```
概论 💿
             线性表 6
             有限制的线性表 •
             串 •
             树◎
             图 0
             查找 🏻
                                   直接插入排序 ⊙ O(n^2), 稳定, 顺序存储, 链式存储
                                   が O(n^2), 稳定, 顺序存储 利用了折半查找
                           插入排序
                                           插入排序的改进
                                           把待排序序列分组进行直接插入排序使之更有序,最后一次直接
                                   希尔排序  插入排序
                                           分组 ⊙ 初始dk=n/2取下界,如dk=3则第一组为0,3,6...
                                           O(n^2), 不稳定, 顺序
                                           两两比较相邻元素的值,如果为逆序就交换直到结束。得到递增
                                           序列, 重复n-1次
                                   冒泡排序 O(n^2),稳定,顺序存储,链式存储
                                           在待排序列表中选择一个元素pivot作为基准,通过一趟排序将待排序表划分为小于pivot和大于等于pivot的两部分。对分隔开的两部分再进行快排,重复至有序。pivot可选第一个元素
                           交换排序
                                           分治的递归的思想
                                   快速排序 😊
                                           时间平均O (nlog2(n))最坏O (n^2), 空间平均O (log2(n)),
                                           不稳定,顺序存储,链式存储。
                                              选取待排序元素 (n-i+1)中关键字最小的元素,作为有序子序列的第i个元素,直到n-1趟做完,待排序元素只剩下一个
数据结构
                                    直接选择排序 O(n^2),不稳定,时间复杂度与初始序列无关,顺序存储,链式
                                                                若L(i)<=L(2i)目L(i)<=L(2i+1), 则为小根堆
                                              n个关键字序列L[1...n]称为堆 o 若L(i)>=L(2i)且L(i)>=L(2i+1),则为大根堆
                                                    1<=i<=n/2下界
                                          堆。
                                              排序过程中将堆视为一棵完全二叉树 小根堆中,结点为以其为根的树中最小
大根堆中,结点为以其为根的树中最大
                            选择排序
                                                                                            如果孩子结点 (2i)(2i+1)小于双亲结点,则结束。
                    内部排序
                                                                                            如果孩子结点大于双亲结点,则把最大的孩子结点与双亲结点交
换,并对该孩子结点重复这个过程直到出现第一种情况或叶子结
点为止
                                                            对所有具有双亲结点含义编号从大到小(n/2下界到1)◎
                                          堆的初始化 ○ 大根堆为例 ○
                                   堆排序
                                                 输出堆顶元素
                                          堆排序 ○ 交換堆顶元素与堆尾 (未排序) 元素,重新调整堆 (未排序)
                                                重复步骤
                                          O(nlog2(n)), O(1), 不稳定, 顺序存储, 链式存储
                                          堆的插入 ○ 把新结点放置在末端然后向上调整
                                            初始序列两两有序归并为一组,然后再把归并后的组两两归并,
                            重复至归并完成序列
                                   n=序列元素个数, d=元素位数, r=每个位数的取值范围 (最大
                                   初始化r个队列,把对应位数放在对应队列中
                            基数排序 🤄
                                   每个队列按r从小到大相连
                                   改变位数, 重复以上分配队列和收集。直到有序 (d次)
                                   O(d(n+r)),稳定,不基于比较
                                             时间复杂度
                                             稳定性
             排序
                                             一趟排序的特点
                                                          n较小时(<50),采用直接插入排序或简单选择排序
                           内部排序算法的比较和应用(
                                                  元素数目 o n较大时,采用快排,堆排或者归并排序
                                                         若n很大,记录关键字位数较少且可分解,采用基数排序
                                                          当记录元素比较大,应该避免大量移动的排序算法,尽量使用链
                                                  元素大小 ○ 式存储
                                             应用 😊
                                                  关键字结构及分布 ○ 若初始基本有序,则应该使用直接插入或冒泡排序
                                                  稳定性
                                                  存储结构
                                                  辅助空间
                            当数据远远大于内存,需要一部分一部分从外存读入
                                      根据缓冲区大小把外存文件分成若干子文件,依次读入内存并使
用内部排序算法进行排序,把排序后得到的有序子文件重新写入
                            多路归并排序 对得到的归并段进行逐趟归并,直至得到整个有序文件
                                     优化 ◎ 减少归并次数的办法 <sup>□</sup> 增加归并路数 减少初始归并段个数
                                  归并时可以用来增加归并路数,加大归并效率
                                                               每个叶子节点存放各归并段再归并过程中当前参加比较的记录
                            败者树 ◎ 树形选择排序的一种变体,可以视为完全二叉树 ◎ 内部节点用来记忆左右子树中的失败者
                                                             胜利者向上继续进行比较,直到根结点
                    外部排序
                                  路数增加会降低内存内每个缓冲区的大小
                                      优化归并段的划分
                                            输入文件FI输入工作区直至装满工作区
                            置换-选择排序
                                            在工作区中选取最小的输出到FO
                                            从输入文件FI再读入
                                      步骤 o 在工作区中选取大于FO最大值的最小值输出
                                            当工作区所有关键字都小于FO的最大值时,则得到了一个有序的
                                            完整归并段
                                            清空FO,循环得到下一个归并段
                                     可以用m叉树来描述m路归并
                                     可以用mx树来相处(1945-1971)
通过置换选择排序得到初始归并段,通过m路归并排序得到一棵
带权路径长度之和为归并过程中的总读记录数
                            最佳归并树。
                                     归并树。结点存放记录的长度
                                     通过置换选择排序得到初始归并段。用哈夫曼树思想得到带权路 。
经长度最短的树,从而得到IO次数最少的归并方式 可能需要补充虚结点
```