯一途径。 它与一般的过程调用的区别：运行在不同的系统状态， 调用程序运行在用户态，而被调用的程序运行在系统态，通过软中断机制，先由用户态转为系统态，经核心分析后，才转向相应的系统调用处理子程序；一般的过程调用返回后继续执行， 但对系统调用，当调用进程仍具有最高优先权时，才返回到调用进程继续处理，否则只能等被重新调度。

Q：解释一下管程

A：
管程是由一组局部变量、 对局部变量进行操作的一组过程和对局部变量进行初始化的语句序列组成。引入它的目的是因为 Wait/Singal 操作太过分散,对它的维护很麻烦且容易造成死锁。管程的特点是:管程的过程只能访问管程的局部变量,管程的局部变量只能由其过程来访问;任何时刻只能有一个进程进入管程执行;进程只能通管程提供的过程入口进入管程。

Q：在可变分区管理中,需要哪些硬件机制?

A：
采用可变分区方式管理时,一般均采用动态重定位方式装入作业。地址变换要靠硬件支持,主要是两个寄存器:基址寄存器和限长寄存器，限长寄存器存放作业所占分区的长度,基址寄存器则存放作业所占分区的起始地址,这两个值确定了一个分区的位置和大小。

转换时根据逻辑地址与限长值比较,如果不有超过这个值,表示访问地址合法,再加上基址寄存器中的值就得到了绝对地址了,否则形成“ 地址越界” 中断。

Q：中断和陷入有什么异同

A：
外中断时指来自处理机和内存外部的中断,如 I/O 中断、定时器中断、外部信号中断等。狭义上也叫中断;
内中断主要指在处理机和内存内部产生的中断,也称陷入,如校验错、页面失效、溢出、除数为零等。
中断和陷阱的主要区别:
(1) 陷入通常由处理机正在执行的现行指令引起,而中断则是由与现行指令无关的中断源引起的。
(2) 陷阱处理程序提供的服务为当前进程所用,而中断处理程序提供的服务则不是为了当前进程的。
管理时,一般均采用动态重定位方式装入作业。地址变换要靠硬件支持,主要是两个寄存器:基址寄存器和限长寄存器,限长寄存器存放作业所占分区的长度,基址寄存器则存放作业所占分区的起始地址,这两个值确定了一个分区的位置和大小。转换时根据逻辑地址与限长值比较,如果不有超过这个值,表示访问地址合法,再加上基址寄存器中的值就得到了绝对地址了,否则形成“ 地址越界” 中断。

Q：中断和系统调用区别？

A：
中断：解决处理器速度和硬件速度不匹配，是多道程序设计的必要条件。每个中断都有自己的数字标识，当中断发生时，指令计数器 PC 和处理机状态字 PSW中的内容自动压入处理器堆栈，同时新的 PC 和 PSW 的中断向量也装入各自的寄存器中。这时， PC 中包含的是该中断的中断处理程序的入口地址，它控制程序转向相应的处理，当中断处理程序执行完毕，该程序的最后一条 iret（中断返回），它控制着恢复调用程序的环境。
中断和系统调用的区别: 中断是由外设产生,无意的, 被动的 系统调用是由应用程序请求操作系统提供服务产生, 有意的,主动的。要从用户态通过中断进入内核态。
（联系）
中断过程：中断请求 中断响应 断点保护 执行中断服务程序 断点恢复 中断返回
系统调用过程：应用程序在用户态执行时请求系统调用，中断，从用户态进入内核态，在内核态执行相应的内核代码。

Q：操作系统的特点？

A：
共享：资源可被多个并发执行的进程使用
并发：可以在同一时间间隔处理多个进程，需要硬件支持
虚拟：将物理实体映射成为多个虚拟设备
异步：进程执行走走停停，每次进程执行速度可能不同，但 OS 需保证进程每次执行结果相同

Q：进程的三个组成部分？

A：
程序段、数据段、 PCB(Process Control Block)

Q：并发与并行区别？

A：
并发：同一间隔 并行：同一时刻

Q：进程切换的过程？

A：
保持处理机上下文 -> 更新 PCB -> 把 PCB 移入相应队列(就绪、阻塞) ->选择另一个进程并更新其 PCB -> 更新内存管理的数据结构 -> 恢复处理机上下文

Q：进程通信的方式

A：
1、低级通信方式
PV 操作（信号量机制）。
– P： wait(S)原语，申请 S 资源
– V： signal(S)原语，释放 S 资源
2、高级通信方式：以较高效率传输大量数据的通信方式
– 共享存储（使用同步互斥工具操作共享空间）
– 消息传递（进程间以格式化的消息进行数据交换，有中间实体，分为直接和间接两种，底层通过发送消息和接收消息两个原语实现）
– 管道通信（两个进程中间存在一个特殊的管道文件，进程的输入输出都通过管道，半双工通信）

Q：死锁的必要条件？

A：
– 互斥条件：资源在某一时刻只能被一个进程占有
– 不剥夺条件：进程所持有的资源在主动释放前不能被其他进程强行夺走
– 请求和占用条件：死锁进程必然是既持有资源又在申请资源的
– 循环等待条件：存在等待链，互相申请，互不释放

