什么是数据结构？

- 数据结构是相互之间存在一种或多种特定关系的数据元素的集合
- 数据结构的三要素：
- 逻辑结构，存储结构，数据的运算
- 逻辑结构：
- 线性结构和非线性结构
- 存储结构：
- 顺序，链式，索引，散列存储

聊一聊常见的排序算法？

- **快速排序**:

- 快速排序是一种基于分治思想的交换类排序算法。首先在待排序的序列中随机挑选一个元素作为枢轴元素，通过头尾定义两个指针，不断的将比枢轴元素小的元素换到左边，将比枢轴元素大的元素放到枢轴右边，一趟排序完成后将枢轴元素左边和右边的序列划分为两个子序列，左右两边分别继续上述迭代，直到最终子序列只有一个元素或为空，就完成了排序。

- **希尔排序**:

- 希尔排序是一种优化后的插入类的排序算法。直接插入算法在元素较少或基本有序的情况下表现很好，希尔排序就是利用这一特性来提高排序效率。首先，选择一个步长n，将序列中所有间隔n步的元素各划分为一个子序列，在子序列内部进行直接插入排序，此时子序列中元素较少，插入排序效率高，通过不断减少n来进行迭代。最终当n迭代为1时，序列中元素基本有序，因此插入排序的效率也很高，最终达到优化的目的。

- **堆排序**:

- 堆排序是一种优化后的选择类的排序算法。堆排序通过维护一个小根堆或大根堆进行最大或最小元素的选择。在移除到最大或最小元素后，根据堆的结构特性，通过调整堆结构可以在O(logN)时间内重新变为大根堆或小根堆，整体时间复杂度时O(NlogN)

介绍一下归并排序？

- **内部归并排序算法**：
  - 归并排序基于分治与归并的思路来对序列进行排序。首先需要引入一个辅助数组用来存储归并后的序列。通过分治最终将序列分割为单个元素，此时单个元素必然有序，然后通过不断的将相邻的有序序列归并为一个有序元素，最终实现真个序列的排序。
- **外部归并排序算法**：
  - 外部归并排序，一般是多路归并，基本思想与二路归并一致。如果通过置换选择排序，生成长度不同的初始归并段，就需要根据多叉哈夫曼树的思想来求一个最小的带权路径长度，来生成一颗最佳归并树。其中在k叉树内部一般使用败者树来加速多个序列首元素的大小比较。

简述如何构建哈夫曼树？

- **哈夫曼树**是含有n个带权叶节点的二叉树中，带权路径长度最小的二叉树，也叫最优二叉树。构建一棵哈夫曼树主要有以下步骤：
- 第一步，
- 将每个节点按照权值从小到大的顺序整理出来。
- 第二步，
- 从节点中挑选两个权值最小的节点，将他们权值相加作为一个新的节点加入节点集合，并移除两个相应的子节点。
- 第三步，
- 不断重复第一步和第二步，直到集合中只剩下一个节点，这个节点就是哈夫曼树的根节点，至此哈夫曼树的根节点构建完成。

贪心算法和动态规划的区别？

- 贪心算法是一种”短视“的算法，通过局部最优选择来求解问题；动态规划则更加注重全局最优，通过求解子问题并利用子问题的解来构建全局最优解。

- **求解思路**：

- 贪心算法是一种自顶向下的策略，每一步仅考虑局部最优，不考虑整体最优，在某些特定问题下有效，如找零钱问题中可以通过局部最优达到全局最优。而动态规划使用自底向上的求解思路，先将大问题分解为小问题的组合，并在每一次迭代过程中保存子问题的解，避免重复计算，最终逐步构造出大问题的解，也被称为”记忆化搜索“。

- **时间复杂度**：

- 贪心算法时间复杂度较低，一般是线性复杂度，而动态规划的时间复杂度一般较高，通常是多项式复杂度

介绍一下Dijkstra算法?

- Dijkstra算法是一种解决单源最短路径问题的经典算法。

- **基本原理**:
  - 它以起始顶点为中心，逐步向外扩展，通过不断更新起始点到各个顶点的最短距离来找到其他所有顶点的最短路径。

- **执行过程**:
  - 第一步，定义两个队列S和U,S存放已找到最短路径节点,U中存放待寻找最短路径的节点以及到他们到起点的距离。第二步，每次迭代更新U中的节点到起点的距离，并把距离最短的点从U中移除加入到S中。直到所有节点都加入到S中算法完成。

- **应用场景**:
  - 主要用在无负权边的图中，常见的应用场景有物流配送寻找一条最短配送路径，地图导航中选择最短路径，以及路由选择算法中选择最佳传输路径。

介绍一下最小生成树算法?

