**数据结构**

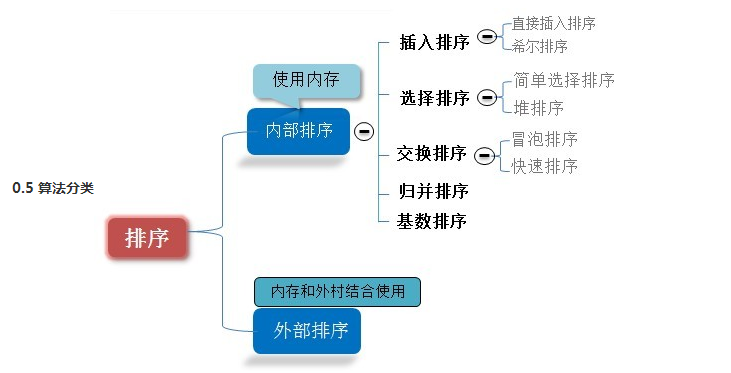
1. **排序算法中哪些最坏和平均的时间复杂度是一样的?**

冒泡排序，简单选择排序，直接插入排序，归并排序，堆排序，基数排序。

1. **数据结构中排序最优和最差相同的排序算法？**

简单选择排序，归并排序，堆排序，基数排序。





1. **最小生成树的算法有哪些，举个例子说明；**

Prim算法:

Prim算法基于一种贪心的思想，通过局部最优策略，每次将一条边加入所构建的生成树中，加完n-1条边后，保证最后所得的生成树是整体最优的，即最小生成树。

Kruskal算法：

Kruskal算法同样是基于贪心策略，但是它和Prim算法不同的是，在算法过程中它并不维护一个连通的分量，而是将多个连通分量合并到一起得到一颗生成树。

Kruskal算法具体实现：此算法可以称为“加边法”，初始最小生成树边数为0，每迭代一次就选择一条满足条件的最小代价边，加入到最小生成树的边集合里。

(1) 把图中的所有边按代价从小到大排序；

(2)把图中的n个顶点看成独立的n棵树组成的森林；

(3)按权值从小到大选择边，所选的边连接的两个顶点ui,vi,应属于两颗不同的树，则成为最小生成树的一条边，并将这两颗树合并作为一颗树。

(4) 重复(3),直到所有顶点都在一颗树内或者有n-1条边为止。

**4. 最短路径的算法，弗洛伊德和迪杰斯特拉有什么不同，用于什么情况。**

迪杰斯特拉(dijkstra)算法:

Dijkstra(迪杰斯特拉)算法是典型的最短路径路由算法，用于计算一个节点到其他所有节点的最短路径。主要特点是以起始点为中心向外层层扩展，直到扩展到终点为止。[Dijkstra算法](http://www.wutianqi.com/?p=1890" \t "_blank)能得出最短路径的最优解，但由于它遍历计算的节点很多，所以效率低。

弗洛伊德（Floyd）算法：

弗洛伊德（Floyd）算法是一个经典的动态规划算法。用通俗的语言来描述的话，首先我们的目标是寻找从点i到点j的最短路径

算法描述：a.从任意一条单边路径开始。所有两点之间的距离是边的权，如果两点之间没有边相连，则权为无穷大。

b.对于每一对顶点 u 和 v，看看是否存在一个顶点 w 使得从 u 到 w 再到 v 比己知的路径更短。如果是更新它。

区别：

一个是单源最短路，一个是每对顶点的最短路。

迪杰斯特拉属于贪心算法，弗洛伊德属于动态规划

迪杰斯特拉不能算有负权的，弗洛伊德可以算有负权的。

时间复杂度不同，O(n2),O(n3)

1. **拓扑排序中用了哪些结构？ 有向无环图？**

拓扑排序的实现步骤：

在有向图中选一个没有前驱的(入度为零)顶点并且输出

从图中删除该顶点和所有出边（即以它为头的弧）

重复上述两步，直至所有顶点输出，或者当前图中不存在无前驱的顶点为止，后者代表我们的有向图是有环的，因此，也可以通过拓扑排序来判断一个图是否有环。

1. **简述一下二叉排序树；**

二叉排序树（Binary Sort Tree），又称[二叉查找树](https://baike.baidu.com/item/%E4%BA%8C%E5%8F%89%E6%9F%A5%E6%89%BE%E6%A0%91/7077965" \t "_blank)（Binary Search Tree），亦称[二叉搜索树](https://baike.baidu.com/item/%E4%BA%8C%E5%8F%89%E6%90%9C%E7%B4%A2%E6%A0%91/7077855" \t "_blank)。它或者是一棵空树；

或者是具有以下性质的二叉树：  
1. 若它的左子树不空，则左子树上所有节点的值均小于它的根节点的值；  
2. 若它的右子树不空，则右子树上所有节点的值均大于它的根节点的值；  
3. 它的左右子树也分别为二叉排序树。

1. **简述一下线索二叉树**

对于n个结点的二叉树，在二叉链存储结构中有n+1个空链域，利用这些空链域存放在某种遍历次序下该结点的前驱结点和后继结点的指针，这些指针称为线索，加上线索的二叉树称为线索二叉树。

1. **二叉树与普通度为二的树的区别**

度为 2 的树要求每个节点最多只能有两棵子树，并且至少有一个节点有两棵子树。

二叉树的要求是度不超过 2，就是说度也可以是 1 或者 0。二叉树还有一个重要特点，区分左右子树；普通的树不分左右子树。

1. **数据结构的 4 个结构，各有什么特点**

集合结构：结构中的数据元素之间除了同属于一种类型外，无其他关系。  
线性结构：结构中的数据元素之间存在一对一的关系。  
树形结构：结构中的数据元素之间存在一对多的关系。  
图状结构或是网状结构：结构中的数据元素之间存在多对多的关系。

1. **线性表存储结构有哪些，优点缺点？线性存储和链式存储的优缺点比较？**

线性表具有两种存储结构即顺序存储结构和链接存储结构。

[顺序存储结构](https://www.baidu.com/s?wd=%E9%A1%BA%E5%BA%8F%E5%AD%98%E5%82%A8%E7%BB%93%E6%9E%84&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)是一种[物理结构](https://www.baidu.com/s?wd=%E7%89%A9%E7%90%86%E7%BB%93%E6%9E%84&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)，是按存储单元的顺序依次连续存放逻辑结构中所有结点形成的结构。 逻辑上彼此相邻的结点，在存储器上的物理位置也彼此比邻。  
[链式存储结构](https://www.baidu.com/s?wd=%E9%93%BE%E5%BC%8F%E5%AD%98%E5%82%A8%E7%BB%93%E6%9E%84&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)的结点是在元素[数据存储](https://www.baidu.com/s?wd=%E6%95%B0%E6%8D%AE%E5%AD%98%E5%82%A8&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)的同时附加存储一个指针数据。 指针的作用是指出该结点逻辑上的后继结点的存储位置。

**优缺点：**  
线性表的顺序存储结构，需要开辟一个定长的空间，不可扩充容量，可以直接存取数据元素，方便灵活、效率高，但插入、删除操作时将会引起元素的大量移动，因而降低效率。  
而在链接存储结构中内存采用动态分配，利用率高，但需增设指示结点之间关系的指针域，存取数据元素不如顺序存储方便，但结点的插入、删除操作较简单。

1. **散列表的建立方法？散列表会不会发生冲突，有哪些冲突解决的办法？**

 哈希表（Hash table，也叫散列表），是根据关键码值(Key value)而直接进行访问的数据结构。也就是说，它通过把关键码值映射到表中一个位置来访问记录，以加快查找的速度。这个映射函数叫做散列函数，存放记录的数组叫做散列表。

哈希函数的构造方法：

1. 数字分析法
2. 平方取中法
3. 去留余数法
4. 伪随机数法
5. 分段叠加法

处理冲突的方法：

1. 开放地址法
2. 再哈希法
3. 链地址法
4. 建立公共溢出区

这种方法的基本思想是：将哈希表分为**基本表**和**溢出表**两部分，凡是 和基本表发生冲突的元素，一律填入溢出表

1. **hash表的特点，hash 表适合存储什么类型的数据?**

散列表有两种用法：一种是 Key 的值与 Value 的值一样，一般我们称这种情况的结构为 Set（集合）；而如果 Key 和 Value 所对应的内容不一样时，那么我们称这种情况为 Map，也就是人们俗称的键值对集合。  
  