Q：死锁与饥饿的区别？

A：
– 都是资源分配问题
– 死锁是等待永远不会释放的资源，而饥饿申请的资源会被释放，只是永远不会分配给自己
– 一旦产生死锁，则死锁进程必然是多个，而饥饿进程可以只有一个
– 饥饿的进程可能处于就绪状态，而死锁进程一定是阻塞进程

Q：FCB 包含什么？

A：
文件指针：上次读写位置。
文件打开数：多少个进程打开了此文件。
文件磁盘位置。
文件的访问权限：创建、只读、读写等。

Q：页面置换算法？

A：
最佳置换算法 OPT
先进先出置换算法 FIFO
最近最久未使用算法 LRU
时钟算法 CLOCK
改进型时钟算法

Q：批处理作业调度算法？

A：
先来先服务 FCFS
最短作业优先 SJF
最高响应比优先 HRN
多级队列调度算法

Q：进程调度算法？

A：
先进先出 FIFO
时间片轮转算法 RR
最高优先级算法 HPF
多级队列反馈算法

Q：磁盘调度算法？

A：
先来先服务 FCFS
最短寻道时间优先 SSTF
扫描算法 SCAN
循环扫描算法 C-SCAN

Q：局部性原理

A：
程序的时间局部性是指程序在运行时呈现出局部性规律,在一段时间间隔内,程序的执行是局限在某个部份,所访问的存储空间也只局限在某个区域。
程序的空间局部性是指若一个存储单元被访问,那么它附近的单元也可能被访问,这是由于程序的顺序执行引起的。

Q：fork 一个进程和生成一个线程有什么区别？

A：
当你 fork 一个进程时，新的进程将执行和父进程相同的代码，只是在不同的内存空间中。
但当你在已有进程中生成一个线程时，它会生成一个新的代码执行路线，但共享同一个内存空间。

Q：
A：

Q：
A：

三、计算机网络

Q：IP 层的协议有哪些？

A：
ICMP 协议： ICMP 协议是指英文全称（Internet Control Message Protocol），就是网际控制信息协议。主要是用于补充 IP 传输数据报的过程中，发送主机无法确定数据报是否到达目标主机。 ICMP 报文分为出错报告报文和查询报文两种。若数据报不能到达目标主机，ICMP 出错报告报文可以以回送信息的方式，向源主机发去信息，并不能纠抄正数据报中的任何出错。除了出错报告， ICMP 还可以诊断出某些网络问题，这就是 ICMP 的查询报文。
IGMP 协议： IGMP 协议是指英文全称（Internet Group Management Protocol），网络组管理协议。主要用于建立和管理多播组，对 IP 分组广播进行控制

Q：简述网卡的功能

A：
1、网卡要进行串行/并行转换：网卡和局域网之间的通信是通过电缆或双绞线以串行传输方式进行的。而网卡和计算机之间的通信则是通过计算机主板上的 I/O 总线以并行传输方式进行。
2、网卡能实现以太网协议：在安装网卡时必须将管理网卡的设备驱动程序安装在计算机的操作系统中。这个驱动程序以后就会告诉网卡，应当从存储器的什么位置上将局域网传送过来的数据块存储下来。
3、网卡能处理正确的帧：当网卡收到一个有差错的帧时，它就将这个帧丢弃而不必通知它所插入的计算机。当网卡收到一个正确的帧时，它就使用中断来通知该计算机并交付给协议栈中的网络层。当计算机要发送一个 IP 数据包时，它就由协议栈向下交给网卡组装成帧后发送到局域网。

Q：TCP 和 UDP 的区别？

A：
TCP 可靠， UDP 不可靠。
TCP 只支持点对点服务， UDP 可以一对一、一对多、多对一和多对多。
TCP 面向连接， UDP 无连接。
UDP 有较好的实时性，工作效率比TCP 高。
TCP 对系统资源要求多， UDP 则无。

Q：UDP 的优点？

A：
发送前无需连接，减少了开销和时延，首部开销小，无拥塞控制，方便实时应用，不保证可靠交付，无需维持连接状态表。 UDP 的可靠性要通过应用层来控制。

Q：RIP 和 OSPF

A：
RIP(Routing Information Protocol)在应用层，最大站点数为 15
OSPF(Open Shortest Path First)网络层，洪泛法，迪杰斯特拉算法

Q：DNS 工作过程？

A：
应用层协议，使用 UDP。分为迭代查询和递归查询。采用分布式集群的工作方式，防止单点故障，增加通信容量。
迭代：主机访问本地域名服务器，若缓存没有 IP 则本地域名服务器进一步向其他根域名服务器查询。
递归：主机分别向多个服务器发出查询请求。