- 常见的最小生成树算法有Prim算法和Kruskal算法，他们都是基于贪心策略。

Prim算法:

- Prim算法是根据顶点集生成树的算法，执行步骤如下：定义一个树T，任取一个顶点加入T中，然后寻找T集合中所有顶点相邻的边，寻找与T中节点距离最近的节点加入T中，直到所有的顶点都加入到T中算法结束。Prim算法时间复杂度为O(|V|^2)，仅与顶点数相关，因此适用于边稠密的图。

Kruskal算法:

- Kruskal算法是基于边集生成树的算法，执行步骤如下：定义树T，按照边的权值从小到大的顺序，以及选取不构成环的最小的一条边加入当前的T中，直到所有顶点都加入树T中。Kruskal算法时间复杂度为O(|E|log|E|),仅与边的个数相关，适用于边稀疏的图。

介绍一下B树和B+树？

- B树和B+树都是优化数据查找的效率的数据结构。

- **B树**：

- 也称多路平衡查找树，m阶b树中每个节点最多有m棵子树。除根结点外每个节点最多有m/2向上取整棵子树，每个节点的关键字个数为子树个数减一。在B树查找的过程中，查找到中间节点中的关键字位置即可拿到记录信息，停止向下查询，不支持顺序查找。

- **B+树**：

- 应用在数据库中的B树的变形树，其中每个关键字都对应一颗子树。在B+树查询中，无论成功与否都需要查找到叶节点，关键字记录信息都存储在叶节点中，并且叶节点之间有指针相互连接，支持顺序查找。

介绍一下常见的哈希冲突的解决办法？

- 哈希冲突是指不同的键通过哈希映射后指到了同一个地址而引发的冲突。

- **开放定址法**：
  - 通过线性探测，平方探测以及再散列法等方式再寻找一个空地址来对应后加入的键值，注意在开放定址法中不可以随便物理删除表中的已有元素，只能通过标记进行逻辑删除。

- **拉链法**：
  - 将关键字通过函数映射到一个链表中，同义词在映射中会不断的添加到对应链表的后面，适用于经常发生插入和删除的情况。

介绍一下KMP算法？

- KMP算法是字符串的模式匹配中的一种优化算法，求的是子串在主串中的位置。
- 在朴素的模式匹配算法中，通过遍历子串在主串中所有可能的情况来求出子串在主串中的位置，时间复杂度通常是O(M*N)。
- KMP算法针对子串自身的特征，通过定义一个Next数组，使得在子串发生失配时，不必都回溯到子串的第一个字符来重新与主串匹配，从而提高了子串的检索速度，KMP的时间复杂度通常为O(M+N)。

介绍一下BFS与DFS?

- BFS与DFS是图的两种不同的遍历方式。

- **BFS**:

- 广度优先搜索类似于二叉树的层序遍历，基本思想如下：维护一个辅助队列和访问标记数组，第一步将起始节点入队作为第一层，第二步记录当前队列的长度，遍历当前层次中的所有元素。每遍历一个元素，就从队列中弹出，并将将其所有未访问的相邻节点加入队列中。重复上述第二步中的步骤，直到队列中元素为空就完成了广度优先搜索。

- **DFS**:

- 深度优先搜索类似于二叉树的先序遍历，基本思想如下：维护一个访问标记数组，首先从起始点出发，依次遍历当前还未访问过的相邻节点，然后添加访问标记，重复上述循环，如果所有相邻节点都被访问，则回溯到上一层节点继续搜索，直到所有的节点都被访问。

介绍一下常见的二叉查找树？

- 构建二叉查找树目的是为提高数据的查找，插入和删除关键字的速度

- **普通二叉查找树**：

- 对于树中的任意一个节点，其左子树中所有节点的键值都小于该节点的键值；其右子树中所有节点的键值都大于该节点的键值。

- **平衡二叉查找树（AVL 树）**：

- AVL 树要求任意节点的左子树和右子树高度差的绝对值不超过1。解决了序列有序时普通二叉查找树会退化为单链表的问题。在插入或删除节点后，通过左旋、右旋、先左旋后右旋、先右旋后左旋等旋转操作保持平衡。

- **红黑树**：

- 也是一种自平衡的二叉查找树，它通过为节点添加颜色（红色或黑色）来维持树的平衡。红黑树在插入和删除操作后，通过变色和旋转操作保持平衡。相比于 AVL 树，红黑树的平衡要求相对宽松，在频繁插入和删除操作场景下性能表现更好 。

操作系统的主要特征？

- 操作系统的基本特征包括并发，共享，虚拟和异步。

- **并发**:
- 并发是指多个时间在同一时间间隔内发生。并行是指在同一时刻完成多种工作。

- **共享**:
- 共享是指系统中的资源可供内存中多个并发执行的进程共同使用。

- **虚拟**:
- 虚拟是指将一个物理上的实体变为若干逻辑上的对应物。操作系统的虚拟技术主要有：时分复用，如虚拟处理器；空分复用，如虚拟存储器

- **异步**:
- 多道程序以不可预知的速度向前推进，这就是进程的异步性。

说一下系统调用的处理过程？

- 系统调用是用户程序在用户态下请求操作系统资源的一组接口，对资源的分配由操作系统一控制。
- **系统调用的过程**:
- **第一步**:
- 用户程序给出系统调用所需的参数。操作系统会执行trap指令，将系统状态转为内核态，并保护CPU中断现场，包括程序计数器PC以及程序状态字PSW等寄存器压入堆栈保存。
- **第二步**:
- 分析系统调用的类型，转入相应的系统调用子程序。
- **第三步**:
- 系统调用处理子程序执行完毕后，恢复被中断的CPU现场并返回被中断的进程继续执行。

