**《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3班 | 学生姓名 | 郝紫桐 |
| 实验  周次 | 13-14 | 实验  日期 | 2023.5.25 | 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：93 | |
| 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   #include <iostream>  using namespace std;  const int MAX\_SIZE = 8;  void insertSort(int\* arry, int n)  {  int i, j, temp;  for(i = 1; i < n; i++)  {  if(arry[i] < arry[i-1]) //反序时  {  temp = arry[i];  j = i-1;  do  {  arry[j+1] = arry[j];  j--;  }while(j>=0 && arry[j] > temp);  arry[j+1] = temp;  }  }  }  void bubbleSort(int\* arry, int n)  {  int i, j;  bool exchange;  for(i = 0; i < n-1; i++)  {  exchange = false;  for(j = n-1; j>i; j--)  {  if(arry[j] < arry[j-1]) {  swap(arry[j], arry[j-1]);  exchange = true;  }  }  if(!exchange)  {  return ;  }  }  }  int partition(int\* arry, int s, int t)  {  int i = s;  int j = t;  int temp = arry[i];  while( i < j)  {  while(j > i && arry[j] >= temp)  {  j--;  }  arry[i] = arry[j];  while(i < j && arry[i] <= temp)  {  i++;  }  arry[j] = arry[i];  }  arry[i] = temp;  return i;  }  void quickSort(int\* arry, int s, int t)  {  int i;  if(s < t)  {  i = partition(arry, s, t);  quickSort(arry, s, i-1);  quickSort(arry, i+1, t);  }  }  void selectSort(int\* arry, int n)  {  int i, j, k;  for(i = 0; i < n-1; i++)  {  k = i;  for(j = i+1; j < n; j++)  {  if(arry[j] < arry[k])  {  k = j;  }  }  if(k != i)  {  swap(arry[i], arry[k]);  }  }  }  void printArray(int\* arry, int n)  {  for(int i = 0; i < n; i++)  {  cout<<arry[i]<<" ";  }  cout<<endl;  }  int main()  {  int iArray[MAX\_SIZE]= {49,38,65,97,76,13,27,49};  insertSort(iArray, MAX\_SIZE);  printArray(iArray, MAX\_SIZE);  int jArray[MAX\_SIZE]= {49,38,65,97,76,13,27,49};  bubbleSort(jArray, MAX\_SIZE);  printArray(jArray, MAX\_SIZE);  int kArray[MAX\_SIZE]= {49,38,65,97,76,13,27,49};  quickSort(kArray, 0, MAX\_SIZE-1);  printArray(kArray, MAX\_SIZE);  int mArray[MAX\_SIZE]= {49,38,65,97,76,13,27,49};  selectSort(mArray, MAX\_SIZE);  printArray(mArray, MAX\_SIZE);  return 0;  }    分析：  利用直接插入排序，冒泡排序，快速排序和选择排序的算法实现了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。 | | | | | | | |

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。