Q：解释 TCP/IP 的三次握手

A：
step1：首先客户机 TCP 向服务器 TCP 发送连接请求报文，报文中 SYN=1,随机发送一个序号 seq=x；
step2：服务器 TCP 接收到连接请求报文后，若同意连接，则返回确认报文，且为该 TCP 连接分配 TCP 变量和资源，确认报文中 SYN=1， ACK（确认位字段） =1,ack =x+1，seq=y;
step3：客户机接收到服务器的确认连接报文后，同样返回确认报文，且为该 TCP连接分配 TCP 变量和资源，确认报文中， ACK=1， ack=y+1， seq=x+1。

Q：虚电路和数据报的区别

A：
虚电路和数据报都是分组交换技术。

①数据报是无连接的数据交换，而虚电路是面向连接的数据交换；

②数据报的分组都是通过独立的路由选择和转发，而同属于一条虚电路的分组按照同一路由转发；
③数据报不保证数据的可靠交付，虚电路可靠性由网络保证；
④数据报不保证分组的有序到达，虚电路保证分组的有序到达。

四、计算机组成原理

Q：为了实现重定位需要哪些硬件？

A：
最简单的方式是在系统中增设一个重定位寄存器，用来存放正在执行作业的内存地址，每次访问数据时，由硬件自动将相对地址和重定位寄存器中的起始地址相加，形成实际的物理地址。当然在分页式与分段式系统中，还有地址变换机构，以及块表等方式。

Q：指令和数据放在一起存储的，计算机是如何区分指令和数据的？

A：
方式一：通过不同时间段来区分指令和数据，即在取指令阶段（或取值微指令）取出的为指令，在执行指令阶段（或相应微程序）取出的即为数据。
方式二：通过地址来源区分，由 PC 提供存储单元地址的取出的是指令，由指令地址码部分提供存储单元地址的取出的是操作数。

五、数据库

Q：数据库系统和文件系统相比的优点

A：
1.安全性高， 可靠性高
2.查询检索速度快
3.可以保证数据的有效性完整性和约束检查
4.有效的解决并发操作的问题
5.编程简单，操作方便。

Q：ACID

A：
原子性（Atomicity）： 事务中所有操作是不可再分割的原子单位。事务中所有操作要么全部执行成功，要么全部执行失败。
一致性（Consistency）： 事务执行后，数据库状态与其它业务规则保持一致。如转账业务，无论事务执行成功与否，参与转账的两个账号余额之和应该是不变的。
隔离性（Isolation）：隔离性是指在并发操作中， 不同事务之间应该隔离开来，使每个并发中的事务不会相互干扰。
持久性（Durability）：一旦事务提交成功， 事务中所有的数据操作都必须被持久化到数据库中，即使提交事务后，数据库马上崩溃，在数据库重启时，也必须能保证通过某种机制恢复数据。

Q：数据库三范式

A：
1NF(Normal Form)： R 的所有属性都不能再分解为更基本的数据单位。
2NF： R 的所有非主属性都依赖于 R 的关键属性，所有列都依赖于任意一组候选关键字。
3NF：每一列都与任意候选关键字直接相关而不是间接相关， 没有传递依赖。BCNF： 3NF 基础上， 关系 R 只有一个单属性，或 R 的子集都是单属性，则 R 满足 BCNF。

Q：数据库表插入 100 个数据和 100 万个数据有何区别？

A：
100 数量级小，可以随意插入； 100 万数量级大，如果表里有索引，则索引更新代价很高，可以采取先删除索引再插入，插入完成后再建索引的策略。

Q：数据库数据可以无限插入吗？

A：
可以。大小受到主机内存的制约。数据量大时要先删索引。减少提交次数，即减少 IO 次数

Q：group by having， having 和 where 的区别？

A：
WHERE 子句用来筛选 FROM 子句中指定的操作所产生的行。
GROUP BY 子句用来分组 WHERE 子句的输出。
HAVING 子句用来从分组的结果中筛选行。
在查询过程中聚合语句(sum,min,max,avg,count)要比 having 子句优先执行.而where 子 句在查询过程中执行优先级别优先于聚合语(sum,min,max,avg,count)。

六、程序设计基础

Q：重载和重写的区别

A：
方法的重载和重写都是实现多态的方式，区别在于前者实现的是编译时的多态性，而后者实现的是运行时的多态性。
重载发生在一个类中， 同名的方法如果有不同的参数列表（参数类型不同、参数个数不同或者二者都不同）则视为重载；重写发生在子类与父类之间，重写要求子类被重写方法与父类被重写方法有相同的参数列表，有兼容的返回类型，比父类被重写方法更好访问，不能比父类被重写方法声明更多的异常（里氏代换原则）。重载对返回类型没有特殊的要求，不能根据返回类型进行区分。

Q：两个栈模仿一个队列？

A：
进队：入 A 栈。
出队：若 B 栈不为空，则 B 栈全部出栈；否则将 A 栈中数据全部入 B 栈，再依次出 B 栈。

Q：两个队列模仿一个栈？

A：
入栈：入 A 队
出栈：将 A 队除队尾元素全部转移到 B 队，出 A 队， 算法思想就是两个队列倒来倒去，只留一个元素时出栈。

Q：如何判断链表是否有环？

A：
设置快慢指针，快指针每次前进两步，当两指针重合则有环，快指针为 null则无环。

Q：如何判断有环链表环的入口？

A：
1、将遍历过的结点都入 set，如果当前结点在 set 里有，则此结点即为入口。
2、快慢指针重合后，重置 fast 指针，此时 fast 每次走一步，再次重合结点即为入口。