大内核和微内核的优缺点？

- 大内核和维内核是操作系统内核的两种不同架构，它们各有优劣。

- **大内核**：

- 大内核是指将系统的主要功能模块作为一个紧密联系的整体都运行在内核态，优点主要是性能较高，系统进程之间的通信无需频繁切换上下文，能有效提升系统性能。缺点则是可扩展性差，整体结构复杂升级更新系统时可能会有意想不到的问题出现；可靠性低，当系统中的一个功能模块出现错误时会导致整个内核崩溃，影响系统稳定性。目前主流的操作系统如Window，Android，Linux都是基于大内核。

- **微内核**：

- 微内核是指将内核中最基本的功能保留在内核，而将一些不需要在内核态运行的功能迁移到用户态，微内核的优点主要有：可扩展性好，可靠性高，可移植性强；主要缺点是性能开销大。

操作系统的引导过程？

- 在启动一个操作系统时有以下步骤：
- 第一步，
- 激活CPU，读取并执行主板上的BIOS指令；
- 第二步，
- 硬件自检，检查各个关键设备是否故障；
- 第三步，
- 加载带有操作系统的硬盘，加载硬盘中的主引导记录；
- 第四步，
- 扫描硬盘分区表，加载分区引导记录；
- 第五步，
- 加载操作系统。

什么是进程和线程，并聊一聊它们的区别？

- 进程和线程都是操作系统中用于管理程序执行的概念。

- **进程**：

- 进程是程序在操作系统中的一次执行过程，是系统进行资源分配和调度的基本单位。每个进程都有自己独立的地址空间，内存，打开文件等资源。

- **线程**：

- 线程是进程中的一个执行单元，是程序执行的最小单位。一个进程可以包含多个线程，这些线程共享进程的资源，如地址空间等。

- **区别**：

- 进程有独立的地址空间，线程没有；进程发生切换时开销较大，线程切换开销相对较小；进程具有较高的独立性，线程独立性低。

进程间通信的方式有哪些?

- 进程通信是指进程间的信息交换。
- **管道**:
- 管道是一种特殊的共享文件，分为无名管道和命名管道。无名管道用于有亲缘关系的进程通信，命名管道可以用允许无亲缘关系的进程间通信。
- **共享内存**:
- 在通信的进程之间存在一块特殊的的公共内存空间，通过对这块空间的读写来实现进程通信。对共享空间的读写一般需要同步互斥工具来进行读写控制。
- **消息传递**:
- 进程间的数据交换以格式化的消息（Message）为单位，通过操作系统提供的发送消息接受消息两个原语进行数据交换。消息传递又分为直接通信方式和间接通讯方式。
- **信号量**:
- 信号量是一种低级通信方式，用来控制多个进程对共享资源的访问，实现进程间的互斥与同步。
- **信号**:
- 用于通知接收进程某个事件的发生。常见的SIGKILL表示强制终止进程，SIGFPE表示浮点数运算错误

常见的进程调度方法?

- 进程调度是指在就绪队列中按照公平高效的原则选择一个进程上处理机运行的过程。
- **先来先服务**：建立一个排队队列，所有就绪队列按照请求顺序以此入队，永远选择队首的进程上处理机运行。特点是对长进程有利对短进程不利，适用于CPU繁忙型作业。
- **短作业优先**：
- 从后备就绪队列中选择一个估计运行时间最短的进程，上处理机运行。该算法的特点是对短进程有利而对长进程不利，可能会导致长进程饿死。
- **优先级调度**：
- 按照优先级的顺序依次调度，可以优先处理用户最紧急的任务，提高用户体验。
- **时间片轮转**：
- 适用于分时系统，每个进程仅允许执行一个时间片。
- **高响应比优先**：
- 综合考虑了作业的等待时间和估计执行时间来确定进程执行的优先级。
- **多级反馈队列调度算法**：
- 结合了时间片轮转以及优先级调度的思想，设置多个就绪队列，每个队列赋予不同时间片，是一种十分均衡的进程调度算法。

死锁产生的必要条件以及处理策略？

- 死锁是指多个进程因竞争有限的资源而造成的一种僵局，使各个进程都被阻塞，无法推进的状态

- **死锁产生必要条件**：

- 互斥条件，即每个资源只能被一个进程占有；不可剥夺，即进程获得的资源在它使用完之前不能被其他进程强行夺走；请求并保持，即进程保持至少一个资源的同时，又提出了新的请求；循环等待，存在一种进程资源的循环等待链，链中进程已获得的资源被链中的下一个进程所请求。

- **死锁处理策略**：

- 死锁预防，设置某些限制条件，破坏产生死锁的必要条件。死锁的避免，在资源的动态分配中，通过某种方式防止系统进入不安全状态，常见的有银行家算法。银行家算法简单来说就是当前系统必须找到一个进程的执行顺序，使得所有进程都能执行完毕不发生死锁。当分配某个资源后导致系统进入不安全状态时就拒绝分配。死锁检测和解除，死锁检测资源分配图来确定，当资源分配图不能完全被简化时就判定发生了死锁；常见的死锁解除的方法主要有，资源剥夺，撤销进程，进程回退等

说说虚拟内存中的分布？

- 当一个程序调入内存时运行时就会有一个进程自己独有的进程地址空间，按照地址从低到高主要有以下几个部分组成：

- **代码段**：即程序的二进制代码，代码段是只读的。
- **数据段**：程序运行时加工处理的对象，包括全局变量和静态变量。
- **堆区**：用来存放动态分配的变量，通过调用malloc函数动态地向高地址增长。
- **栈区**：存储局部变量，函数参数值，栈地址从高到低增长。
- **系统内核区**：主要实现操作系统内核的主要功能，内存管理，进程管理等。

简述一下虚拟内存，为什么要用虚拟内存?

