**实 验 报 告**

|  |  |  |
| --- | --- | --- |
| **课程名称**：数据结构 | **班级**：软工21101 | **实验成绩**： |
| **实验名称**：排序的实现 | **学号**：2105006207 | **批阅教师签字：董傲霜** |
| **实验编号**：实验十二 | **姓名**：方福涛 | **实验日期：**2022 年 12 月3 日 |
| **指导教师**：董傲霜 | **组号**：wuyanzu | **实验时间**： 时 分－ 时 分 |

1. **实验设计思想**

说明典型的排序算法实现的设计思想。

**冒泡排序**

每一趟排序（包括多次两两比较和交换）找出余下的最大元素放在数组的最后，叫沉底排序更贴切。

**选择排序**

每一趟排序找出余下元素中最小的元素放在数组的最前面。

**插入排序**

每一趟排序把本次排序的元素往前插入到合适的位置。

**希尔排序**

[插入排序](https://so.csdn.net/so/search?q=%E6%8F%92%E5%85%A5%E6%8E%92%E5%BA%8F&spm=1001.2101.3001.7020)的改进；插入排序比较移动的步长是1，希尔排序中改进该步长，从大缩小到1。

**快速排序**

每趟排序把基准值放到对的位置即左边的元素都比它小，右边的元素都比它大。

**堆排序**

利用[大顶堆](https://so.csdn.net/so/search?q=%E5%A4%A7%E9%A1%B6%E5%A0%86&spm=1001.2101.3001.7020)的特点：根的值比左右子树都大先构造堆交换堆的第一个元素和最后一个元素，使最大的元素放在数组的最后，再对前面的堆进行调整，重复该操作。

**归并排序**

递归地合并两个有序数组。

**二、程序说明**

1. 给出排序数据元素存储结构定义及其属性含义的说明；

#define MAXSIZE 20//设记录不超过20个

typedef int KeyType;//设关键字为整型量（int型）

Typedef struct{//定义每个记录（数据元素）的结构

    KeyType key;//关键字

    InfoType otherinfo;//其它数据项

}RedType;

Typedef struct{//定义顺序表的结构

    RedType r[MAXSIZE+1];//存储顺序表的向量

    //r[0]一般作哨兵或缓冲区

    int length;//顺序表的长度

}SqList;

1. 给出你设计的相关排序函数的说明（函数，参数，执行的结果）

    //=====================================================

    //主系统

void ShowinStructions();

//输入数据

void InputData(int \*arr,int\* number);

    //输出数据

    void OutputData(int\* arr, int\* number);

    void swap(int\* a, int\* b); //数值交换

    //=====================================================

    //排序系统

    void showSortStruct();

    void DataSort();

    //冒泡排序

    void popo(int\* x, int n);

    //选择排序

    void selectSort(int\* arr, int num);

    //插入排序

    void insertSort(int\* arr, int num);

    //希尔排序

    void shellSort(int\* arr, int num);

    //归并排序

    void mergeSort(int\* arr, int num);

    //快速排序

    void quicksort(int left, int right);

    //堆排序

    void heap(int\* x, int n);

    //基数排序

    void rads(int\* x, int n, int d, int k);

    //=====================================================

    //查找系统

    void showFindStruct();

    void DataFind();