Q：最长公共子序列求解(LCS)？

A：

DP。由最长公共子序列问题的最优子结构性质可知，要找出 X=<x1, x2, ...,xm>和 Y=<y1, y2, ..., yn>的最长公共子序列，可按以下方式递归地进行： 当xm=yn 时，找出 Xm-1 和 Yn-1 的最长公共子序列，然后在其尾部加上 xm(=yn)即可得 X 和 Y 的一个最长公共子序列。当 xm≠yn 时，必须解两个子问题，即找出Xm-1 和 Y 的一个最长公共子序列及 X 和 Yn-1 的一个最长公共子序列。这两个公共子序列中较长者即为 X 和 Y 的一个最长公共子序列.

Q：链表能否使用二分查找？

A：

可以。先将链表排序，将各个结点的值记入数组，再二分查找。

Q：给定一颗二叉树的头结点，和这颗二叉树中 2 个节点 n1 和 n2，求这两个节点的最近公共祖先

A：

后序遍历方法

```
public class Solution {
    public TreeNode lowes tCommonAncestor(TreeNode root, TreeNode
p, TreeNode q) {
        if(root == null|I root == p|| root == q)
            return root;
        TreeNode left = lowestCommonAncestor(root.left, p, q);
        TreeNode right = lowestCommonAncestor(root.right, p, q);
        if(left != null && right != null) return root;
        return left != null ? left : right ;
}
```

Q：栈应用括号匹配？

A：

左括号入栈，右括号出栈进行匹配，栈空仍未匹配到则失败

Q：汉诺塔问题？

A：

```
void Hanoi(char src, char des, char via, int n) {
    Hanoi(src, via, des, n - 1);
    Move(src, des, n); //把第n个盘子直接从src移动到des
    Hanoi(via,des, src, n - 1);
```

Q：二叉树删除节点？

A：
1.被删除的节点是叶子节点，这时候只要把这个节点删除，再把指向这个节点的父节点指针置为空就行。
2.被删除的节点有左子树，或者有右子树，而且只有其中一个，那么只要把当前删除节点的父节点指向被删除节点的左子树或者右子树就行。
3.被删除的节点既有左子树而且又有右子树，这时候需要把左子树的最右边的节点（最大前驱结点）或者右子树最左边的节点（最小后驱结点）提到被删除节点的位置。

Q：三种传参方式？

A：
值传递:传递的是实参的一个拷贝，修改形参不会改变实参值。
地址传递:
引用传递:传递的是实参的一个别名，修改形参会导致改变实参。被调用函数的形参只有在被调用时才会临时分配存储单元，一旦调用结束则释放内存。

七、其他前沿知识

Q：人工智能的三种形态

A：
弱人工智能 (Artificial Narrow Intelligence, ANI) 是擅长与单个方面的人工智能，比如有能战胜象棋世界冠军的人工智能，但是它只会下象棋，你要问它怎样更好地在硬盘上存储数据，它就不知道怎么回答你了；
强人工智能 (Artificial General Intelligence, AGI)，是人类级别的人工智能， 强人工智能是

指在各方面都能和人类比肩的人工智能，人类能干的脑力活它都能干。创造强人工智能比创造弱人工智能要难得多，我们现在还做不到。Linda Gottfredson 教授把智能定义为“ 一种宽泛的心理能力，能够进行思考、计划、解决问题、抽象思维、理解复杂理念，快速学习和从经验中学习等操作” 。强人工智能在进行这些操作时，应该和人类一样得心应手；
超人工智能 (Artificial Super Intelligence, ASI)，牛津哲学家，知名人工智能思想家 Nick Bostrom 把超级智能定义为“ 在几乎所有领域都比最聪明的人类大脑都聪明很多，包括科技创新、通识和社交技能” 。超人工智能可以是各方面都比人类强一点，也可以是各方面都比人类强万亿倍，超人工智能也正是为什么人工智能这个话题这么火热的缘故

Q：云计算？

A：
云计算是一种按使用量付费的模式，这种模式提供可用的、便捷的、按需的网络访问， 进入可配置的计算资源共享池，这些资源能够被快速提供，只需投入很少的管理工作，或与服务供应商进行很少的交互。

Q：大数据的特点？

A：
1.Volume:数据量巨大
体量大是大数据区分于传统数据最显著的特征。一般关系型数据库处理的数据量在 TB 级，大数据所处理的数据量通常在 PB 级以上。
2.Variety:数据类型多
大数据所处理的计算机数据类型早已不是单一的文本形式或者结构化数据库中的表，它包括订单、日志、BLOG、微博、音频、视频等各种复杂结构的数据。
3.Velocity:数据流动快
速度是大数据区分于传统数据的重要特征。在海量数据面前，需要实时分析获取需要的信息，处理数据的效率就是组织的生命。
4.Value:数据潜在价值大
在研究和技术开发领域，上述三个特征已经足够表征大数据的特点。但在商业应用领域，第四个特征就显得非常关键！投入如此巨大的研究和技术开发的努力，就是因为大家都洞察到了大数据的潜在巨大价值。如何通过强大的机器学习和高级分析更迅速地完成数据的价值“提纯”，挖掘出大数据的潜在价值，这是目前大数据应用背景下苛待解决的难题。

