

班级_____

实验题目： 查找排序的关键算法实现

一、实验目的

1. 能够运用高级程序设计技术实现查找排序及其关键算法
2. 理解查找排序算法的基本原理

二、实验原理

1、查找排序的基本原理

(1)、查找

- 在数据集合中寻找满足某种条件的数据元素的过程称为查找
- 查找表的定义：用于查找的数据集合称为查找表，由同一类型的数据元素组成，可以是一个数组或链表等数据类型
- 数据元素中唯一标识该元素的某个数据项的值，使用基于关键字的查找，查找结果唯一。

(2)、排序

- 排序的定义：将原本无序的序列重新排列成有序序列的过程
- 内部排序：在排序期间元素全部存放在内存中的排序
- 外部排序：在排序期间元素无法全部同时存放在内存中，必须在排序的过程中根据要求不断地在内、外存之间

2. 关键算法设计原理及性能影响因素分析

(1)、查找

- 线性结构顺序查找：从表的一端开始，顺序扫描线性表，依次将扫描到的关键字和给定值 k 比较基本思路若当前扫描的关键字与 k 相等，则查找成功若扫描结束后，仍未发现关键字等于 k 的记录，则查找失败
- 线性结构折半查找：折半查找要求线性表是有序的
- 线性结构分块查找：确定待查找的元素属于哪一块，在块内精确查找该元素

(2)、排序

- 直接插入排序：每趟将一个待排序的关键字按照其值的大小插入到已经排好的部分有序序列的适当位置上，直到所有待排关键字都被插入到有序序列中为止算法思想稳定，适用于顺序存储和链式存储
- 折半插入排序：每趟将一个待排序的关键字按照其值的大小插入到已经排好的部分有序序列的适当位置上，直到所有待排关键字都被插入到有序序列中为止。

三、实验方案设计

1. 存储方案设计

(1)、查找

- 散列结构散列表

散列函数：把查找表中的关键字映射成该关键字对应的地址的函数， $\text{Hash}(\text{key})=\text{addr}$ 。

散列表：根据关键字而直接进行访问的数据结构，建立了关键字和存储地址之间的种直接映射关系。

(2)、排序

- 简单选择排序：从头至尾顺序扫描序列，找出最小的一个关键字，和第一个关键字交换，接着从剩下的关键字中继续这种选择和交换，最终使序列有序。算法思想不稳定
- 堆排序：将一个无序序列调整为一个堆，就可以找出这个序列的最大(或最小)值，然后将找出的这个值交换到序列的最后(或最前)，这样有序序列关键字增加一个，无序序列关键字减少一个，对新的无序序列重复这样的操作，实现排序。
- 归并排序：假定待排序表含有 n 个记录，则看成是 n 个有序的子表，每个子表长度为 1，然后两两归并，得到 $n/2$ (上取整) 个长度为 2 或 1 的有序表；再两两归并，... 如此重复，直到合并成一个长度为 n 的有序表为止。

2. 算法的设计和实现

- (1)、查找

折半查找的性能分析可以用二叉判定树来衡量，平均查找长度和最大查找长度都是；二叉排序树的查找性能与数据的输入顺序有关，最好情况下的平均查找长度与折半查找相

- (2)、排序

快速排序是递归进行的，递归需要栈的辅助，需要的空间比前面几类排序算法大空间复杂度：空间复杂度： $O(\log 2n)$ 同，但最坏情况即形成单支树时，其查找长度为 $O(n)$ 。

3. 测试方案

测试用例：

3

26 78 233

31

期望输出：

0