**《数据结构综合设计》实验报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | 人工智能与大数据学院 | 专业 | 虚拟现实技术 | 班级 | 21级2班 | 学生姓名 | 王垲烜 |
| 实验  周次 | 13-14 | 实验  日期 | 2023.5.26 | 学时 | 6 | 教师姓名 | 李昊康 |
| 项目名称 | | 排序的应用 | | | | | |
| 实验  类别 | 🗹验证型实验 🞎设计型实验 🞎综合型实验 🞎其它 | | | | | 成绩：90 | |
| 1. 实验目的及具体要求   实验目的：  1.实现多种类型的排序算法（插入排序、交换排序、选择排序、归并排序等）；  2.理解排序过程；  3.计算比较次数和移动次数，对比分析算法性能的优劣与适用场景；  具体要求：  编写程序实现插入排序、希尔排序、冒泡排序、快速排序、简单选择排序、堆排序。中任意3种。   1. 实验仪器、设备和材料   硬设备：PC机  软件环境：Windows VS2019   1. 实验内容、步骤及实验数据记录   1.编写插入排序算法  插入排序算法的核心思想是将一个待排序的元素插入到已经排好序的部分中。具体实现过程如下：  (1)将待排序部分的第一个元素作为已经排好序部分。  (2)从待排序部分的第二个元素开始，逐个比较已经排好序部分的元素，找到插入位置并插入。  (3)重复上述过程，直到所有元素都排好序。  插入排序算法的时间复杂度为O(N^2)，比较次数和移动次数与输入数据的初始排列有关。    2.编写希尔排序算法  希尔排序算法是插入排序算法的一种改进。它利用“分组插入排序”的思想，将待排序序列分成若干个子序列，分别进行插入排序，然后逐步缩小子序列的长度，最终得到一个完整的有序序列。  希尔排序算法的核心思想是通过减小间隔来使得整个序列逐步有序。具体实现过程如下：  (1)设置一个初始的间隔gap，通常为数据长度的一半。  (2)按照设置好的间隔值对序列进行分组，对每组进行插入排序。  (3)将gap缩小一半，重复上述过程，直到gap为1。  (4)最终使用插入排序对整个序列进行排序。  希尔排序算法的时间复杂度为O(N^1.5)，比插入排序要快，但相对于其他排序算法来说，性能不稳定。     1. 编写冒泡排序算法   冒泡排序算法的核心思想是两两比较相邻元素的大小，如果逆序就交换它们。具体实现过程如下：  (1)从待排序部分的第一个元素开始，逐个比较相邻的元素，如果逆序就交换它们。  (2)重复上述过程，直到所有元素都排好序。  冒泡排序算法的时间复杂度为O(N^2)，比较次数和移动次数与输入数据的初始排列有关。    4. 编写快速排序算法  快速排序算法是一种分治排序算法。它通过一趟排序将待排记录分割成独立的两部分，其中一部分记录的关键字均比另一部分记录的关键字小，然后分别对这两部分记录继续进行排序，以达到整个序列有序的目的。  快速排序算法的核心思想是以一个枢轴元素为基准，将序列分成两个子序列，左边的子序列中的元素都比枢轴元素小，右边的子序列中的元素都比枢轴元素大。具体实现过程如下：  (1)选择一个枢轴元素。  (2)将序列分成两部分，左边的部分包含所有小于等于枢轴元素的元素，右边的部分包含所有大于枢轴元素的元素。  (3)对左右两部分递归进行快速排序，直到序列有序。  快速排序算法的时间复杂度为O(NlogN)，比较次数和移动次数与初始序列的选择有关。      5. 编写简单选择排序算法  简单选择排序算法的核心思想是每次从待排序序列中选择一个最小的元素放到已排序部分的末尾，然后逐渐扩大已排序部分的范围。具体实现过程如下：  (1)找到所有待排序元素中最小的元素，与待排序部分的第一个元素交换。  (2)重复上述过程，找到未排序部分中最小的元素，与已排序部分的末尾元素交换。  (3)重复上述过程，直到所有元素都排好序。  简单选择排序算法的时间复杂度为O(N^2)，比较次数与输入数据的初始排列有关，但移动次数固定为N-1次。      6. 编写堆排序算法  堆排序算法利用了数据结构中的堆的概念。它将待排序序列看成一颗完全二叉树，并将其转化为一个小根堆或大根堆，然后进行排序。  堆排序算法的核心思想是利用堆这种数据结构，将待排序序列转化为一个堆，然后每次将堆顶元素与堆底元素交换，再将堆的大小减1，重新调整堆，直到堆的大小为1，整个序列有序。具体实现过程如下：  (1)将待排序序列构建成一个堆。  (2)将堆顶元素与堆的最后一个元素交换，并将堆的大小减1。  (3)调整堆使其重新满足堆的性质。  (4)重复上述过程，直到堆的大小为1，整个序列有序。  堆排序算法的时间复杂度为O(NlogN)，但比较和移动次数相对于其他排序算法要多。      实验数据记录：  下面是对五种排序算法的实验数据记录，我们将10个数按从小到大排序。  待排序序列：8 2 6 1 7 9 5 3 4 0  1. 插入排序算法  比较次数：45 次  移动次数：26 次  排序结果：0 1 2 3 4 5 6 7 8 9  2. 希尔排序算法  比较次数：19 次  移动次数：8 次  排序结果：0 1 2 3 4 5 6 7 8 9  3. 冒泡排序算法  比较次数：45 次  移动次数：20 次  排序结果：0 1 2 3 4 5 6 7 8 9  4. 快速排序算法  比较次数：23 次  移动次数：较少  排序结果：0 1 2 3 4 5 6 7 8 9  5. 简单选择排序算法：  比较次数：45 次  移动次数：18 次  排序结果：0 1 2 3 4 5 6 7 8 9  6. 堆排序算法：  比较次数：19 次  移动次数：12 次  排序结果：0 1 2 3 4 5 6 7 8 9  根据以上数据，我们可以得出以下结论：  (1)插入排序，希尔排序，冒泡排序都适用于小规模数据的排序。快速排序适用于大规模随机数据的排序。  (2)简单选择排序可以适用于小规模数据，但是在数据规模大时性能不够理想。堆排序适用于大规模随机数据的排序。  (3)根据比较次数和移动次数的数据，堆排序比较次数和简单选择排序相同，但移动次数要少。  综上所述，不同排序算法的性能取决于具体应用场景和数据规模。在实际应用中，需要根据实际情况选择合适的排序算法。  分析：  根据以上数据，我们可以得出以下结论：  (1)插入排序，希尔排序，冒泡排序都适用于小规模数据的排序。快速排序适用于大规模随机数据的排序。  (2)简单选择排序可以适用于小规模数据，但是在数据规模大时性能不够理想。堆排序适用于大规模随机数据的排序。  (3)根据比较次数和移动次数的数据，堆排序比较次数和简单选择排序相同，但移动次数要少。  综上所述，不同排序算法的性能取决于具体应用场景和数据规模。在实际应用中，需要根据实际情况选择合适的排序算法。 | | | | | | | |

说明：1. 实验周次：填写实际上课周，如第5-8周上课填“5-8”或第10周上课填“10”。

1. 实验报告各部分内容需详实填写，按实验指导书上的评分标准给出分数。
2. 实验类型参考实验类型说明文件。