- 虚拟内存可以使得系统能够利用硬盘空间来扩充物理内存的容量，以满足程序运行对内存的需求。

- **解决问题**:

- 早期内存分配为内存空间一次性全部分配，其中部分内存进程运行过程中不经常使用会造成的资源浪费。利用虚拟内存技术允许作业分多次调入内存，基于局部性原理，大大提高了内存空间利用率。

- **优点**:

- 扩大地址空间，允许比实际物理内存更大的程序的编写和运行。方便实现内存的保护和共享，将连续的逻辑地址映射到离散的物理地址减少了内存碎片的产生。

- **缺点**:

- 需要构建页表段表等额外的数据结构，占用内存空间；虚拟地址到物理地址的转换增加了程序执行时间。

简述从代码到可执行程序？

- 代码到程序的过程主要由编译器完成，主要有以下步骤：
- **预编译**：
- 预编译主要将代码中的宏定义展开，头文件导入，以及注释的过滤操作。
- **编译**：
- 编译过程主要包括词法分析，语法分析，语义分析，汇编代码生成等。
- **汇编**：
- 将汇编代码转换为机器可以执行的指令。
- **链接**：
- 将不同的源文件产生的目标文件进行链接，从而形成一个可执行程序，链接又分为静态链接和动态链接。

说一下常见的内存分配方式？

- 常见的内存分配方式可以分为两大类：连续内存分配和离散内存分配

- **连续内存分配**：

- 连续分配又分为固定分区分配和动态分区分配，其中固定分区分配可能会存在内部碎片，而动态分区分配根据进程实际需要来分配内存，无内部碎片但可能会产生外部碎片。常见的动态分区分配算法有：首次适应，循环首次适应，最佳适应，最坏适应等。

- **离散内存分配**：

- 常见的离散内存分配包括基本分页存储和基本分段存储管理。基本分页存储管理，通过页表机制，将连续的逻辑地址映射到离散的物理内存块上，解决了大量内存碎片的问题。分页操作完全由硬件完成，对用户透明。基本分段存储管理，满足了程序员编码过程中的信息保护和共享，动态增长以及动态链接等需求

说一下常见的页面置换算法?

- 页面置换算法主要应用在当一个进程分配的物理内存块用完之后，选择哪些页面进行换入换出，以便尽可能的减少进程执行过程的缺页率。

- **最佳置换算法**:
- 淘汰以后永远不使用的页面，以便获得尽可能低的缺页率。由于实际执行过程中无法预测进程后续访问的页面，因此该算法无法实现，但可以评价其他算法。

- **先进先出置换算法**：淘汰最早进入内存的页面。

- **LRU置换算法**:
- LRU算法选择淘汰最近最长时间未使用的页面。该算法为每个页面设置一个访问字段，用来记录上次访问以来的时间，淘汰所有页面中等待时间最长的页面，性能高开销大。

- **CLOCK置换算法**:
- 为每个进程添加一个访问位和修改位，使所有进程构成一个环形的等待队列，简单CLOCK算法优先淘汰最近未使用的页面，而改进后的CLOCK算法优先淘汰未访问未修改的内存块。

说一下常见的文件物理结构？

- 文件的物理结构是研究文件数据在物理存储设备上是如何分布和组织的。主要有三种分配方式：

- **连续分配**：

- 与数组在内存中分配方式类似，给每个文件在磁盘中找出一块连续的物理地址空间。优点是支持顺序和随机访问，顺序访问速度快，缺点是可能会产生很多外部碎片，并且无法满足文件动态增长的需求，插入删除操作较为复杂。

- **链接分配**：

- 与链表在内存中的分配方式类似，主要分为隐式链接和显式链接。优点在于消除了外部碎片，提高磁盘利用率，可以动态分配盘块，插入删除和修改都比较方便。显示链接利用FAT实现随机访问。

- **索引分配**：

- 索引分配为每个文件建立一个索引表，通过索引表可以快速将文件的逻辑块号转换为物理块号。优点是支持随机访问，并且方便文件扩展，缺点是索引表会占用一定空间。

说一下文件链接的方式？

- 常见的又两种不同的文件链接方式，软链接和硬链接，他们都实现了文件的共享。

- **软链接**：
  - 也叫符号链接，它是一种特殊的文件，包含了指向其他文件或目录的路径信息。软链接可以实现网络文件共享。

- **硬链接**：
  - 硬链接是基于索引节点的共享方式。将文件的物理地址和属性等信息放在一个索引节点上，在文件目录中只设置文件名以及指向相应索引节点的指针。每个索引节点设置一个链接计数，只有当计数器为0时才会删除该文件。

介绍一下IO软件层次结构?