**三、实验环境**

Window10、DEV-C++

**四、实验过程分析**

实验调试过程中的问题及改正的事件举例说明（只要调程序就不可能没有错误，现象、原因、修正方法，可以截图但不能过多）。

插入排序——稳定排序

（1） 少了对数据索引范围的检查

（2） 同堆中siftUp siftDown一样，循环外的赋值出错了

希尔排序

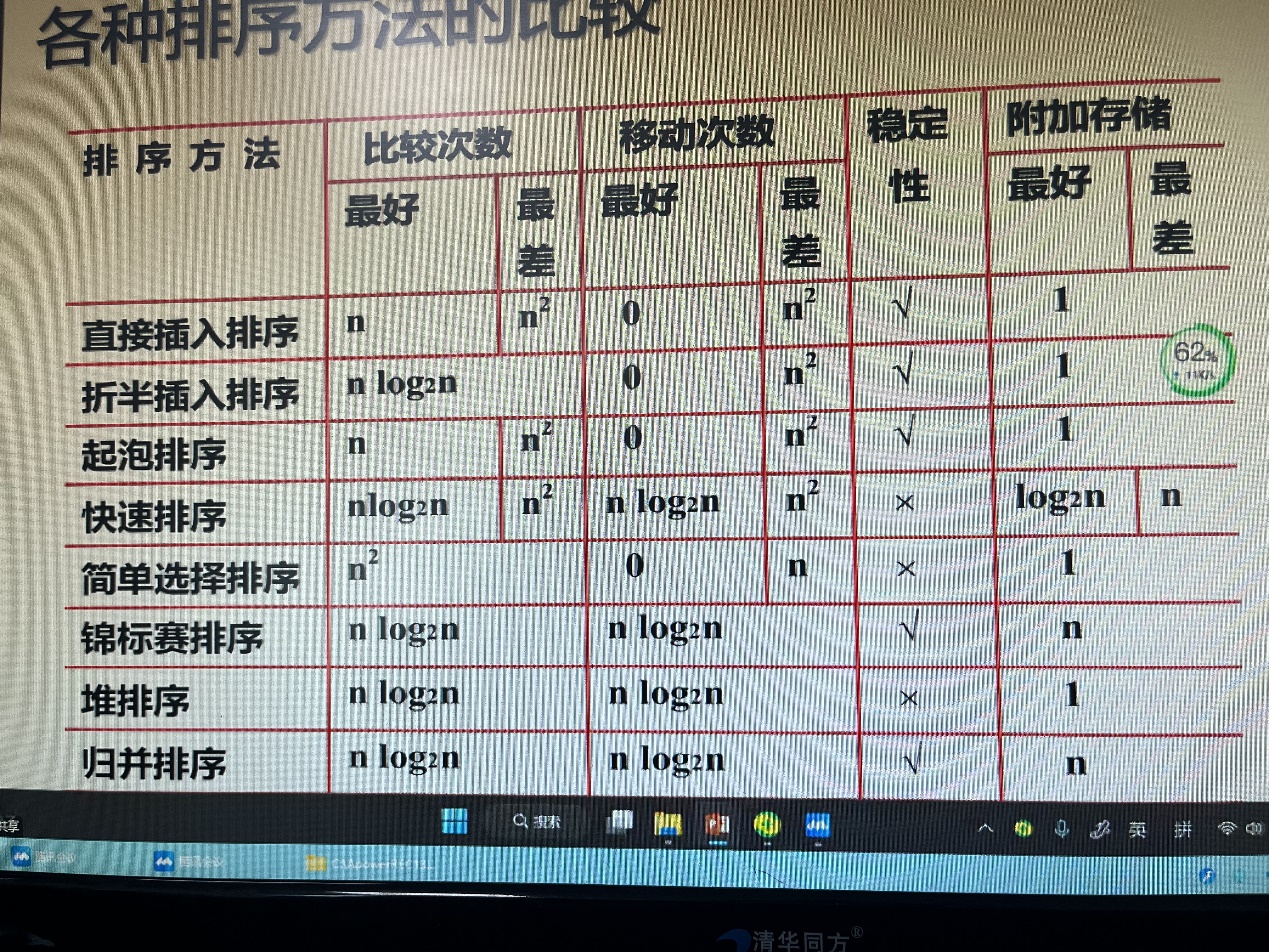
1. while循环常犯的错误，三要素中忘了更新推进循环的变量

快速排序

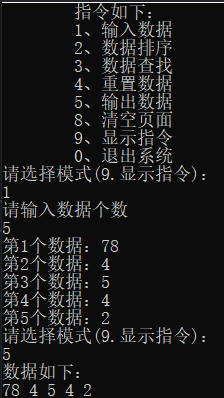
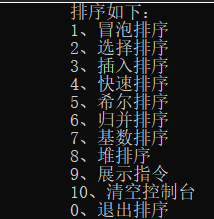
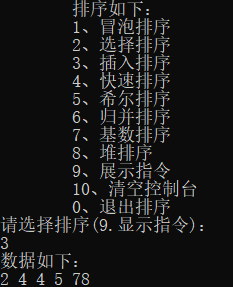
1. 递归算法，缺少终止条件
2. for循环中i变量的初始状态不要想当然写成i = 0

**五、实验结果总结**

1、阐述实现的各种排序算法的性能分析。



2、给出你的测试方法和测试截图（不能超过三个截图）；



**六、附录**

1. 意见和建议（没有可不写）。

2. 思考题：

回答以下问题：

1. 举例说明快速排序算法的最好、最坏情况下的性能。
2. 说明稳定的排序算法的应用领域。

稳定的：

如果存在多个具有相同排序码的记录，经过排序后，这些记录的相对次序仍然保持不变，则这种排序算法称为稳定的。  
    插入排序、冒泡排序、归并排序、分配排序（桶式、基数）都是稳定的排序算法。

**七、打分表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目 | 考核点 | 分数 | **得分** | 备注 |
| 程序 | 逻辑是否正确  程序可读性  创新点 | 50 | **0** |  |
| 报告完整性 | 实验过程阐述是否完整  测试数据设计是否合理  运行结果是否正确 | 40 | **0** |  |
| 调试问题及解决方法 | 是否对调试过程问题进行阐述 | 5 | **0** |  |
| 思考题目 | 回答是否正确 | 5 | **0** |  |
| 合计 |  | 100 | **0** |  |