Q：大数据发展的瓶颈？

A：
没有成熟的方法采集和处理大数据。
数据涉及到隐私， 法律法规还没有完善。
大量不同类别的数据不知道怎么存储。
数据的独占性：有价值的数据别人不一定会分享。

计算机考研(408)复试准备集合

微机原理

1、计算机的主要硬件指标？

机器字长：cpu一次能处理的数据位数
运算速度：（单位：MIPS 每秒执行百万条指令）
存储容量：存放二进制信息的总位数

2、虚拟存储器

指采用一定的方法讲一定的外村容量模拟成内存，同时对程序进出内存的方式进行管理，从而得到一个比实际内存大得多的内存空间，使得存储系统既具有相当于外存的容量，又有接近于主存的速度。

3、总线复用

地址线与数据线复用

4、I/O接口的功能

实现设备的选择、实现数据缓冲达到速度匹配、实现数据格式转换

5、程序效率

指程序的执行速度和占用的存储空间

6、程序存储

指将程序和数据事先存放在计算机内部的存储器中，计算机在工作时自动取出并执行，不需要人工干预。

7、高速缓冲存储器Cache的作用

提高cpu访问主存的速度，极大缓和cpu和主存之间速度不匹配的矛盾。

8、端口是指可由cpu读或写的寄存器。

（数据端口，状态端口，控制端口）

9、寄存器间接寻址用到的寄存器：

BP,BX,DI,SI

10、总线：连接计算机各组成部件的公共数据通路。分为片级总线（连接cpu，存储器和I/O接口），片内总线（连接cpu内部的各个部件），系统总线（连接外部设备）。

三总线：

数据总线：用来传送数据
地址总线：同来传送存储器或外设端口地址
控制总线：用于传送各种控制信号

11、8259A的结构：

IRR（中断请求寄存器）：记录IR线上的各级中断请求
ISR(中断服务寄存器)：保存当前正在响应的中断
IMR（中断屏蔽寄存器）：对IRR中的相应中断源进行屏蔽

12、8086内部结构？

总线接口部件：完成cpu与存储器和外设的数据传传送
执行部件：取指令，指令译码，算术运算

13、指令队列的作用？

在执行指令的同时从内存中取出下一条指令，将取出的指令放在指令队列中，较少了cpu的等待时间，从而提高cpu效率。

14、怎样用16位寄存器实现对20位地址的寻址？

引入分段管理机制，先将16位段基地址左移4位，再加上16位偏移地址，构成20位物理地址。

15、中断是什么？

Cpu暂时停止当前工作，转去处理对应的中断服务程序，待事件处理完毕后，回到原先工作地方继续工作的过程。

16、响应中断的过程？

关中断，保护断点，获取中断类型号。

17、中断过程？

中断请求，中断判优，中断响应，中断服务，中断返回

18、指令执行过程

取指令、分析指令、读取操作数、执行指令、存放结果

19、吞吐量

指系统在单位时间内处理请求的数量

20、计算机的工作过程

①把程序和数据装入内存
②将源程序转换成可执行文件
③从可执行文件的首地址开始逐条执行指令

21、总线周期

利用总线完成一次读/写所休要的时间

22、A/D转换器的分辨率

数字输出量每增加一位所对应的模拟输入量的变化数值

23、如何实现地址线与数据线分时复用？

8086有16个地址/数据复用引脚，作为复用引脚，在总线周期的T1状态用来输出地址信息，在T2-T3状态，对于读周期来说处于浮空状态，对于写周期来说则传输数据

24、8086/8088最小/最大模式的区别？

最小模式是单处理器模式，只有一片微处理器；
最大模式可有多个微处理器，一个主处理器，多个协处理器；
最大模式相比于最小模式增设了总线控制器。

25、cpu与外设之间的数据传送方式

程序控制：分为程序查询和无条件传送方式。程序查询是cpu与外设数据传送之前。都要先检查外设状态，只有当外设处于“就绪”状态才能传送；无条件传送是始终要求外设处于“就绪”状态。
中断方式：让外设在数据准备好后再通知cpu去执行IN/OUT操作
DMA方式：外设直接和存储器进行数据交换，cpu放弃总线控制权交给DMAC进行控制
通道方式