- IO软件目前普遍采用层次式结构，主要有一下几个层次：

- **用户层软件**：
  - 为用户提供与I/O操作相关的接口，如系统调用，库函数等。
- **设备独立性软件**：
  - 使得用户程序可以同一的方式访问不同类型的设备，例如对于网卡和磁盘都可以使用read（）和write（）命令来实现读写，提高了软件的可移植性和可维护性。
- **设备驱动程序**：
  - 负责将设备独立性软件层传过来的命令和参数转换为硬件设备可以理解的指令序列。
- **中断处理程序**：
  - 中断处理程序负责响应中断，保存当前的CPU环境，处理设备的中断请求，处理完毕再恢复被中断进程的现场后，返回到被中断进程。

介绍一下SPOOLing技术?

- 为了缓和CPU和I/O设备之间的速度矛盾，引入了假脱机技术，它是一种将独占设备改造为共享设备的技术。
- 该技术主要通过专门的外围控制机将低速I/O设备上的数据传送到高速的磁盘上，或者相反。主要的数据流向，从输入设备到输入缓冲区到输入井，或从输出井到输出缓冲区到输出设备。使得所有的慢速设备都可以拥有磁盘一样的速度，来执行系统发过来的I/O请求操作。大大提高了系统I/O速度

计算机网络中三种数据交换的方式?

- **电路交换**:
  - 电路交换分为三步：连接建立，数据传输和连接释放；优点是通信时延小，实时性强，缺点是在数据传输过程中物理线路只能被同一个用户独占，线路利用率低。
- **报文交换**:
  - 报文交换采用存储转发技术，优点是无需建立连接，可以动态分配线路缺点是对于较长的报文转发延时较高。
- **分组交换**:
  - 也采用存储转发技术，解决了报文交换中报文过长的问题。优点是线路利用率高；缺点是可能会出现失序，丢失和重复问题，使用虚电路可以解决失序问题。

OSI七层模型每一层的作用?

- **物理层**:
- 传输单位是比特，定义物理设备的标准，主要对物理连接方式，电气特性，机械特性等制定统一标准。
- **数据链路层**:
- 传输单位是帧，主要功能有封装成帧，透明传输和差错检验。还通过滑动窗口机制实现了流量控制和可靠传输。
- **网络层**:
- 传输单位为分组，网络层屏蔽了局域网内部的细节，实现了异构网络的互联，和路由与转发的功能。
- **传输层**:
- 传输单位为TCP报文段或UDP数据报，为运行在不同主机上的进程提供端到端的通信服务。提供面向连接和面向无连接的服务。
- **会话层**：负责建立维护和管理会话，协调不同应用程序之间的通信。
- **表示层**：进行编码转换，数据解析，管理数据的加密和解密。
- **应用层**：直接面向用户，提供各种类型的服务。

说一下TCP的三次握手过程？

- TCP连接建立的过程称为三次握手，过程如下：
- **第一次握手**：
- 客户端向服务器发送一个SYN标志位为1的数据包，表示想同服务器建立连接同时客户端进入SYN_SEND状态。
- **第二次握手**：
- 服务器收到客户端的SYN包后，回复客户端一个SYN和ACK标志位为1的数据包，表示愿意同客户端建立连接，同时服务器进入SYN_RCVD状态。
- **第三次握手**：
- 客户端收到服务器回复数据包后，向服务器发送一个ACK标志位为1的数据包，确认服务器对连接请求的确认，客户端进入ESTABLISHED状态，服务器在收到第三次握手的数据包后也进入ESTABLISHED状态。
- 通过这三次握手，服务器端和客户端都能够确认双方的发送和接受能力正常，从而建立起可靠的TCP连接。

介绍一下TCP的四次挥手过程？

- TCP连接断开的过程称为四次挥手，过程如下：
- **第一次挥手**：
- 客户端发送FIN标志位为1的数据包，表示请求断开与服务器的连接，客户端进入FIN-WAIT-1状态。
- **第二次挥手**：
- 服务器端回复ACK标志位为1的数据包，表示确认收到服务器的断开连接的请求，服务器端进入CLOSE-WAIT状态，客户端收到后进入FIN-WAIT-2状态。
- **第三次挥手**：
- 服务器会将剩余待发送的数据报文发送完毕后，发送给客户端一个FIN标志位为1的数据包，表示请求断开与客户端的连接，服务端进入LAST-ACK状态。
- **第四次挥手**：
- 客户端收到第三次挥手的报文后，回复给客户端ACK标志位为1的确认报文，同时进入TIME-WAIT状态，等待2MSL时间后会自动进入CLOSED状态，服务器端收到第四次挥手的报文后会进入CLOSED状态，连接完全断开

介绍一下UDP协议传输的过程？

- UDP协议是一种面向数据报的无连接不可靠的传输层协议，传输过程大致如下：

- **发送端**：

- 应用层将数据交给传输层，UDP协议会为数据添加UDP首部，包含源端口号，目的端口号，长度，校验和等字段。添加首部后UDP将数据报交给网络层处理，网络层将UDP数据报分割为若干个IP数据报，再添加IP首部然后根据目的IP进行路由选择，通过网络将数据发送出去。