根据散列表的存储结构，我们可以得出散列表的以下特点。

**1) 访问速度很快**

由于散列表有散列函数，可以将指定的 Key 都映射到一个地址上，所以在访问一个 Key（键）对应的 Value（值）时，根本不需要一个一个地进行查找，可以直接跳到那个地址。所以我们在对散列表进行添加、删除、修改、查找等任何操作时，速度都很快。

**2) 需要额外的空间**

首先，散列表实际上是存不满的，如果一个散列表刚好能够存满，那么肯定是个巧合。而且当散列表中元素的使用率越来越高时，性能会下降，所以一般会选择扩容来解决这个问题。另外，如果有冲突的话，则也是需要额外的空间去存储的，比如链地址法，不但需要额外的空间，甚至需要使用其他数据结构。  
这个特点有个很常用的词可以表达，叫作“空间换时间”，在大多数时候，对于算法的实现，为了能够有更好的性能，往往会考虑牺牲些空间，让算法能够更快些。

**3) 无序**

散列表还有一个非常明显的特点，那就是无序。为了能够更快地访问元素，散列表是根据散列函数直接找到存储地址的，这样我们的访问速度就能够更快，但是对于有序访问却没有办法应对。

**4) 可能会产生碰撞**

没有完美的散列函数，无论如何总会产生冲突，这时就需要采用冲突解决方案，这也使散列表更加复杂。通常在不同的高级语言的实现中，对于冲突的解决方案不一定一样。

1. **图的存储**

## 邻接矩阵表示法

## 邻接表表示法

## 十字链表（有向图的优化）

## 邻接多重表

1. **图的深度和广度遍历是什么，工程上有什么实际应用？**

图的深度优先搜索(Depth First Search)，和树的先序遍历比较类似。

它的思想：假设初始状态是图中所有顶点均未被访问，则从某个顶点v出发，首先访问该顶点，然后依次从它的各个未被访问的邻接点出发深度优先搜索遍历图，直至图中所有和v有路径相通的顶点都被访问到。 若此时尚有其他顶点未被访问到，则另选一个未被访问的顶点作起始点，重复上述过程，直至图中所有顶点都被访问到为止。显然，深度优先搜索是一个递归的过程。

广度优先搜索算法(Breadth First Search)，又称为"宽度优先搜索"或"横向优先搜索"，简称BFS。

它的思想是：从图中某顶点v出发，在访问了v之后依次访问v的各个未曾访问过的邻接点，然后分别从这些邻接点出发依次访问它们的邻接点，并使得“先被访问的顶点的邻接点先于后被访问的顶点的邻接点被访问，直至图中所有已被访问的顶点的邻接点都被访问到。如果此时图中尚有顶点未被访问，则需要另选一个未曾被访问过的顶点作为新的起始点，重复上述过程，直至图中所有顶点都被访问到为止。

1. **如何在一个数组中找出两个数相加等于一个固定值的所有数对？**
2. 思路1: 暴力穷举法：两层for循环
3. 思路2：可以用hash表来存储数组中的元素，这样我们取得一个数后，去判断sum - val 在不在数组中，如果在数组中，则找到了一对二元组，它们的和为sum，该算法的缺点就是需要用到一个hash表，增加了空间复杂度。
4. 思路3：同样是基于查找，我们可以先将数组排序，然后依次取一个数后，在数组中用二分查找，查找sum -val是否存在，如果存在，则找到了一对二元组，它们的和为sum，该方法与上面的方法相比，虽然不用实现一个hash表，也没不需要过多的空间，但是时间多了很多。排序需要O(nLogn)，二分查找需要(Logn)，查找n次，所以时间复杂度为O(nLogn)。
5. 思路4：该方法基于第2种思路，但是进行了优化，在时间复杂度和空间复杂度是一种折中，但是算法的简单直观、易于理解。首先将数组排序，然后用两个指向数组的指针，一个从前往后扫描，一个从后往前扫描，记为first和last，如果 fist + last < sum 则将fist向前移动，如果fist + last > sum，则last向后移动。
6. **怎么确定单链表是一个环（数据结构）、**

**方法1：穷举遍历**  思路：首先从头节点开始，依次遍历单链表的每一个节点。每遍历到一个新节点，就从头节点重新遍历新节点之前的所有节点，用新节点ID和此节点之前所有节点ID依次作比较。如果发现新节点之前的所有节点当中存在相同节点ID，则说明该节点被遍历过两次，链表有环；如果之前的所有节点当中不存在相同的节点，就继续遍历下一个新节点，继续重复刚才的操作。

**方法2：哈希表缓存** 首先创建一个以节点ID为键的HashSet集合，用来存储曾经遍历过的节点。然后同样是从头节点开始，依次遍历单链表的每一个节点。每遍历到一个新节点，就用新节点和HashSet集合当中存储的节点作比较，如果发现HashSet当中存在相同节点ID，则说明链表有环，如果HashSet当中不存在相同的节点ID，就把这个新节点ID存入HashSet，之后进入下一节点，继续重复刚才的操作。这个方法在流程上和方法一类似，本质的区别是使用了HashSet作为额外的缓存。

**方法3：快慢指针**  首先创建两个指针1和2（在java里就是两个对象引用），同时指向这个链表的头节点。然后开始一个大循环，在循环体中，让指针1每次向下移动一个节点，让指针2每次向下移动两个节点，然后比较两个指针指向的节点是否相同。如果相同，则判断出链表有环，如果不同，则继续下一次循环。 此方法也可以用一个更生动的例子来形容：在一个环形跑道上，两个运动员在同一地点起跑，一个运动员速度快，一个运动员速度慢。当两人跑了一段时间，速度快的运动员必然会从速度慢的运动员身后再次追上并超过，原因很简单，因为跑道是环形的。

1. **汉络塔**

**递归**

**if n!=0 then ;预定值**

**func(n-1, a, c, b) ;将n-1个盘子由a移动到b，以c为辅助柱子**

**move a[n] to c ;将a上的最后一个盘子移动到c**

**func(n-1, b, a, c) ;将n-1个盘子由b移动到c，以a为辅助柱子**

**endif ;完成**

**组成原理**

1. **中断的过程是什么，断点的概念，什么是中断向量？**

中断全过程分为5个阶段：

[中断请求](https://baike.baidu.com/item/%E4%B8%AD%E6%96%AD%E8%AF%B7%E6%B1%82" \t "_blank)、[中断判优](https://baike.baidu.com/item/%E4%B8%AD%E6%96%AD%E5%88%A4%E4%BC%98" \t "_blank)、[中断响应](https://baike.baidu.com/item/%E4%B8%AD%E6%96%AD%E5%93%8D%E5%BA%94" \t "_blank)、[中断处理](https://baike.baidu.com/item/%E4%B8%AD%E6%96%AD%E5%A4%84%E7%90%86" \t "_blank)和中断返回

所谓断点就是程序被中断的地方

断点是人为设置的，意思就是让程序执行到此“停住”，不再往下执行，然后主动权就交给调试者，进行调试。

中断[标识码](https://baike.baidu.com/item/%E6%A0%87%E8%AF%86%E7%A0%81" \t "_blank)（[中断类型号](https://baike.baidu.com/item/%E4%B8%AD%E6%96%AD%E7%B1%BB%E5%9E%8B%E5%8F%B7" \t "_blank)）：

由硬件（通常是中断控制器）产生，以标识不同的[中断源](https://baike.baidu.com/item/%E4%B8%AD%E6%96%AD%E6%BA%90" \t "_blank)。

中断向量：

[中断服务程序](https://baike.baidu.com/item/%E4%B8%AD%E6%96%AD%E6%9C%8D%E5%8A%A1%E7%A8%8B%E5%BA%8F" \t "_blank)的入口地址。在某些计算机中，中断向量的位置存放一条跳转到中断服务程序入口地址的跳转指令。

[中断向量地址](https://baike.baidu.com/item/%E4%B8%AD%E6%96%AD%E5%90%91%E9%87%8F%E5%9C%B0%E5%9D%80" \t "_blank)：

存储中断向量的[存储单元](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8%E5%8D%95%E5%85%83" \t "_blank)地址