26、8253的6种工作模式种有4种计数模式，2种定时模式

27、8259A有哪些中断优先级方式？

普通全嵌套（只能嵌套高优先级的中断）
特殊全嵌套（可以嵌套同级或高优先级中断）
优先级自动循环（其实优先级固定）
优先级特殊循环（起始优先级可设定）
中断屏蔽

28、串行通信的异步方式和同步方式？

同步：每次传送一个数据块，数据块内部字符之间的间隔是固定的
异步：通信中两个字符间的发送时间间隔是不固定的

29、多级存储系统

“cache——主存”层次和“主存——辅存”层次
目的：为了解决存储系统大容量、高速度和低成本之间相互制约的矛盾

数据库

1、什么是数据库

数据库是长期存储在计算机中的有组织的，可共享的数据集合

2、概念模式

数据库中全部数据的整体逻辑结构的描述

3、事务

数据库系统一组操作序列，这些操作要嘛都成功执行，要嘛都不执行，是一个不可分割的工作单位。

4、索引的目的

可快速定位数据在磁盘中的位置，避免从头至尾查找，从而加快数据查询速度。

5、游标的作用

通过游标机制可对SELECT语句的查询结果进行按行按列处理。

6、相容矩阵

说明事务对某数据对象的加锁请求在什么情况下可获准或被拒绝。

7、数据模型的三要素

数据结构，数据操作，数据约束

8、DBMS的基本功能

独立性，安全性，完整性，并发控制，故障恢复

9、DBMS的数据独立性是如何是是实现的？

将数据结构和数据文件从应用程序中分离出来，交给DBMS管理。

10、事务的性质？

ACID：原子性，一致性，隔离性，持久性

11.UML图

UML（统一建模语言），分为结构图和行为图；况图，类图，对象图，组件图，部署图；状态图，活动图

12、并发执行可能引起的问题？

丢失更新，读脏数据，读值不可复现
解决办法：采用加锁机制实现并发控制

13、数据字典

是对系统中数据的详尽描述，它提供对数据库数据描述的集中管理。

14、数据模型的分类？

层次模型，网状模型，关系模型，面向对象模型

15、什么是死锁？产生死锁的原因？

死锁是指多个进程因争夺资源而造成的一种阻塞现象
事务双方持有彼此所需要的锁而未释放，造成循环等待。

16、解除死锁状态？

将一事务作为牺牲品，把它撤销并回退，解除它的所有封锁，并恢复到初始状态

17、关系数据模型的数据完整性规则有哪些？

实体完整性规则，参照完整性规则，用户自定义完整性规则

18、数据库恢复的几种方法？

定期对整个数据库进行复制和转储；建立日志文件

19、为什么视图被称为虚表？/什么是视图？

在建立视图时，在数据字典中只存储有关视图的定义，而不存放数据，数据仍存放在基表中。
视图是建立在基本表之上的虚拟表，它具有与基本表相同的功能，可以对视图进行操作，但是对视图的修改不会影响基本表。

20、文件系统与数据库系统相比较，缺陷表现在？

数据联系弱，数据冗余，数据不一致

21、关系模式存在的问题？/产生异常的原因？/如何解决？

插入异常，删除异常，冗余及更换异常。
将多种数据集于一个关系模式中，使得属性间产生复杂的依赖关系。
按照需要，依照不同等级的范式分解模式。

22、什么是范式？

范式是指关系模式的规范形式。

23、为什么要进行关系模式的分解？

将低级别范式向高级别范式转换，其目的是消除异常。

24、互斥并发控制协议？

当一个事务访问某个数据对象时，不允许其他事务更新该数据对象。

25、数据库发生错误如何处理？

扫描日志文件，找到故障事务，撤销该事务并回退到该事务之前的正常状态

26、x锁协议：一个事务对数据对象A加了x锁，就可以对A进行读取和更新，加锁期间其他事务对A不能加任何锁

（s，x）加锁协议：读操作时，必须先给要操作的数据对象加s锁；写操作时，必须先给要操作的数据对象加x锁。（s锁与x锁，x锁与x锁互斥）

27、活锁现象

（s，x）加锁协议可能会出现活锁现象，若一数据对象已被加了s锁，其他事务要申请对它的x锁，就需要等待。若不断有事务对它申请s锁，使得数据对象一直被s锁占用，x锁得申请一直得不到获准