- **接收端**：

- 当数据报到达时，首先由网络层根据IP首部中的信息，将数据报合并后后交给传输层的UDP协议处理，UDP协议根据长度校验和检验数据完整性然后根据目的端口交付给对应的应用层程序。
- UDP协议主要在IP报文的基础上实现了复用和分用以及差错校验的功能。

TCP和UDP协议的区别？

- TCP和UDP协议是两种不同的传输层协议。

- **TCP**：

- TCP是基于字节流的面向连接的可靠的传输层协议，TCP通过一系列的机制来保证传输的 可靠性，如滑动窗口，序列号，重传机制等等，还拥有流量控制和拥塞控制的功能。TCP主要应用在一些对数据可靠性要求较高的场景上，如邮件传输，文件传输，和网页传输上。

- **UDP**：

- UDP是基于数据报的面向无连接的不可靠的传输层协议，UDP通过减少报文头长度，提高了传输的效率，它主要优点在于有很高的实时性，主要用在对实时性要求高而对可靠性要求不高的场景中，如网络直播，在线游戏等

UDP协议能够实现可靠传输吗？

- UDP本身是无连接不可靠的传输协议，但通过一些额外的机制，UDP也能够实现可靠传输。

- **序列号**：
- 通过在应用层为每个数据报添加编号，来解决UDP数据报可能的乱序问题。

- **超时重传机制**：
- 通过在应用层添加超时重传机制，来解决UDP可能发生的数据丢失问题。
- 这种可靠性的实现需要开发者自行在应用层开发和维护，基本上相当于在应用层上把TCP协议的内容重写一遍。

介绍一下各种碰撞协议？

- 各种碰撞协议目的在于解决在共享信道的前提下尽可能的减少碰撞，提高数据传输效率。

- ALOHA协议：
- ALOHA协议的基本思路是不管当前信道中的情况如何当需要发数据时就发送，一段时间没有回复就重发，直到收到回复。这个协议的信道利用率很低。

- CSMA协议：
- CSMA协议添加了一个载波监听的装置，在发送前检测一下信道状态，如果信道忙就等待一段时间再检测，只有信道空闲的时候才会发送数据。

- CSMA/CD协议：
- 可简单概括为先听后发，边听边发，冲突停发，随机重发。

- CSMA/CA协议：
- 通过信道预约的机制来减少无线网络中发生碰撞的概率，从而提高信道传输效率。

介绍一下常见的网络层路由协议？

- 路由协议是指通过一系列算法找到一条从源路由器到目的路由器的最佳路径。

- **RIP协议**：
- RIP是一种基于距离向量的内部网关协议，通过向相邻的路由器交换自身全部的路由表来实现整体网络的收敛。主要缺点时故障信息收敛慢。

- **OSPF协议**：
- OSPF协议是一种基于链路状态的内部网关协议，当自身的路由表发生变化是，通过向网络中所有路由器交换变化信息来实现路由更新。使用Dijkstra算法来计算最佳的路径。

- **BGP协议**：
- BGP是一种用于不同自治系统的路由器之间交换信息的外部网关协议。它基于TCP协议，只力求寻找一条能够达到目的网络且较好的路由即可。

什么是数据库系统?

- 数据库库系统是由数据库，数据库管理系统，应用程序和数据库管理员组成的存储，管理，处理和维护数据的系统。
- **数据库**:
- 数据库是长期存储在计算机内，有组织的，可共享的大量数据的集合。
- **数据库管理系统**:
- 是位于用户和操作系统之间的一层数据管理软件，主要功能有，数据定义；数据操纵；数据组织存储和管理；数据库事务管理；数据库建立和维护。
- **数据库管理员**:
- 负责管理和控制数据库，主要职责：保证数据安全性和完整性；监控数据库的使用和运行。

什么是数据库的三级模式结构？

- 数据库的三级模式包括外模式，模式和内模式
- **外模式**：
- 也称用户模式，是用户能够看见和使用的局部数据的逻辑结构和特征的描述。
- **模式**：
- 也称逻辑模式，是数据库中全体数据的逻辑结构和特征的描述。
- **内模式**：
- 也称存储模式，一个数据库只有一个内模式，它是数据物理结构和存储方式的描述。
- **外模式/模式映像**：
- 当模式改变时，外模式可以保持不变，应用程序根据外模式编写，因此应用程序可以不必修改，保证了数据与程序的逻辑独立性。
- **模式/内模式映像**：
- 当数据库存储结构发生改变时，模式可以保持不变，进而应用程序也无需改变，保证了数据与程序的物理独立性。

介绍一下SQL的特点和功能?

- SQL特点:
  - 综合统一；高度非过程化；面向集合的操作方式；同一种语法提供多种使用方式；语言简洁，易学易用。
- SQL的主要功能:
  - 数据定义；数据查询；数据操纵；数据控制

介绍一下数据库查询操作?

