**数据结构**

2022

**实 验 报 告**

|  |  |
| --- | --- |
| 实验项目名称： | 数据结构实验九 |
| 班级： | 21级9班 |
| 学号 | 2021302121185 |
| 姓名： | 陈聪睿 |
| 指导教师： | 董红斌 |
| 实验时间： | 2022.5.26 |

**实验一： 实现快速排序算法**

**一、实验要求**



****

**二、实验环境**

**Visual studio 2019/ Windows平台**

**三、实验步骤及思路**

（1）题目分析与主要步骤

根据题目提示可知，需要创建一个快速排序算法。

其算法描述如下：

在待排序的n 个元素

中任取一个元素（通常取第一个元素）作为基准，把该元素放入适当位置后，数据序列被此元素划分成两部分。所有关键字比该元素关键字小的元素放置在前一部分，所有比它大的元索放置在后一部分，并把该元素排在这两部分的中间，这个过程称为一趟快速排序，即一趟划分。

之后对产生的两个部分分别重复上述过程，直至每部分内只有一个元素或空为止。

简而言之，每趟使表的第一个元素放入适当位置，将表一分为二，对子表按递归方式继续这种划分， 直至划分的子表的长为1或0 。

（2）实验具体步骤截图

测试数据：题目已给出，结果如下图。

图形用户界面, 文本

描述已自动生成

**四、实验结果及分析**

可以看出已经按照需要打印出了正确结果。

**五、总结**

根据对代码进行分析可知，快速排序最好的时间复杂度为O(nlog2n)最坏的时间复杂度为O(n2), 平均时间复杂度为O(nlog2 n), 也就是说，算法的平均时间复杂度接近最好情况。

当初始数据序列为正序或者反序时，显然呈现出最坏的情况。如果初始数据序列是随机的， 每次可以划分为两个长度差不多相同的子区间， 会呈现出最好的情况。

同时也可以看出，快速排序算法是一种不稳定的排序方法。两个相同的关键字的相对位置可能会发生对调。

**实验二： 实现堆排序算法**

**一、实验要求**



****

**二、实验环境**

**Visual studio 2019/ Windows平台**

**三、实验步骤及思路**

（1）题目分析与主要步骤

通过题目可知，需要创建一个堆排序算法

其算法描述如下：

假如完全二叉树的根结点是R [i] , 它的左、右子树已是大根堆， 将其两个孩子的关键字R [2i]. key 、R [2i+l] . key 的最大者与R [i]. key 比较，若R[i]. k ey 较小，将其与最大孩子进行交换，这有可能破坏下一级的堆。继续采用上述方法构造下一级的堆，直到这棵完全二叉树变成一个大根堆为止。

假设对R [ low …… high]进行筛选，必须满足以R [low] 为根结点的左子树和右子树均为大根堆这一条件。

构建初始堆R [ l……n] 的过程是：对一棵完全二叉树，从最后一个分支结点开始， 反复利用上述筛选方法建堆。大者”上浮＂，小者被“筛选“下去。

在初始堆R [ l…… n]构造好以后，根结点R [ l ] 一定是最大关键字结点，将其放到排序序列的最后，也就是将堆中的根与最后一个叶子交换。由于最大元素已归位，整个待排序的元素个数减少一个。由于根结点的改变， 这n-1 个结点R [ l……n-1 ]不一定为堆，但其左子树和右子树均为堆，再调用一次sift 算法将这n - 1 个结点R [ l..n-1 ] 调整成堆，其根结点为次大的元素，将它放到排序序列的倒数第2个位置，即将堆中的根与最后一个叶子交换，待排序的元素个数变为n - 2 个，即R [ l……n - 2], 再调整，再将根结点归位，如此这样，直到完全二叉树只剩一个根为止。

（2）实验具体步骤截图

测试数据：题目已给出，结果如下图。

图片包含 箭头

描述已自动生成

**四、实验结果及分析**

可以看出已经按照需要打印出了正确结果。

**五、总结**

综上所述，堆排序的最坏时间复杂度为0(nlog2n) 。堆排序的平均性能分析较难，但实验研究表明，它较接近最坏性能。

堆排序算法的最好、最坏和平均时间复杂度都是O(nlog2n) 。

由于建初始堆所需的比较次数较多，所以堆排序不适合元素数较少的排序表。

堆排序只使用i、j 、tmp 等辅助变量，其辅助空间复杂度为0(1) 。

另外，在进行筛选时可能把后面相同关键字的元素调整到前面，所以堆排序算法是一种不稳定的排序方法。