1. 什么是二叉树

二叉树是每个节点最多有两个子树的树结构。通常子树被称作“左子树”（left subtree）和“右子树”（right subtree）。二叉树常被用于实现二叉查找树和二叉堆。

1. 已知病毒特征码一百万个和文件一个，问用什么查找算法能尽快的检测出该文件是否有病毒？
2. ）hashmap基于哈希表的 Map 接口的实现
3. ）采用trie树

3.快排和插入排序那个更高效？

快速排序的平均时间复杂度为O(N\*logN)，最坏时间复杂度为O(N\*N);

而插入排序的平均和最坏都是O（N\*N）；

一般来说，规模较大的排序使用快速排序，当规模变小时，用插入排序更高效一些

1. 简单描述九宫格算法
2. 学数据结构的意义

当我们使用计算机来解决一个具体问题时，一般需要经过下列几个步骤：首先要从该具体问题抽象出一个适当的数学模型，然后设计或选择一个解此数学模型的算法，最后编出程序进行调试、测试，直至得到最终的解答。

打好“数据结构”这门课程的扎实基础，对于学习计算机专业的其他课程，如操作系统、数据库管理系统、软件工程、编译原理、人工智能、图视学等都是十分有益的。

1. 离散数学的图论和数据结构图论的相同点和不同点
2. 堆栈和堆的区别

　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。

1. 递归变成非递归需要什么

堆栈

1. 堆栈溢出是怎么回事儿

堆栈溢出就是不顾堆栈中分配的局部数据块大小，向该数据块写入了过多的数据，导致数据越界，结果覆盖了老的堆栈数据。

1. 算法的几种策略，迪杰斯特拉算法

贪心、动态规划、分治、分支限界、

1. 要得到文件的后N行，需要什么数据结构实现
2. 链表 使用快慢指针、使用循环链表
3. 栈
4. 数据库中B+树和B-树的区别

1 .B-树定义

B-树是一种平衡的多路查找树，它在文件系统中很有用。

定义：一棵m 阶的B-树，或者为空树，或为满足下列特性的m 叉树：

⑴树中每个结点至多有m 棵子树；

⑵若根结点不是叶子结点，则至少有两棵子树；

⑶除根结点之外的所有非终端结点至少有[m/2] 棵子树；

⑸所有的叶子结点都出现在同一层次上，并且不带信息

B-树主要应用在文件系统

为了将大型数据库文件存储在硬盘上 以减少访问硬盘次数为目的 在此提出了一种平衡多路查找树——B-树结构 由其性能分析可知它的检索效率是相当高的 为了提高 B-树性能’还有很多种B-树的变型，力图对B-树进行改进

B+树是应文件系统所需而产生的一种B-树的变形树。一棵m 阶的B+树和m 阶的B-

树的差异在于：

⑴有n 棵子树的结点中含有n 个关键码；

⑵所有的叶子结点中包含了全部关键码的信息，及指向含有这些关键码记录的指针，且

叶子结点本身依关键码的大小自小而大的顺序链接。

⑶所有的非终端结点可以看成是索引部分，结点中仅含有其子树根结点中最大（或最小）关键码。

1）在数据库索引的应用

在数据库索引的应用中，B+树按照下列方式进行组织 ：

① 叶结点的组织方式 。B+树的查找键 是数据文件的主键 ，且索引是稠密的。也就是说 ，叶结点 中为数据文件的第一个记录设有一个键、指针对 ，该数据文件可以按主键排序，也可以不按主键排序 ；数据文件按主键排序，且 B +树是稀疏索引 ， 在叶结点中为数据文件的每一个块设有一个键、指针对 ；数据文件不按键属性排序 ，且该属性是 B +树 的查找键 ， 叶结点中为数据文件里出现的每个属性K设有一个键 、 指针对 ， 其中指针执行排序键值为 K的 记录中的第一个。

② 非叶结点 的组织方式。B+树 中的非叶结点形成 了叶结点上的一个多级稀疏索引。 每个非叶结点中至少有ceil( m/2 ) 个指针 ， 至多有 m 个指针 。

2）B+树索引的插入和删除

①在向数据库中插入新的数据时，同时也需要向数据库索引中插入相应的索引键值 ，则需要向 B+树 中插入新的键值。即上面我们提到的B-树插入算法。

②当从数据库中删除数据时，同时也需要从数据库索引中删除相应的索引键值 ，则需要从 B+树 中删 除该键值 。即B-树删除算法

为什么使用B-Tree（B+Tree）

二叉查找树进化品种的红黑树等数据结构也可以用来实现索引，但是文件系统及数据库系统普遍采用B-/+Tree作为索引结构。

　一般来说，索引本身也很大，不可能全部存储在内存中，因此索引往往以索引文件的形式存储的磁盘上。这样的话，索引查找过程中就要产生磁盘I/O消耗，相对于内存存取，I/O存取的消耗要高几个数量级，所以评价一个数据结构作为索引的优劣最重要的指标就是在查找过程中磁盘I/O操作次数的渐进复杂度。换句话说，索引的结构组织要尽量减少查找过程中磁盘I/O的存取次数。为什么使用B-/+Tree，还跟磁盘存取原理有关。

1. 什么是树？什么是图？树和图有什么区别？

树是由根结点和若干颗子树构成的

树是图的子集  
 树有一个根节点，图没有  
 树可以递归遍历，图要看情况  
 树有层次划分，图没有  
 树的非根节点必定有一个父节点，图不一定

线性表中数据元素仅有线性关系，树则具有明显的层次性，而图的复杂性要远远超过线性表和树．在图中结点之间的关系是任意的，任何两个结点都可能相关，因此图能用来解决现实世界中一些极其复杂问题．

树是一种“层次”关系，图是“网络”关系

1. 矩阵相乘的时间复杂度是多少？

一般来说，最朴素的方法是O（N^3），最为优化的解法,是2014年秋由Fran?ois Le Gall简化的斯坦福方法，能够达到时间复杂度O(n2.3728639)。据某论文提出的猜测模型，最优可能是2.36

1. 现在有一未知大小的文件，里面是单词的集合，现要将文件读入内存，问采用什么存储结构较好？