- 数据库查询是数据库提供给用户的一个基本的功能，在SQL语言中，主要使用‘select’语句来进行查询。
- select语句中首先需要指明查询的属性名，然后给出查询的表名；然后通过where子句来添加查询条件；通过group by子句来对数据进行分组；通过order by可以实现按某一列的数据升序或降序排列。
- 当涉及到多个表的查询时还可以通过select语句实现连接查询，嵌套查询和集合查询。

介绍一下视图和它的作用？

- 视图是从一个或多个基本表或视图中导出的表，它是一个虚表，在数据库中仅存放视图的定义，不存放视图对应的数据。
- 视图的作用主要有以下几点：
- 简化用户操作；提供一定程度上的逻辑独立性；使用户可以以多角度看待同一数据；对机密数据提供安全保护；能更清晰地表达查询。

介绍一下数据库的规范化理论？

- 不合理的数据库设计会导致数据冗余，插入异常，删除异常，修改复杂等问题，通常是由不合理的数据依赖导致的，数据库的规范化解决了不合理的数据依赖的问题。
- 1NF：
- 数据库中的属性不可再分。
- 2NF：
- 在1NF的基础上，消除了非主属性对候选码的部份依赖。
- 3NF：
- 在2NF的基础上，消除了非主属性对候选码的传递依赖。
- BCNF：
- 在3NF的基础上，消除了主属性之间的部分函数依赖或传递依赖。

什么是数据库的事务，有哪些特性？

- 数据库的事务是指一系列的数据库操作，要么全部执行，要么全部不执行，是一个不可分割的整体。数据库主要有以下四个特性（ACID）：
- 原子性
- 一致性
- 隔离性
- 持久性

介绍一下数据库的死锁以及应对方式？

- 数据库死锁是指多个事务彼此互相等待对方的资源而陷入的一种僵持的局面。
- 数据库解决死锁主要有预防和诊断解除两种方法：
- **死锁预防**：
- 数据库的死锁预防主要有，一次封锁法和顺序封锁法。
- **死锁诊断解除**：
- 死锁诊断方法主要有，超时法和等待图法，对已经产生的死锁，一般选择持有资源最少的事务进行撤销操作。

什么是机器学习？

- 机器学习是指设计一种算法，使得该算法可以自动地寻找出输入数据中的规律和特征，并利用学到的规律和特征，对新的数据进行预测和分类的过程。
- 机器学习主要分为监督学习，无监督学习和强化学习。
- **监督学习**：
- 监督学习通过使用人工标记数据来学习输入数据和对应标签的关系。
- **无监督学习**：
- 无监督学习通过处理未标记数据，自动寻找数据中的结构模式或关系。
- **强化学习**：
- 强化学习通过奖励机制，来对模型的行为做出价值反馈，从而是使模型自动调节以获得最大奖励的过程。

什么是过拟合以及解决过拟合常见的办法?

- 过拟合是指模型在训练集上表现很好，而在测试集或新数据上表现很差，模型的泛化能力弱。

- **过拟合的原因**:

- 选择的模型太过复杂；使用了一些与寻找规律无关的输入变量，成为了模型学习过程中的噪音；数据量过少，导致无法准确寻找规律。

- **解决办法**:

- 正则化通过在损失函数中加入惩罚项限制模型的复杂度；使用特征工程对数据的关键特征进行提取降维，去除噪音；增加数据量使得模型可以学习到更广泛的规律。

介绍一下常见的损失函数?

- 损失函数主要分为回归与分类两种
- 回归问题:
- 均方误差（MSE），常用于回归问题，能直观反应预测值和真实值的平均平方误差，值越小表示模型性能越好;
- 平均绝对误差（MAE），也用于回归问题，不会放大异常值，但计算较为复杂
- 分类问题:
- 交叉熵损失，主要用于分类问题，尤其是在神经网络中，能衡量两个概率分布之间的差异，交叉熵越小，模型预测越准。

介绍一下决策树模型？

- 决策树模型是一种树形的机器学习模型，它的执行过程类似于一系列的IF-ELSE操作。
- 决策树的构建通过对数据中的不同特征计算相应的信息熵，信息增益或基尼系数，选择信息增益最大的特征作为根节点将所有数据集划分为两类，然后两边递归地执行上述过程，不断选择最佳划分特征，直到满足停止条件（叶节点为同一类别或特征使用完毕等）
- 预防过拟合的方法：
- **预剪枝**：
- 在决策树生成时，提前对节点进行一个评估，如果继续分裂不再带来性能的提升则停止分裂。
- **后剪枝**：
- 当一颗决策树生成完毕后，自底向上寻找，如果将一棵子树替换为叶节点可以提升模型性能则将其替换为叶节点。

介绍一下支持向量机?

- 支持向量机是一种有监督的机器学习算法，广泛用于分类和回归问题。
- 在线性可分的情况下，通过寻找一个超平面来将两类数据完全分开，且使两类数据点到超平面的距离最大化，这个超平面就是决策边界，距离超平面最近的哪些数据点被称为支持向量。
- 当线性不可分的时候，通过引入核函数，将原始数据映射到更高的特征空间，使数据在高维空间变得线性可分，然后再在高维空间中寻找最优超平面。
- 主要优点有泛化能力强，适合高维数据，缺点有计算复杂度高，核函数不太好选择

介绍一下神经网络?

