**《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 | 缪文瑞 |
| 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   1.插入排序：  #include<iostream>  using namespace std;  void print(int a[], int n)  {  for (int j = 0; j < n; j++)  {  cout<<a[j]<< " ";  }  cout << endl;  }  void insertSort(int a[], int n)  {  for (int i = 1; i < n; i++)  {  if (a[i] < a[i - 1])  {  int j = i - 1;  int x = a[i];  while (j >= 0 && x < a[j])  {  a[j + 1] = a[j];  j--;  }  a[j + 1] = x;  }  }  }  int main()  {  int a[10] = { 21,14,52,76,17,88,91,100,110,102 };  cout << "初始序列：";  print(a, 10);  insertSort(a, 10);  cout << "排序结果：";  print(a, 10);  system("pause");  }  IMG_256  2.快速排序  #include<iostream> using namespace std; void quickSort(int a[], int, int); int main() { int array[] = { 13,75,66,15,99,12,10,23,70,89 }, k; int len = sizeof(array) / sizeof(int); cout << "初始序列:" << endl; for (k = 0; k < len; k++) cout << array[k] << ","; cout << endl; quickSort(array, 0, len - 1); cout << "排序结果:" << endl; for (k = 0; k < len; k++) cout << array[k] << ","; cout << endl; system("pause"); return 0; } void quickSort(int s[], int l, int r) { if (l < r) { int i = l, j = r, x = s[l]; while (i < j) { while (i < j && s[j] >= x) j--; if (i < j) s[i++] = s[j]; while (i < j && s[i] < x) i++; if (i < j) s[j--] = s[i]; } s[i] = x; quickSort(s, l, i - 1); quickSort(s, i + 1, r); } }  IMG_256   1. 冒泡排序   #include <iostream>  using namespace std;  void maopao(int a[], int n) {  int temp, i, j;  for (i = 0; i < n - 1; i++) {  for (j = 0; j < n - i - 1; j++) {  if (a[j] > a[j + 1]) {  temp = a[j];  a[j] = a[j + 1];  a[j + 1] = temp;  }  }  }  }  int main() {  int i, n;  int a[100];  cout << "请输入数组总数：" << endl;  cin >> n;  cout << "请输入一组整数：" << endl;  for (i = 0; i < n; i++) {  cin >> a[i];  }  maopao(a, n);  cout << "排序后的数组为：" << endl;  for (i = 0; i < n; i++) {  cout << a[i] << " ";  }  cout << endl;  }  IMG_256分析：  插入排序：优点：稳定，快；缺点：比较次数不一定，比较次数越少，插入点后的数据移动越多，特别是当数据总量庞大的时候；  冒泡排序 : 优点:比较简单,空间复杂度较低，是稳定的；缺点:时间复杂度太高，效率慢；  快速排序：优点：最快的排序方法，数据移动少；缺点：不稳定，代码易出错。  掌握了实现多种类型的排序算法。 | | | | | | | |

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。