中断向量 是指早期的[微机](https://baike.baidu.com/item/%E5%BE%AE%E6%9C%BA" \t "_blank)系统中将由硬件产生的中断标识码（中断源的识别标志，可用来形成相应的中断服务程序的入口地址或[存放](https://baike.baidu.com/item/%E5%AD%98%E6%94%BE/9391805" \t "_blank)中断服务程序的首地址）。中断是指在计算机执行程序的过程中，当出现异常情况或者特殊请求时，计算机停止现行的程序的运行，转而对这些[异常处理](https://baike.baidu.com/item/%E5%BC%82%E5%B8%B8%E5%A4%84%E7%90%86/6250107" \t "_blank)或者特殊请求的处理，处理结束后再返回到现行程序的中断处，继续执行原程序。

1. **中断有几种?**

按照中断的功能来分,中断有以下五种类型：

I/O中断（输入输出中断）、外中断、硬件故障中断、程序性中断、访管中断

1. **cpu 和外设之间数据交换有哪几种**

1.程序查询方式。其特点是主机与I/O串行工作。CPU启动I/O后，时刻查询I/O是否准备好，若设备准备就绪，CPU便转入处理I/O与主机间传送信息的程序；若设备未做好准备，则CPU反复查询，“踏步”等待直到I/O准备就绪为止。可见这种方式CPU效率很低。

2.程序中断方式。其特点是主机与I/O并行工作。CPU启动I/O后，不必时刻查询I/O是否准备好，而是继续执行程序。当I/O准备就绪时，向CPU发中断请求信号，CPU在适当的时候响应I/O的中断请求，暂停现行程序为I/O服务。这种方式消除了“踏步”现象，提高了CPU的效率。

3.DMA方式。其特点是主机与I/O并行工作，主机与I/O之间有一条直接数据通路。CPU启动I/O后，不必查询I/O是否准备好

4.通道方式。通道是一个具有特有功能的处理器，CPU把部分权利下放给通道，由它实现对外围设备的统一管理和外围设备与主存之间的数据交换，大大提高了CPU的效率，但它是以花费更多的硬件为代价的。

5.I/O处理机方式。它是通道方式的进一步发展，CPU将I/O操作及外围设备的管理权全部交给I/O处理机，其实质是多机系统，因而效率有更大提高。

1. **Cache 有几种地址映射方法，分别有什么优缺点？**

**1.直接映射：**主存的一块只能复制到Cache的一个特定行位置上去，主存的地址有高位标记、字块地址、块内地址三个标记。这种映射关系实现简单，但是主存的块只能固定地对应着某个缓存块，不够灵活，命中率低。较适合容量大的Cache。

**2.全相联映射：**主存中的任一块都可以映像到Cache的任一块上，主存的地址有高位标记、块内地址两个标记。机制灵活，命中率高，但所需要的逻辑电路较多，成本高。较适合容量小的Cache。

**3.组相联映射：**是前两中的折中。主存的地址有高位标记、组地址、块内地址三个标记。比直接映像灵活，命中率高，比全相联映射所需成本低。较适合容量小的Cache。

1. **cpu 有几种设计方式**

CPU设计的两种方法：

硬布线逻辑控制

微序列控制器（微程序控制）

**计算机网络**

1. **中断交换机，路由器和集线器三者的区别**

路由器：（Router）是连接因特网中各局域网、广域网的设备。在路由器中记录着路由表，它会根据信道的情况自动选择和设定路由，以最佳路径，按前后顺序发送信号。发生在网络层。   
  交换机：（[Switch](https://www.baidu.com/s?wd=Switch&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "_blank)）是一种用于电（光）信号转发的网络设备。它可以为接入交换机的任意两个网络节点提供独享的电信号通路，把传输的信息送到符合要求的相应路由上。发生在数据链路层。   
  [集线器](https://www.baidu.com/s?wd=%E9%9B%86%E7%BA%BF%E5%99%A8&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "_blank)：（Hub）是指将多条以太网[双绞线](https://www.baidu.com/s?wd=%E5%8F%8C%E7%BB%9E%E7%BA%BF&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "_blank)或光纤集合连接在同一段物理介质下的设备。发生在物理层。

路由器和交换机的区别：

路由器 交换机

工作层次 网络层 数据链路层

转发依据 IP地址 Mac地址

功能 连接不同的网络 连接局域网中的电脑

宽带影响 共享宽带 独享宽带

集线器和交换机的区别：

交换机又称交换式集线器，它们俩很相似，都是基于MAC识别的，但是又有本质上的区别。

交换机 集线器

工作层次 数据链路层 物理层

宽带影响 独享 共享

数据传输 有目的发送 广播发送

传输模式 全双工或半双工 半双工

1. **交换机能不能用在大型的网络中**

**能**

1. **IP 地址和 MAC 地址的区别**

[IP地址](https://www.baidu.com/s?wd=IP%E5%9C%B0%E5%9D%80&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)  
对于[IP地址](https://www.baidu.com/s?wd=IP%E5%9C%B0%E5%9D%80&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)，即指使用[TCP/IP协议](https://www.baidu.com/s?wd=TCP%2FIP%E5%8D%8F%E8%AE%AE&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)指定给主机的32位地址。IP地址由用点分隔开的4个8八位组构成，如192.168.0.1就是一个IP地址，这种写法叫[点分十进制](https://www.baidu.com/s?wd=%E7%82%B9%E5%88%86%E5%8D%81%E8%BF%9B%E5%88%B6&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)格式。IP地址由网络地址和[主机地址](https://www.baidu.com/s?wd=%E4%B8%BB%E6%9C%BA%E5%9C%B0%E5%9D%80&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)两部分组成，分配给这两部分的位数随地址类（A类、B类、C类等）的不同而不同。网络地址用于[路由选择](https://www.baidu.com/s?wd=%E8%B7%AF%E7%94%B1%E9%80%89%E6%8B%A9&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)，而[主机地址](https://www.baidu.com/s?wd=%E4%B8%BB%E6%9C%BA%E5%9C%B0%E5%9D%80&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)用于在网络或子网内部寻找一个单独的主机。一个IP地址使得将来自源地址的数据通过路由而传送到目的地址变为可能。

[MAC地址](https://www.baidu.com/s?wd=MAC%E5%9C%B0%E5%9D%80&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)  
OSI（Open System Interconnection，[开放系统互连](https://www.baidu.com/s?wd=%E5%BC%80%E6%94%BE%E7%B3%BB%E7%BB%9F%E4%BA%92%E8%BF%9E&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)）7层网络协议（物理层，[数据链路层](https://www.baidu.com/s?wd=%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF%E5%B1%82&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)，网络层，传输层，会话层，表示层，应用层）参考模型中，第二层为[数据链路层](https://www.baidu.com/s?wd=%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF%E5%B1%82&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)（Data Link）。它包含两个子层，上一层是逻辑链路控制（LLC：Logical Link Control），下一层即是我们前面所提到的MAC（Media Access Control）层，即[介质访问控制](https://www.baidu.com/s?wd=%E4%BB%8B%E8%B4%A8%E8%AE%BF%E9%97%AE%E6%8E%A7%E5%88%B6&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)层。所谓介质（Media），是指[传输信号](https://www.baidu.com/s?wd=%E4%BC%A0%E8%BE%93%E4%BF%A1%E5%8F%B7&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)所通过的多种物理环境。常用网络介质包括电缆（如：双绞线，同轴电缆，光纤），还有微波、激光、红外线等，有时也称介质为物理介质。MAC地址也叫物理地址、硬件地址或链路地址，由网络设备制造商生产时写在硬件内部。这个地址与网络无关，也即无论将带有这个地址的硬件（如网卡、集线器、路由器等）接入到网络的何处，它都有相同的MAC地址，MAC地址一般不可改变，不能由用户自己设定。

MAC地址的长度、表示方法、分配方法及其唯一性  
MAC地址的长度为48位（6个字节），通常表示为12个16进制数，每2个16进制数之间用冒号隔开，如：08:00:20:0A:8C:6D就是一个MAC地址，其中前6位16进制数08:00:20代表网络硬件制造商的编号，它由IEEE（Istitute of Electrical and Electronics Engineers，电气与电子工程师协会）分配，而后3位16进制数0A:8C:6D代表该制造商所制造的某个网络产品（如网卡）的系列号。每个网络制造商必须确保它所制造的每个以太网设备都具有相同的前三字节以及不同的后三个字节。这样就可保证世界上每个以太网设备都具有唯一的MAC地址。