28、活锁得解决办法

采用先来先服务原则，按照请求加锁的顺序对事务排队

29、多粒度锁

行锁，页锁，表锁

30、产生死锁的必要条件

互斥条件，请求和保持条件，不剥夺条件，环路等待条件

31、处理死锁的基本方法

预防死锁，避免死锁，检测死锁，解除死锁

软件工程

1、瀑布模型？

前一阶段工作结束才能进行下一阶段工作。核心思想是按工序将问题化简，将功能的实现与设计分开，便于分工协作。

2、软件工程三要素？

方法，工具，过程

3、螺旋模型？

结合快速原型和瀑布模型，不需要在刚开始就把所有事情都定义清楚，强调风险分析的特点。

4、软件工程步骤？

问题定义，可行性研究，需求分析，总体设计，详细设计，编码，单元测试

5、有哪些软件测试方法？

黑盒测试：仅根据I/O数据条件来设计测试样例
白盒测试：分析程序内部逻辑和执行路线来设计测试样例

6、死代码：永远不会被执行到的代码

7、内聚：一个模块内各元素结合的紧密程度

耦合：各个模块之间的互联程度

8、软件由数据、程序、文档组成

9、敏捷编程：敏捷开发以用户的需求为核心，采用迭代，循序渐进的方法进行软件开发

10、软件危机：软件开发和维护过程中遇到的一系列严重问题

11、软件生存周期：一个软件从定义，开发到使用和维护，直到最终被弃用的过程

12、影响软件维护的因素？

可理解性，可 测试性，可修改性

13、软件维护的目标

纠正在使用过程中暴露出的错误而进行的改正性维护

14、主要的软件工程方法

生命周期方法，面向对象方法

15、需求分析的任务

深入描述软件的功能，解决“做什么”的问题

16、结构化设计的要点

模块化、自顶向下、逐步求精

计算机网络

1、OSI体系结构TCP/IP五层模型的区别?

OSI模型（开放互联模型）：应，表，会，传，网，数，物
TCP/IP五层模型：应，传，网，数，物。将表示层和会话层与应用层合并

2、各层模型的最小传输单位？

应用层：报文 （HTTP,FTP,SMTP,DNS）
传输层：报文段 (TCP,UDP)
网络层：数据报 (IP，ICMP)
数据链路层：帧 (PPP,CSMA/CD)
物理层：比特
ICMP(因特网控制报文协议)：为提高IP数据报交付成功的几率，采用ICMP协议来让主机或路由器报告差错情况

3、ARP（地址解析协议）协议：用来解释IP地址所对应的MAC地址

4、DHCP（动态主机配置协议）协议：为新加入到计算机网络中的计算机自动分配IP地址

5、TCP的三次握手

①主机A向主机B发送SYN请求，请求建立连接
②主机B收到请求后向主机A发送SYNACK确认信号
③主机A收到SYNACK信号后再向主机B发送确认信号，准备传输数据

6、TCP，UDP的区别？

TCP是面向连接的可靠传输，面向字节流数据，传输慢
UDP是无连接的不可靠传输，面向报文数据，传输快

7、网络协议的三个核心要素：语法，语义，同步

8、TCP保证可靠传输的方法？

确认应答机制、超时重传机制、滑动窗口机制，流量控制机制，拥塞控制机制

9、主机间的通信方式/网络应用模型

客户——服务器（C/S）：客户是服务的请求方，服务器是服务的提供方
对等（p2p）：不区分客户服务器

10、电路交换：整个报文的比特流从源点直达终点

报文交换：将整个报文分组转发到相邻节点，通过查找转发表，转发到下一个节点
分组交换：将报文分组转发到相邻节点，通过查找转发表，转发到下一个节点

11、计算机网络的主要想能指标：带宽、时延

12、流量控制的常见方式？

停止——等待、滑动窗口，回退N步、选择重传

13、路由器的功能：路由选择、分组转发

14、DNS域名解析协议：实现域名与IP地址的相互转发

15、HTTP超文本传输协议：规定了在浏览器和服务器之间的请求和响应的格式和规则

16、浏览器输入地址后发生的事情？

①浏览器向DNS请求解析域名地址获取对应的IP地址
②浏览器与该服务器建立TCP连接
③浏览器向服务器发送HTTP请求
④服务器通过HTTP响应把文件发送给浏览器
⑤释放TCP连接
⑥浏览器解释文件

程序设计

1、C和C++还有java的区别？

C是面向过程的结构化编程语言，c++也支持面向过程编程，但更重要的是c++支持面向对象编程。C++具有封装，继承，多态特性
Java是纯面向对象的语言，也具有封装，继承，多态的特性，但是java不提供指针直接访问内存，更加安全。Java提供自动内存管理机制，不需要手动释放无用内存。Java不支持多重继承，也不支持操作符重载。