- 神经网络是一种通过模仿生物神经网络结构和功能的计算模型，具有强大的学习和模式识别的能力。
- 主要工作流程分为前向传播和反向传播两个部分：
- **前向传播**：
- 输入数据从输入层进入神经网络，经过隐藏层，到达传输层。数据在神经元中每传输一步，神经元根据其连接权重对输入信号进行加权求和，并通过激活函数进行非线性变换，并将处理后的信号传递到下一层。
- **反向传播**：
- 用于训练神经网络，通过定义损失函数衡量神经网络的预测误差，并将误差反向传播到输入层，根据误差来调整权值，不断迭代到最终的满意水平。
- **优点**：对复杂特征的拟合能力强，适应能力强；可以并行计算
- **缺点**：需要大量数据和计算资源；模型复杂度高可解释性差；容易过拟合

什么是神经网络的学习率？

- 学习率是神经网络训练过程中一个重要的超参数，决定了每次参数更新的步长。
- 在神经网络训练过程中，较小的学习率可以确保收敛到最优点，但计算时间长耗费资源多，较大的学习率可以使得神经网络模型快速收敛，但是可能会导致模型不收敛。
- 神经网络训练中通常通过试错法和自适应学习率等方法调整学习率。

说一下L1正则化和L2正则化？

- L1正则化和L2正则化是机器学习中常见的正则化方法，用于防止模型过拟合。

- **L1正则化**：

- 通过在损失函数中添加参数的绝对值之和作为惩罚项，即L1范数，来约束模型的复杂度。L1正则化会使部分参数变为0，从而达到特征选择的目的，使模型更具有稀疏性。

- **L2正则化**：

- 通过在损失函数中添加参数的平方和作为惩罚项，即L2范数，来约束模型的复杂度。L2正则化不会使参数变为0，而实将参数值缩小，使模型的权重分布更加均匀，减少模型对某些特征的过度依赖。

神经网络中常见的激活函数有哪些?

- 激活函数主要是在神经元数据的线性计算中添加非线性元素，从而提高神经网络对非线性规律的拟合能力。
- Sigmoid函数:
- 函数将实数域映射到（0，1）之间，具有较好的非线性特性，能用于表示概率。但存在梯度消失问题，当x的绝对值较大时，梯度趋近于0，导致模型训练速度减慢。
- Tanh函数:
- 将实数域映射到（-1，1）之间，适用于要求输出为对称的场景，同时也存在梯度消失问题。
- ReLU函数:
- 计算简单，收敛速度快，能有效缓解梯度消失问题。但在训练过程中可能会出现某些神经元死亡的现象，即输出值一直为0。

机器学习模型的评价指标有哪些？

- 常见的评价指标分为分类任务和回归任务
- **分类任务**：
- 准确率：指预测正确的样本占总体的比例，在正负例均衡时才有参考意义。
- 精确率：预测为正例的样本中，真正为正例的个数比例。
- 召回率：真正为正例的样本中，预测为正例的个数比例。
- F1值：二倍的精确率乘召回率除以精确率加召回率。
- **回归任务**：
- 均方误差（MSE），平均绝对误差（MAE），R方拟合优度，衡量对数据的拟合程度。
- **聚类任务**：
- 轮廓系数，兰德指数。

神经网络梯度下降的优化方法?

- 神经网络一般对损失函数使用梯度下降的方式来寻求最优参数，但是原始的梯度下降计算方法每进行一次迭代都需要将神将网络所有参数迭代一遍，计算量过大，因此常采用以下优化方法：
- 小批量梯度下降：
- 每次迭代使用一小部分样本计算梯度并更新参数，即能减少训练时间，也能保证收敛稳定。
- 动量法：
- 在更新参数时，不仅考虑当前的梯度，还结合上一次的更新方向，通过积累动量来加速收敛，减少震荡。
- Adam：
- 可以在梯度下降的过程中对学习率进行自适应调整，在各种深度学习任务中都有很好的表现，是目前最常用的优化算法之一。能快速收敛到较优的解。

介绍一下Transformer?

- Transformer是一种基于注意力机制的深度学习架构，在自然语言处理领域有广泛的应用。Transformer的核心是自注意力机制，它能让模型在处理序列数据时更好的捕捉不同位置信息的依赖关系，主要解决了循环神经中无法处理长距离依赖的问题。它由编码器和解码器组成，编码器负责将序列数据转换为向量，并通过多层自注意力层和前馈神经网络来提取特征。解码器则根据编码器的输出以及之前生成的输出序列来生成下一个输出，同样包含自注意力层和前馈神经网络，不过解码器中的自注意力是掩码自注意力，防止在生成位置看到未来的信息。
- 优点：并行计算能力强；可以处理任意长度的序列数据；具有很强的表征能力，能学到复杂的语言模式和语义关系。

介绍一下Deepseek?

- Deepseek是基于Transformer的一种深度神经网络模型。它的核心创新点是采用了混合专家模型，它将前馈神经网络层分为了不同的子模型，即专家网络，每个专家擅长处理特定的上下文标记，换句话说就是把不同领域不同行业的知识进行了一个切分，每个专家只负责回答自己领域的问题；通过门控网络计算专家选择概论，使用SoftMax激活函数生成概率分布，每次仅激活概率最大前两名专家进行回答。减少了内存量和计算量。
- DeepSeek最大优势在于低成本高性能；同时它是一个开源模型，在可预见的未来发展一定会更加迅速。