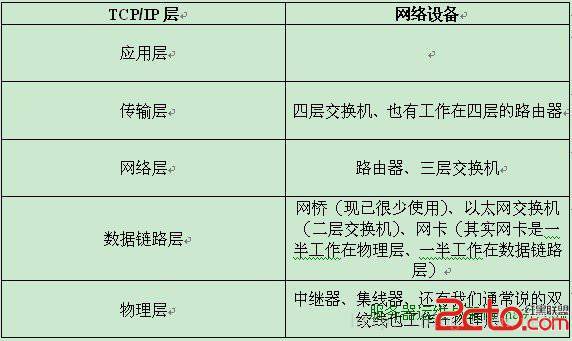
IP地址和MAC地址相同点是它们都唯一，不同的特点主要有：  
  
1.对于网络上的某一设备，如一台计算机或一台路由器，其IP地址可变（但必须唯一），而MAC地址不可变。我们可以根据需要给一台主机指定任意的IP地址，如我们可以给局域网上的某台计算机分配IP地址为192.168.0.112 ，也可以将它改成192.168.0.200。而任一网络设备（如网卡，路由器）一旦生产出来以后，其MAC地址永远唯一且不能由用户改变。  
2. 长度不同。IP地址为32位，MAC地址为48位。  
3. 分配依据不同。IP地址的分配是基于网络拓朴，MAC地址的分配是基于制造商。  
4. 寻址协议层不同。IP地址应用于OSI第三层，即网络层，而MAC地址应用在OSI第二层，即数据链路层。 数据链路层协议可以使数据从一个节点传递到相同链路的另一个节点上（通过MAC地址），而网络层协议使数据可以从一个网络传递到另一个网络上（ARP根据目的IP地址，找到中间节点的MAC地址，通过中间节点传送，从而最终到达目的网络）

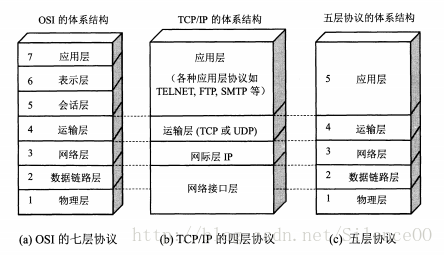
1. **计算机网络七层五层协议 4 层协议及各层作用各层设备，各设备之间的区别，各层的协议及相关概念**

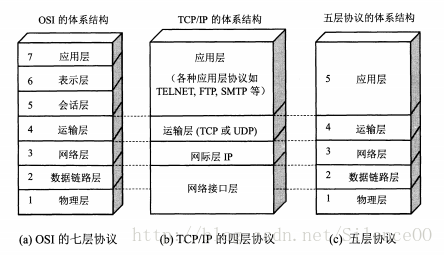
7层是指OSI七层协议模型，主要是：应用层（Application）、表示层（Presentation）、会话层（Session）、传输层（Transport）、网络层（Network）、数据链路层（Data Link）、物理层（Physical）。

5层只是OSI和TCP/IP的综合，是业界产生出来的非官方协议模型，但是很多具体的应用。实际应用还是TCP/IP的四层结构。为了方便可以把下两层称为网络接口层。五层体系结构包括：应用层、运输层、网络层、数据链路层和物理层。

4层是指TCP/IP四层模型，主要包括：应用层、运输层、网际层和网络接口层







1. **TCP/IP 有几层，还有就是为什么没有物理层？**

TCP/IP分四层：应用，传输，网络，网际接口。

TCP模型在网际接口几乎没定义任何协议，使TCP模型可以通过网际接口连接其他类型网络，比如：802的几个局域网协议（所以叫接口层）。（注：很多书中 网际接口被认为是OSI模型中物理层与[数据链路层](https://www.baidu.com/s?wd=%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF%E5%B1%82&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)的合并，这个说法就见仁见智了，我个人觉得也可以）  
  
至于为什么没有，就要从二者的区别说起，主要原因有这几点：OSI虽然完善但非常复杂，几乎无法实现，而TCP删除了很多不必要的层次，以达到简化的作用；主推OSI的人是各种专家，模型出来后却没有产品所以无法把握市场，而TCP是几大IT寡头共同推出，直接占领了市场。即TCP/IP模型出来时，OSI和很多通讯方面已经定义好底层的协议，不适合也没必要再改，同时TCP协议为了向后兼容未来的设备和开放性，故留了个模棱两可的网际接口层。

1. **OSI 模型有几层？其中传输层是做什么的？**

OSI参考模型分为七层结构，从下到上顺序依次为：

物理层、数据链路层、网络层、传输层、会话层、表示层、应用层

各层的作用

**物理层功能：**

利用传输介质为数据链路层提供物理连接，负责处理数据传输率并监控数据出错率，实现数据流的透明传输。

**数据链路层：**

在物理层提供的服务基础上，数据链路层在数据实体之间建立数据链路连接，传输以帧为单位的数据包，在采用差错控制和[流量控制](https://www.baidu.com/s?wd=%E6%B5%81%E9%87%8F%E6%8E%A7%E5%88%B6&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)方法，是有差错的物理链路便成无差错的数据链路。

**网络层：**

为分组通过网络选择合适的路径，实现路由选择和分组转发拥塞控制等。

**传输层：**

向用户提供的端到端服务，处理数据报错误，数据包次序，向高层屏蔽了下层数据通讯细节。

**会话层：**

维护两个计算机之间的传输链接，保证点到点传输不中断，以及管理数据交换等。

**表示层：**

用于处理两个通信系统中交换信息的表示方式，主要有数据格式交换，数据加密数据解秘，数据压缩等。

**应用层：**

为应用软件提供服务。

1. **网络层的设备有哪些？**

路由器、[三层交换机](https://www.baidu.com/s?wd=%E4%B8%89%E5%B1%82%E4%BA%A4%E6%8D%A2%E6%9C%BA&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)

1. **每个层都有什么设备 ，比如交换机 、路由器、 集线器等 、南桥北桥等概念。**

物理层设备：中继器，集线器

数据链路层设备：网桥，交换机

网路层设备：三层交换机，路由器

**南桥北桥：**

[主板芯片组](https://www.baidu.com/s?wd=%E4%B8%BB%E6%9D%BF%E8%8A%AF%E7%89%87%E7%BB%84&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank):（chIPsets）(pciset) ：分为南桥SB和北桥NB  
南桥（主外）:即系统[I/O芯片](https://www.baidu.com/s?wd=I%2FO%E8%8A%AF%E7%89%87&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)（SI/O）：主要管理中低速[外部设备](https://www.baidu.com/s?wd=%E5%A4%96%E9%83%A8%E8%AE%BE%E5%A4%87&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)；集成了中断控制器、[DMA控制器](https://www.baidu.com/s?wd=DMA%E6%8E%A7%E5%88%B6%E5%99%A8&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)。

北桥（主内）：系统控制芯片，主要负责CPU与内存、CPU与AGP之间的通信。掌控项目多为高速设备，如：CPU、Host Bus。后期北桥集成了[内存控制器](https://www.baidu.com/s?wd=%E5%86%85%E5%AD%98%E6%8E%A7%E5%88%B6%E5%99%A8&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)、Cache高速控制器；

1. **TCP，UDP 有什么区别？UDP有哪些应用？**

TCP(Transmission Control Protocol 传输控制协议)：是一种面向连接的、可靠的、基于字节流的传输层通信协议，使用三次握手协议建立连接、四次挥手断开连接。面向连接意味着两个使用TCP的应用(通常是一个客户端和一个服务器)在彼此交换数据包之前必须先建立一个TCP连接。在一个TCP连接中，仅有两方进行彼此通信，广播和多播不能用TCP。TCP 协议的作用是，保证数据通信的完整性和可靠性，防止丢包。TCP把连接作为最基本的对象，每一条TCP连接都有两个端点，这种端点我们叫作套接字(socket)，端口号拼接到IP地址即构成了套接字。

UDP(User Datagram Protocol 用户数据报协议)：是OSI(Open System Interconnection 开放式系统互联)参考模型中一种无连接的传输层协议，提供面向事务的简单不可靠信息传送服务。UDP协议的主要作用是将网络数据流量压缩成数据包的形式。

