**《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 | 窦天昊 |
| 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：92 | |
| 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<stdio.h>  #define MaxSize 50//可输入50个字符  void InsertionSort(int a[], int N) {  int tmp = 0, p, j = 0;  /\*p控制趟数，j控制移动的次数\*/  for (p = 1; p < N; p++) {  /\*需要排N-1趟\*/  tmp = a[p];  for (j = p; j > 0 && a[j - 1] > tmp; j--) {  /\*控制移动的次数，只要前面的数大于位置p的数，就往右移\*/  a[j] = a[j - 1];  }  a[j] = tmp;/\*跳出循环，合适的位置已找到，将位置p的值插入\*/  }  }  int main() {  int a[MaxSize];  int num, i;  printf("输入排序个数:");  scanf\_s("%d", &num);  printf("输出插入排序\n");  for (i = 0; i < num; i++) {  scanf\_s("%d", &a[i]);  }  InsertionSort(a, num);  printf("after ording\n");  for (i = 0; i < num; i++) {  printf("%d\t", a[i]);  }  return 0;  }  **2.希尔排序**  #include<stdio.h>  #define MaxSize 50 //可输入50个字符  /\*希尔排序\*/  void ShellSort(int a[], int N)  {  int tmp = 0, gap, i = 0, j = 0;  for (gap = N / 2; gap > 0; gap /= 2) { //缩减增量  for (i = gap; i < N; i++) {  tmp = a[i];  for (j = i; j >= gap; j -= gap) { //插入排序  if (tmp < a[j - gap])  a[j] = a[j - gap];  else  break;  }  a[j] = tmp;  }  }  }  int main() {  int a[MaxSize];  int num, i;  printf("输入排序个数:");  scanf\_s("%d", &num);  printf("输出希尔排序\n");  for (i = 0; i < num; i++) {  scanf\_s("%d", &a[i]);  }  ShellSort(a, num);  printf("after ording\n");  for (i = 0; i < num; i++) {  printf("%d\t", a[i]);  }  return 0;  }  **3.冒泡排序**  #include<stdio.h>  #define MaxSize 50 //可输入50个字符  /\*快速排序  \*left:数组第一个元素  \*right数组最后一个元素  \*/  void Quick\_Sort(int arr[], int left, int right)  {  if (left >= right) {  return; //数组中只有一个元素活不符合实际情况，直接返回  }  int key = arr[left]; //以第一个元素作为枢纽元  int begin = left;  int end = right;  while (begin != end) { //只要beign与and不相遇就一直进行扫描过程  while (begin < end && arr[end] >= key) { //从右向左扫描，直到扫描到小于key的数，循环终止（或相遇）  end--;  }  if (end > begin) {  arr[begin] = arr[end]; //end所指向的数赋给begin所指向的数  }  while (begin < end && arr[begin] <= key) { //接下来从左向右扫描，直到扫描到大于于key的数，循环终止（或相遇）  begin++;  }  if (begin < end) {  arr[end] = arr[begin];  }  }  arr[begin] = key;  Quick\_Sort(arr, left, begin - 1);//对左半部分排序  Quick\_Sort(arr, begin + 1, right);//对右半部分排序  }  int main() {  int a[MaxSize];  int num, i;  printf("输入排序个数:");  scanf\_s("%d", &num);  printf("输出冒泡排序\n");  for (i = 0; i < num; i++) {  scanf\_s("%d", &a[i]);  }  Quick\_Sort(a, 0, num - 1);  printf("after ording\n");  for (i = 0; i < num; i++) {  printf("%d\t", a[i]);  }  return 0;  }  分析： | | | | | | | |

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。