2、java和python的区别？

两者都是解释性语言。但是，
Java是静态语言，定义变量时必须声明数据类型。
Python是动态语言，定义变量不用声明数据类型。

3、指针和引用的区别？

指针是一个存储地址的变量，引用是原变量的一个别名。
指针可以为空，引用不能空；
指针可以重新赋值，而引用只能初始化一次

4、const的作用？

通常用来定义常量，被const修饰符修饰的变量不能被修改

5、static的作用？

定义局部静态变量：只初始化一次，在项目启动时就分配内存
定义全局静态变量/函数：只能在本文件中使用
定义类的静态成员变量/函数：不依赖于类对象的存在而存在

6、java创建一个线程类的方式？

继承Thread类，实现Runnable接口

7、线程的几种状态？

新生状态，可运行状态，阻塞状态，死亡状态

8、解释一下多态性？

指一段程序能够处理多种类型对象的能力

9、内联函数：将一些功能简单，规模较小又使用频繁的函数设计为内联函数，提高程序执行效率。（编译时将函数体嵌入在每一个调用处）

10、构造函数：初始化对象，在对象被创建时自动被调用。它的函数名与类名相同，没有返回值。

11、析构函数：完成对象被删除前的一些清理工作，在对象生存期即将结束前时被自动调用

12、友元函数：在类中用friend修饰的非成员函数，它可以直接访问类的私有成员。实现了不同类之间的数据共享。友元关系是单向的，不能传递也不能继承

13、函数重载：具有相同的函数名的多个函数，其形参类型和个数不同，编译器根据实参和形参进行最佳匹配，自动确定调用哪个函数，从而实现相同函数名完成不同的功能。

14、虚函数:在基类中声明为virtual并在一个或多个派生类中被重新定义的成员函数。它的作用是实现多态性。

15、纯虚函数:一个在基类中声明的虚函数，没有函数体，只能由派生类实现。

16、重写和重载的区别？

重写发生在子类和父类间，是同名函数有相同的参数列表，但是函数体不同
重载是同名函数的参数列表不同

17、const和define的区别？

const修饰变量，系统会分配存储单元并存放在静态存储区，进行类型检查
define本质是宏替换，不会分配存储单元且不进行类型检查

18、面向对象编程和面向过程编程的区别？

面向过程是根据解决问题的步骤编写函数然后一个一个依次调用
面向对象是把构成问题事务分解成各个对象。对象是为了描述某个事务在整个解决问题的步骤中的行为。

19、抽象数据类型

是把数据对象，数据对象之间的关系，数据对象的基本操作封装在一起的一种表达方式

20、栈和队列的区别？

1.栈先进后出，队列先进先出
2.对插入和删除操作的限定不同。栈是限定只能在一段进行插入和删除的线性表，队列是限定在一段进行插入，在另一端进行删除的线性表

21、栈和堆的区别？

1.栈是只允许在一端进行插入和删除的线性表，而堆是一种特殊的完全二叉树
2.栈由操作系统自动分配和释放，无需手动控制；堆的申请和释放由程序员控制

22、c中的malloc，free和c++中的new，delete有什么区别？

new，delete是操作符，可以重载；
malloc，free是函数，可以覆盖；
new可以调动对象的构造函数，delete调用相应的析构函数；
malloc和free仅仅是分配内存和释放内存

23、类与结构体的区别？

结构体存储在栈中，类的实例化可以存储在栈中，也可以存储在堆中
结构体的执行效率要比类高
结构体不可以继承，类可以继承

24、函数模板是什么？

指使用了模板计数定义了参数化类型的非成员函数，使程序能使用不同的参数类型调用相同的函数

25、泛型？

是把类型明确的工作推迟到创建或调用方法的时候才去明确的特殊类型

操作系统

操作系统是计算机资源的管理者

1、操作系统的特点：共享，并发，虚拟，异步

2、操作系统的主要功能：处理机管理，存储器管理，设备管理，文件管理

3、进程和线程的区别？

进程是资源分配的最小单位，线程是cpu调度的最小单位。
同一进程的线程共享本进程的资源和地址空间，而进程间则是资源独立，地址空间独立的。

4、存储器的层次结构有哪些？

主存，外存，高速缓存，寄存器

5、实时操作系统和分时操作系统的区别？

分时操作系统按照相等的时间片调度进程轮流运行，无法实时响应外部异步事件；实时操作系统能够在限定的时间内执行完所规定的功能，并在限定的时间内对外部异步事件做出响应。

6、os的组成？

驱动程序，内核，支承库，外围

7、什么是内存？

内存是与cpu直接交换数据的内部存储器，它用于暂时存放cpu中的运算数据

8、分页和分段的区别？

页是信息的物理单位，分页是系统的需要，为了提高内存利用率
段是信息的逻辑单位，分段是为了更好的满足用户的需要

9、什么是物理内存？什么是虚拟内存？两者的关系？

物理内存即内存条，当打开程序时，系统会将程序自动加载到物理内存上
虚拟内存可代替物理内存行使存储的功能，但是无法替代物理内存行使加载程序的功能。
关系：当运行的程序过多，物理内存不够时，系统会把一部分硬盘空间当作内存条使用，这就变成了虚拟内存

10、硬中断和软中断的区别？

硬中断：是由外设引发的，中断号由中断控制器提供，硬中断是可屏蔽的
软中断：是执行中断指令产生，中断号由指令直接指出，无需中断控制器，软中断不可屏蔽

11、中断和异常的区别？

中断是由系统中某事件引起的，该事件与现行指令无关
异常是由于执行现行指令引起的

12、线程的几种状态：新生状态，可运行状态，阻塞状态，死亡状态

13、进程的调度算法？

先来先服务、最短作业优先、优先级调度、时间片轮转

14、为什么需要进程同步？

当多个进程同时访问一个共享资源时，可能会发生冲突，因此需要进程同步，多个进程按顺序访问资源

15、程序的装入方式有哪些？

绝对装入，可重定位装入，动态运行装入

16、原子操作：不会被线程调度机制打断的操作

17、内存连续分配管理方式有哪些？

单一连续分配，固定分区分配，动态分区分配

18、饥饿：等待时间给进程推进和响应带来明显影响时成为饥饿

19、银行家算法:主要思想是为了避免系统进入不安全状态，在每次进行资源分配时，先检查系统是否有足够多的资源满足要求，若有则分配，若无则拒绝