区别：

1.TCP提供的是面向连接的、可靠的数据流传输；UDP提供的是非面向连接的、不可靠的数据流传输。

2.TCP提供可靠的服务，通过TCP连接传送的数据，无差错、不丢失、不重复，按序到达；UDP尽最大努力交付，即不保证可靠交付。

3.TCP面向字节流；UDP面向报文。

4.TCP连接只能是点到点的；UDP支持一对一、一对多、多对一和多对多的交互通信。

5.UDP具有较好的实时性，工作效率比TCP高，适用于对高速传输和实时性有较高的通信或广播通信。

6.TCP对系统资源要求较多，UDP对系统资源要求较少。TCP首部有20字节；UDP的首部只有8个字节。

7.TCP的逻辑通信信道是全双工的可靠信道；UDP的逻辑通信信道是不可靠信道。

TCP是一种面向连接的、可靠的、基于字节流的传输层通信协议。是专门为了在不可靠的互联网络上提供一个可靠的端到端字节流而设计的，面向字节流。

UDP（用户数据报协议）是iso参考模型中一种无连接的传输层协议，提供面向操作的简单不可靠的非连接传输层服务，面向报文。

它们之间的区别：

1、TCP是基于连接的，可靠性高；UDP是基于无连接的，可靠性较低；

2、由于TCP是连接的通信，需要有三次握手、重新确认等连接过程，会有延时，实时性差；由于协议所致，安全性较高；而UDP无连接，无建立连接的过程，因而实时性较强，安全略差；

3、在传输相同大小的数据时，TCP首部开销20字节；UDP首部开销只有8个字节，TCP报头比UDP复杂，故实际包含的用户数据较少。TCP无丢包，而UDP有丢包，故TCP开销大，UDP开销较小；

4、每条TCP连接只能是点到点的；UDP支持一对一、一对多、多对一、多对多的交互通信。

所以对于应用这方面：

如果对实时性要求高和高速传输的场合下需要使用UDP；如果需要传输大量数据且对可靠性要求高的情况下应该使用TCP；在可靠性要求较低，追求效率的情况下应该使用UDP。

对于TCP来说，既然需要面向连接，就会涉及到三个阶段：连接建立、数据传送、连接释放三个过程；而“三次握手”就发生在连接建立这个阶段，下面对三次握手和四次挥手解释一下：

三次握手：

第一次握手：客户端发送一个TCP的syn标志位置为1的包（连接请求），指明客户打算连接服务器的端口；

第二次握手：当服务器收到连接请求之后，返回确认包（ack）应答，即将syn和ack标志位同时致为1（授予连接），并为这次连接分配资源；

第三次握手：客户端收到服务器的授予连接请求之后，再次发送确认包（ack）（syn标志位为0，ack标志位为1），并分配资源，这样TCP就建立连接了。

四次挥手：

中断连接端可以是客户端也可以是服务端

假设是客户端发起的中断连接请求，则客户端发送fin报文，服务端接收到客户端发送的fin报文后，意思是说：“客户端已经没有数据要发送给你了”，但是这时候如果服务端有数据没有发送完成，则不必着急关闭socket，可以继续发送数据，所以服务端先发送ack，告诉客户端，收到了客户端刚才的关闭连接请求，但是服务端还没有准备好，还有数据没有发送完，所以请客户端继续等我的消息，这个时候客户端就进入了fin\_wait状态，继续等待服务端的fin报文；当服务端确定已经发送完数据后，则想客户端发送fin报文，告诉客户端已经发送完数据，可以关闭连接了；当客户端接收到服务端的fin报文后，就知道可以关闭连接了，但是还是不相信网络，怕服务端不知道要关闭，所以客户端发送ack给服务端后进入了time\_wait状态，如果服务端没有收到ack则可以重传；服务端收到ack后，就知道可以关闭连接了；当服务端等待了2msl后依然没有收到回复，则证明服务端已经正常关闭了，那客户端也会关闭连接了，就这样，TCP连接就完成了关闭。

1. **内部网关协议有哪些，各有什么特点**

[IGP](https://baike.baidu.com/item/IGP" \t "_blank)（内部[网关](https://baike.baidu.com/item/%E7%BD%91%E5%85%B3/98992" \t "_blank)[协议](https://baike.baidu.com/item/%E5%8D%8F%E8%AE%AE/670528" \t "_blank)）是在一个自治[网络](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C/143243" \t "_blank)内[网关](https://baike.baidu.com/item/%E7%BD%91%E5%85%B3" \t "_blank)（[主机](https://baike.baidu.com/item/%E4%B8%BB%E6%9C%BA/455151" \t "_blank)和[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8/108294" \t "_blank)）间换[路由](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1/363497" \t "_blank)信息的[协议](https://baike.baidu.com/item/%E5%8D%8F%E8%AE%AE" \t "_blank)。[路由](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1/363497" \t "_blank)信息能用于[网间协议](https://baike.baidu.com/item/%E7%BD%91%E9%97%B4%E5%8D%8F%E8%AE%AE/10502135" \t "_blank)（[IP](https://baike.baidu.com/item/IP" \t "_blank)）或者其它[网络协议](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%8D%8F%E8%AE%AE/328636" \t "_blank)来说明路由传送是如何进行的。IGP[协议](https://baike.baidu.com/item/%E5%8D%8F%E8%AE%AE/670528" \t "_blank)包括RIP、OSPF、IS-IS、IGRP、EIGRP。

[RIP](https://baike.baidu.com/item/RIP" \t "_blank)（Routing Information Protocol） ,[路由信息协议](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E4%BF%A1%E6%81%AF%E5%8D%8F%E8%AE%AE" \t "_blank)，是内部[网关](https://baike.baidu.com/item/%E7%BD%91%E5%85%B3" \t "_blank)协议中应用最广泛的一种协议，它是一种分布式的，基于[距离向量](https://baike.baidu.com/item/%E8%B7%9D%E7%A6%BB%E5%90%91%E9%87%8F" \t "_blank)的[路由选择协议](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E9%80%89%E6%8B%A9%E5%8D%8F%E8%AE%AE" \t "_blank)，其特点是协议简单。适用于相对较小的[自治系统](https://baike.baidu.com/item/%E8%87%AA%E6%B2%BB%E7%B3%BB%E7%BB%9F" \t "_blank)，它们的直径“跳数”一般小于15。 RIP[协议](https://baike.baidu.com/item/%E5%8D%8F%E8%AE%AE" \t "_blank)特点是：（1）仅和相邻[路由](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1" \t "_blank)交换信息（2）[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8" \t "_blank)交换的信息是当前本[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8" \t "_blank)所知道的全部信息，即自己的路由表。也就是说，交换的信息是：“我到本[自治系统](https://baike.baidu.com/item/%E8%87%AA%E6%B2%BB%E7%B3%BB%E7%BB%9F" \t "_blank)中所有[网络](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C" \t "_blank)的（最短）距离，以及到那个网络应经过的[下一跳](https://baike.baidu.com/item/%E4%B8%8B%E4%B8%80%E8%B7%B3" \t "_blank)路由器。”（3）按固定时间间隔交换[路由](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1" \t "_blank)信息，例如，每隔30秒。然后[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8" \t "_blank)根据收到的路由信息更新[路由表](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E8%A1%A8" \t "_blank)。

[OSPF](https://baike.baidu.com/item/OSPF" \t "_blank)开放最短路径优先(Open Shortest Path First)，是一个内部[网关](https://baike.baidu.com/item/%E7%BD%91%E5%85%B3" \t "_blank)[协议](https://baike.baidu.com/item/%E5%8D%8F%E8%AE%AE" \t "_blank)(Interior Gateway Protocol,简称IGP)，用于在单一[自治系统](https://baike.baidu.com/item/%E8%87%AA%E6%B2%BB%E7%B3%BB%E7%BB%9F" \t "_blank)(autonomous system,AS)内决策[路由](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1" \t "_blank)。与RIP相对，OSPF是[链路状态路由协议](https://baike.baidu.com/item/%E9%93%BE%E8%B7%AF%E7%8A%B6%E6%80%81%E8%B7%AF%E7%94%B1%E5%8D%8F%E8%AE%AE" \t "_blank)，而RIP是[距离向量路由协议](https://baike.baidu.com/item/%E8%B7%9D%E7%A6%BB%E5%90%91%E9%87%8F%E8%B7%AF%E7%94%B1%E5%8D%8F%E8%AE%AE" \t "_blank)。链路是[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8" \t "_blank)接口的另一种说法，因此OSPF也称为接口状态[路由协议](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%8D%8F%E8%AE%AE" \t "_blank)。OSPF通过[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8" \t "_blank)之间通告[网络](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C" \t "_blank)接口的状态来建立链路状态数据库，生成[最短路径](https://baike.baidu.com/item/%E6%9C%80%E7%9F%AD%E8%B7%AF%E5%BE%84" \t "_blank)树，每个OSPF[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8" \t "_blank)使用这些最短路径构造路由。最主要的特点是使用分布式的链路状态[协议](https://baike.baidu.com/item/%E5%8D%8F%E8%AE%AE" \t "_blank)，而不是像RIP那样的[距离向量协议](https://baike.baidu.com/item/%E8%B7%9D%E7%A6%BB%E5%90%91%E9%87%8F%E5%8D%8F%E8%AE%AE" \t "_blank)。三个要点：（1）向本[自治系统](https://baike.baidu.com/item/%E8%87%AA%E6%B2%BB%E7%B3%BB%E7%BB%9F" \t "_blank)中所有路由器发送信息。（2）发送的信息就是与本[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8" \t "_blank)相邻的所有路由器的链路状态，但这只是路由器所知道的部分信息。（3）只有在链路状态发生变化时，[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8" \t "_blank)才向所有路由器用[洪泛](https://baike.baidu.com/item/%E6%B4%AA%E6%B3%9B" \t "_blank)法发送此信息。

## IGRP协议（内部网关路由协议）

## EIGRP（增强内部网关路由协议）

## IS-IS协议（中间系统到中间系统路由协议）

1. **路由表中有哪些字段？路由表表项？**

路由表中的表项内容包括：

* destination：目的地址，用来标识IP包的目的地址或者目的网络。
* mask：网络掩码，与目的地址一起标识目的主机或者路由器所在的网段的地址。
* pre：标识路由加入IP路由表的优先级。可能到达一个目的地有多条路由，但是优先级的存在让他们先选择优先级高的路由进行利用。
* cost：路由开销，当到达一个目的地的多个路由优先级相同时，路由开销最小的将成为最优路由。
* interface：输出接口，说明IP包将从该路由器哪个接口转发。
* nexthop：下一跳[IP](https://baike.baidu.com/item/IP/224599" \t "_blank)地址，说明IP包所经过的下一个路由器。

1. **计算机网络的地址、地址的分类？**

计算机网络地址共分为五类：A类、B类、C类、D类、E类。

**A类地址:0-127**

A类地址的表示范围为：0.0.0.0~126.255.255.255，默认网络掩码为：255.0.0.0；A类地址分配给规模特别大的网络使用。A类网络用第一组数字表示网络本身的地址，后面三组数字作为连接于网络上的主机的地址。分配给具有大量主机（直接个人用户）而局域网络个数较少的大型网络。例如IBM公司的网络。

**B类地址：128-191**

B类地址的表示范围为：128.0.0.0~191.255.255.255，默认网络掩码为：255.255.0.0；B类地址分配给一般的中型网络。B类网络用第一、二组数字表示网络的地址，后面两组数字代表网络上的主机地址。

**C类地址：192-223**

C类地址的表示范围为：192.0.0.0~223.255.255.255，默认网络掩码为：255.255.255.0；C类地址分配给小型网络，如一般的局域网和校园网，它可连接的主机数量是最少的，采用把所属的用户分为若干的网段进行管理。C类网络用前三组数字表示网络的地址，最后一组数字作为网络上的主机地址。

**D类地址：**

D类地址主要是用于组播，前面4位主要起的作用是引导，后面28位位组播的地址ID。

**E类地址：**

总是以1111四位引导 ，E类地址用于研究用。

**特殊的IP地址**

1.网络地址：

IP中主机地址为0的地址表示网络地址，如128.211.0.0。

2.广播地址：

网络号后跟所有位全是1的后缀，就是直接广播地址。

3.回送地址：

即127.0.0.1，用于测试。

1. **TCP/IP 位于哪层？**

TCP是[TCP/IP](https://www.baidu.com/s?wd=TCP%2FIP&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)的第三层传输层，对应OSI的第四层传输层；  
IP是[TCP/IP](https://www.baidu.com/s?wd=TCP%2FIP&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)的第二层互联层，对应OSI的第三层网络层。

1. **路由选择协议 RIP 和OSPF**

RIP是一种分布式的基于距离向量的路由选择协议，

OSPF是一种分布式的基于链路状态的路由选择协议

RIP：仅于相邻的路由器交换状态；路由器交换的信息是当前本路由器所知道的全部信息，即路由表；按固定的时间间隔交换路由信息

OSPF：向本自治系统所有的路由器发送信息；发送的信息是与本路由器的相邻的所有路由器的链路状态，即本路由器周边的网络拓扑；只有当链路状态发生改变时，路由器才向所有路由器用洪泛法发送信息。

在RIP中，每个路由器只知道到所有路由器的距离以及下一跳路由器，不知道全网的拓扑结构，且由于所处位置不同，每个路由器的路由表不同；发生网络故障时，要经过较长时间才能将此信息传送给所有路由器，即所谓的“好消息传得快，坏消息传得慢”，收敛慢

在OSPF中，由于交换的信息是相邻的路由器的链路状态，故每一个路由器都能建立起一个链路状态数据库（即全网的拓扑结构图），该数据库在全网范围内是一致的（这称为“链路状态数据库的同步”）。在ＯＳＰＦ中，链路数据库能较快的更新，故ＯＳＰＦ的更新过程收敛快

ＲＩＰ协议使用运输层的用户数据报ＵＤＰ进行传送，ＯＳＰＦ直接使用ＩＰ数据报进行传送

1. **路由的算法****。**

路由算法，又名选路算法，可以根据多个特性来加以区分。算法的目的是找到一条从源路由器到目的路由器的“好”路径（即具有最低费用的路径）。

典型种类：

LS算法、Dijkstra算法、距离向量[算法](https://baike.baidu.com/item/%E7%AE%97%E6%B3%95" \t "_blank)（也称为[Bellman-Ford算法](https://baike.baidu.com/item/Bellman-Ford%E7%AE%97%E6%B3%95" \t "_blank)）

路由控制有各种各样的算法，其中最具代表性的有两种，是距离向量算法和链路状态算法

距离向量算法(Distance-Vector)是指根据距离(代价：相当于所要经过的路由器的个数)和方向决定目标网络或目标主机位置的一种方法。

链路状态算法(Link-State)是路由器在了解网络整体连接状态的基础上生成路由控制表的一种方法。该方法中，每个路由器必须保持同样的信息才能进行正确的路由选择。

1. **路由器内部有什么协议？**

RIP（路由信息协议）、IGRP（内部网关路由协议）、EIGRP（增强型内部网关路由协议）、OSPF（开放式最短路径优先）、IS-IS（中间系统到中间系统）是内部网关协议（[IGP](https://baike.baidu.com/item/IGP" \t "_blank)）

1. **计算机网络里面哪里有流量控制，****~~缓存满了怎么调度~~**

所谓流量控制就是让发送方的发送速率不要太快，要让接收方来得及接收。**TCP滑动窗口的流量控制和拥塞控制**

TCP网络传输是面向字节流：TCP中的流（Stream）就是指的是流入到进程或者从进程流出的字节序列。虽然应用程序和TCP的交互是一次一个数据块（也就是我们定义的不同大小的Buffer长度），但是TCP把应用程序交互下来的数据仅仅是看成是一连串的无结构的字节流，并且将这一连串应用程序交互下来的数据放入TCP中的发送缓存中。

1. **子网掩码的作用**

子网掩码是一种用来指明一个IP地址所标示的主机处于哪个子网中。 子网掩码不能单独存在，它必须结合IP地址一起使用。

子网掩码是一个32位地址，是与IP地址结合使用的一种技术。它的主要作用有两个，一是用于屏蔽IP地址的一部分以区别[网络标识](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E6%A0%87%E8%AF%86" \t "_blank)和[主机](https://baike.baidu.com/item/%E4%B8%BB%E6%9C%BA" \t "_blank)标识，并说明该IP地址是在[局域网](https://baike.baidu.com/item/%E5%B1%80%E5%9F%9F%E7%BD%91" \t "_blank)上，还是在远程网上。二是用于将一个大的IP网络划分为若干小的子网络。（将某个IP地址划分成[网络地址](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%9C%B0%E5%9D%80" \t "_blank)和[主机地址](https://baike.baidu.com/item/%E4%B8%BB%E6%9C%BA%E5%9C%B0%E5%9D%80" \t "_blank)两部分）

**操作系统**

1. **处理机管理**

处理机管理也称进程管理。重点研究诸进程之间的并发特性，以及进程之间相互合作与资源竞争产生的问题。

1. **文件管理**

文件管理是操作系统的五大职能之一，主要涉及文件的逻辑组织和物理组织，目录的结构和管理。所谓文件管理，就是操作系统中实现文件统一管理的一组软件、被管理的文件以及为实施文件管理所需要的一些数据结构的总称（是操作系统中负责存取和管理文件信息的机构）从系统角度来看，[文件系统](https://baike.baidu.com/item/%E6%96%87%E4%BB%B6%E7%B3%BB%E7%BB%9F/4827215" \t "_blank)是对文件[存储器](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8%E5%99%A8/1583185" \t "_blank)的存储空间进行组织，分配和回收，负责文件的存储，检索，共享和保护。从用户角度来看，文件系统主要是实现"按名取存"，文件系统的用户只要知道所需文件的文件名，就可存取文件中的信息，而无需知道这些文件究竟存放在什么地方。

**软件工程**

1. **软件生命周期是什么，解释一下****瀑布模型？它适合哪种用户？**

软件生命周期(SDLC，Systems Development Life Cycle,SDLC)是软件的产生直到报废或停止使用的生命周期.周期内有问题定义、[可行性分析](https://www.baidu.com/s?wd=%E5%8F%AF%E8%A1%8C%E6%80%A7%E5%88%86%E6%9E%90&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)、总体描述、系统设计、编码、调试和测试、验收与运行、维护升级到废弃等阶段。

瀑布模型是将软件生存周期的各项活动规定为按固定顺序而连接的若干阶段工作，形如瀑布流水，最终得到软件产品。

在瀑布模型中，软件开发的各项活动严格按照线性方式进行，当前活动接受上一项活动的工作结果，实施完成所需的工作内容。当前活动的工作结果需要进行验证，如果验证通过，则该结果作为下一项活动的输入，继续进行下一项活动，否则返回修改。

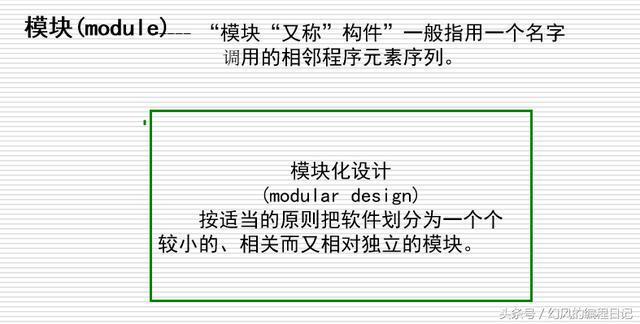
瀑布模型适用于那些结构比较固定，不经常变化的项目。

1. **软件工程的瀑布模型有哪几个阶段？**

计划时期，开发时期，运行维护时期  
计划时期包括：问题定义，可行性研究，需求分析  
开发时期包括：概要设计，详细设计，软件实现，软件测试  
运行维护时期包括：运行维护

1. **问了模块这个概念和一些设计模式，~~做项目的过程等~~**

**模块是什么意思 区分模块的要素是什么？**



模块独立模块的独立程度可以由两个定性标准度量，这两个标准分别称为内聚和耦合

耦合是模块和模块直接的，尽量追求低耦合

内聚是模块内元素之间的，尽量追求低内聚

1. **模块间的耦合度**。



1. **软件工程的三个模型:瀑布，增量，喷泉**

**瀑布模型**：线性顺序模型从系统分析开始，逐步经过各个开发阶段到软件开发完毕、交付使用止。每个阶段的变换结果是下一个阶段的变换的输入，相邻的两个阶段具有极其密切的因果关系。该模型以软件的需求能够完全被确定为前提，这种模型的特点是“一 泻千里”、易“下”而几乎不可能“上”，因此又得名“瀑布模型”。

**增量模型**：是把待开发的软件系统模块化，将每个模块作为一个增量组件，从而分批次地分析、设计、编码和测试这些增量组件。运用增量模型的软件开发过程是递增式的过程。相对于瀑布模型而言，采用增量模型进行开发，开发人员不需要一次性地把整个软件产品提交给用户，而是可以分批次进行提交

**喷泉模型：**（fountain model）是一种以用户需求为动力，以对象为驱动的模型，主要用于描述[面向对象](https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1/2262089" \t "_blank)的软件开发过程。该模型认为[软件开发过程](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91%E8%BF%87%E7%A8%8B/3758917" \t "_blank)自下而上周期的各阶段是相互迭代和无间隙的特性。

**数据库**

1. **数据库有哪些锁，各自的作用，举个例子说明一下。**

锁分类：

从数据库系统角度分为三种：排他锁、共享锁、更新锁。   
共享（S)锁：多个事务可封锁一个共享页；任何事务都不能修改该页； 通常是该页被读取完毕，S锁立即被释放。   
排它（X)锁：仅允许一个事务封锁此页；其他任何事务必须等到X锁被释放才能对该页进行访问；X锁一直到事务结束才能被释放。   
更新（U)锁：用来预定要对此页施加X锁，它允许其他事务读，但不允许再施加U锁或X锁；当被读取的页将要被更新时，则升级为X锁；U锁一直到事务结束时才能被释放。

1. **什么是事务**

数据库事务（简称：事务）是[数据库管理系统](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F" \t "_blank)执行过程中的一个逻辑单位，由一个有限的[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93" \t "_blank)操作序列构成。

1. **sql server中事务有哪三种语句？**

BEGIN TRANSACTION 事务名 /\*标记一个显式本地事务的起始点\*/  
COMMIT TRAN 事务名 /\*提交事务\*/  
ROLLBACK TRAN 事务名 /\*[回滚事务](https://www.baidu.com/s?wd=%E5%9B%9E%E6%BB%9A%E4%BA%8B%E5%8A%A1&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)\*/

1. **存储过程，索引，函数。**

存储过程：是存储在数据库目录中的一些的声明性SQL语句。

索引：数据库中专门用于帮助用户快速查找数据的一种数据结构。类似于字典中的目录，查找字典内容时可以根据目录查找到数据的存放位置吗，然后直接获取。

函数：

单行函数：针对sql语句影响的每一行都进行处理，并针对这一行返回一个结果sql影响多少行就返回多少个结果

组函数 ：对sql语句影响的所有行进行综合处理，最终返回一个结果无论sql语句影响多少行都只返回一个结果（组函数会忽略空值）

1. **索引的作用是什么？**

帮助用户快速查找数据。可以利用索引快速访问数据库表中的特定信息。

1. **主键、外键？**

主键--唯一标识一条记录，不能有重复的，不允许为空，只有一个

外键--表的外键是另一表的主键,外键可以有重复的,可以是空值，可多个

主键--用来保证数据完整性 外键--用来和其他表建立联系用的

1. **~~还有~~****~~数据库的几个英文含义？~~**

**编译原理**

1. **编译的过程分为那些阶段，其中哪几个阶段是必不可少的？**

编译程序的工作过程一般划分为五个阶段：

词法分析、语法分析、语义分析与中间代码产生、优化、目标代码生成。

**词法分析**的功能是对输入的高级语言源程序进行词法分析，识别其中的单词符号，确定它们的种类，交给语法分析器，即把字符串形式的源程序分解为单词符号串形式。  
   **语法分析**的功能是在词法分析结果的基础上，运用语言的语法规则，对程序进行语法分析，识别构成程序的各类语法范畴及它们之间的层次关系，并把这种层次关系表达成语法树的形式。  
   **词义分析和中间代码生成**的功能是在语法分析的基础上，对程序进行语义分析，“理解”其含义，产生出表达程序语义的内部表达形式(中间代码)。表达语义的中间代码与机器代码相似，可以认为这一步进行了翻译工作，把源程序翻译成了与具体机器指令系统无关的抽象指令代码。  
   **优化**的功能是按照等价变换的原则，对语义分析器产生的中间代码序列进行等价变换，删除其中多余的操作，对耗时耗空间的代码进行优化，以期最后得到高效的可执行代码。  
   **目标代码生成**的功能是把优化后的中间代码变换成机器指令代码，得到可在目标机器上执行的机器语言程序。

**其他**

1. **讲一讲 java，python ，c 和 c++的区别？**

python： 快速开发应用程序  
java： 健壮的大型软件  
C++： 需求效率的软件  
C： 操作系统及驱动

C: C是几种语言中最古老的。C是C++的子集。C的最初出现是为了代替运行效率高但是开发效率低下的汇编语言。C语言现在多应用于操作系统编程，或者驱动开发。比如著名的Linux系统就是使用C语言开发的。C++也可以开发操作系统但是太过于笨重。像python 或者java这样的语言不适合这样低级的开发。

C++ : C++是C和java特点的结合。它不仅支持传统的面向过程编程，也支持面向对象编程。C++运行效率较高，同时能够比较容易地建立大型软件，适合对效率要求高的软件，比如机器学习中的神经网络，大型游戏编程等等。

JAVA : [java](https://cloud.tencent.com/developer/information/java" \t "_blank)由于其优越的跨平台可移植性，在Web开发中是主流语言。在加上手机Android系统的发展，使得java开发人员的需求量很大。同时，Java现在也用来开发手游。Java 的语法相对规范。  
Python : Python是动态形的灵活的解释性语言，从软件开发到Web开发，Python都有在被使用。因为他的解释性，适合轻量级开发。Python是比较好学的语言。

1. **图形和图像的区别**

图形是指由外部轮廓线条构成的矢量图，一般指用计算机绘制的画面，如直线、圆、圆弧、任意曲线和图表等；不会失真。

图像则是指由输入设备捕捉的实际场景画面或以数字化形式存储的任意画面。失真。

1. **~~数字信号处理接的正交变换~~**

1. **还有 bs，cs 的区别**

CS即Client/Server(客户机/服务器)结构，C/S结构在技术上很成熟，它的主要特点是交互性强、具有安全的存取模式、网络通信量低、响应速度快、利于处理大量数据。但是该结构的程序是针对性开发，变更不够灵活，维护和管理的难度较大。通常只局限于小型局域网，不利于扩展。并且，由于该结构的每台客户机都需要安装相应的客户端程序，分布功能弱且兼容性差，不能实现快速部署安装和配置，因此缺少通用性，具有较大的局限性。要求具有一定专业水准的技术人员去完成。   
   
 BS即Browser/Server(浏览器/服务器)结构，就是只安装维护一个服务器(Server)，而客户端采用浏览器(Browse)运行软件。B/S结构应用程序相对于传统的C/S结构应用程序是一个非常大的进步。 B/S结构的主要特点是分布性强、维护方便、开发简单且共享性强、总体拥有成本低。但数据安全性问题、对服务器要求过高、数据传输速度慢、软件的个性化特点明显降低，这些缺点是有目共睹的，难以实现传统模式下的特殊功能要求。例如通过浏览器进行大量的数据输入或进行报表的应答、专用性打印输出都比较困难和不便。此外，实现复杂的应用构造有较大的困难。  
  
一、硬件环境  
C/S用户固定，并且处于相同区域，要求拥有相同的操作系统。  
B/S要有操作系统和浏览器就行。与操作系统平台无关。  
二、客户端要求  
C/S客户端的计算机电脑配置要求较高。  
B/S客户端的计算机电脑配置要求较低。  
  
四、软件安装  
C/S每一个客户端都必须安装和配置软件,客户端不必安装，使用浏览器访问，易推广。  
B/S最大的优点就是可以在任何地方进行操作而不用安装任何专门的软件。  
五、软件升级和维护  
C/S每一个客户端都要升级程序。可以采用自动升级。BS客户端不必安装及维护。  
六、安全性  
C/S一般面向相对固定的用户群，程序更加注重流程，它可以对权限进行多层次校验，提供了更安全的存取模式，对信息安全的控制能力很强。一般高度机密的信息系统采用C/S结构适宜。

1. **傅里叶变换？**

[傅立叶变换](https://baike.baidu.com/item/%E5%82%85%E7%AB%8B%E5%8F%B6%E5%8F%98%E6%8D%A2/3472079" \t "_blank)，表示能将满足一定条件的某个函数表示成三角函数（正弦和/或余弦函数）或者它们的积分的线性组合。在不同的研究领域，傅立叶变换具有多种不同的变体形式，如连续傅立叶变换和离散傅立叶变换。最初傅立叶分析是作为热过程的解析分析的工具被提出的。

1. **高等代数里的矩阵特征值特征向量？**

设 A 是n阶方阵，如果存在数m和非零n维[列向量](https://baike.baidu.com/item/%E5%88%97%E5%90%91%E9%87%8F" \t "_blank) x，使得 Ax=λx 成立，则称 λ 是矩阵A的一个特征值,列向量x为A对应特征值 λ 的特征向量。

1. **static 的特点？**

静态static变量/方法在类加载的过程中被初始化，在内存中只存在一份，所以可以把它当作是全局变量/方法。

java中static修饰的是静态的对象，Static关键字的特点：

1. 随着类的加载而加载。
2. 优于对象存在，随着字节码的文件存在而存在。
3. 被类的所有对象所共享.
4. 可以通过类名调用.

### 优点

1. 属于类级别的，不需要创建对象就可以直接使用.
2. 全局唯一，内存中唯一，静态变量可以唯一标识某些状态.
3. 在类加载时候初始化，常驻在内存中，调用快捷方便.

应用场景：   
1. 静态方法最适合工具类中方法的定义；比如文件操作，日期处理方法等.   
2. 静态方法适合入口方法的定义；如单例模式，因为从外部拿不到构造函数，所有定义一个静态的方法获取对象非常有必要.   
3. 静态变量适合全局变量的定义.（如布尔型静态成员变量做控制符）

### 缺点

1. 静态方法不能调用非静态的方法和变量.（**非静态方法可以任意的调用静态方法/变量**）
2. 不能使用this和super关键字（属于类级别，没有创建对象签不可用this/super）

1. **做一个项目怎么开始划分？**

做项目大概有这么几步： 1.[可行性分析](https://www.baidu.com/s?wd=%E5%8F%AF%E8%A1%8C%E6%80%A7%E5%88%86%E6%9E%90&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)和项目开发计划，做事先要计划这是很重要的； 2.需求分析，这步做得好关系着软件的质量好坏； 3.原型，这步就像做模版一样，先把样子做出来； 4.编码，上步样子是做出来了，但功能还没实现，编码来实现； 5.测试，软件有无漏洞，错误，就靠这步来测试，检查了； 6.维护，做出来的软件总需要维护更新吧。

1. **j2ee 中间件是什么？**

顾名思义，中间件就是处于中间的软件。

中间件处于操作系统软件与用户的应用软件的中间。

我国学术界一般认可的定义是中间件是指网络环境下处于操作系统、数据库等系统软件和应用软件之间的一种起连接作用的分布式软件，主要解决异构网络环境下分布式应用软件的互连与互操作问题，提供标准接口、协议，屏蔽实现细节，提高应用系统易移植性（北京大学梅宏）。

中间层一般用到的企业服务有事务服务（JTA）、目录服务（JNDI）、消息服务（JMS）、异步组件（Message-DrivenEJB）、远程调用（RMI/IIOP）。

◆JDBC，用于存取数据的Java标准API；

    ◆ JNDI(Java Namingand Directory Interface)，查找和访问各种命名和目录服务的通用、统一的标准API；

    ◆ JMS(Java MessageService)，通过消息中间件发送和接收信息的标准API；

    ◆JavaMail，发送电子邮件的标准API；

◆JavaIDL，调用CORBA服务的标准API。

1. 设计模式是什么？谈谈算法和所用的算法？

设计模式（Design Pattern）是一套被反复使用、多数人知晓的、经过分类的、代码设计经验的总结。

使用设计模式的目的：为了代码可重用性、让代码更容易被他人理解、保证代码可靠性。 设计模式使代码编写真正工程化；设计模式是[软件工程](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E5%B7%A5%E7%A8%8B/25279" \t "_blank)的基石脉络，如同大厦的结构一样。

1. 并发机制怎么实现？什么是内组件什么是外组件？