**《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级虚拟1班 | 学生姓名 | 薛睿铭 |
| 实验  周次 | 13-14 | 实验  日期 | 2023.5.29 | 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：89 | |
| 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 2. 插入排序   #include <stdio.h>  void print(int a[], int n, int i) {  printf("%d:", i);  for (int j = 0; j < 8; j++) {  printf("%d", a[j]);  }  printf("\n");  }  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 > -1 && x < a[j]) {  a[j + 1] = a[j];  j--;  }  a[j + 1] = x;  }  print(a, n, i);  }  }  int main() {  int a[8] = { 3,1,7,5,2,4,9,6 };  InsertSort(a, 8);  return 0;  }  q   1. 希尔排序   #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  #include <stdlib.h>  #include <time.h>  #include <sys/timeb.h>  #define MAXSIZE 10  void Swap(int\* a, int\* b)  {  int temp = \*a;  \*a = \*b;  \*b = temp;  }  void InsertSort\_Up(int\* arr, int length)  {  for (int i = 1; i < length; i++)  {  int temp = arr[i];  int j;  for (j = i - 1; j >= 0 && arr[j] > temp; j--)  {  arr[j + 1] = arr[j];  }  arr[j + 1] = temp;  }  return;  }  void ShellSort\_Up(int\* arr, int length)  {  int increase = length;  int i, j, k, temp;  do  {  increase = increase / 3 + 1;    for (i = 0; i < increase; i++)  {  for (j = i + increase; j < length; j += increase)  {  temp = arr[j];  for (k = j - increase; k >= 0 && temp < arr[k]; k -= increase)  {  arr[k + increase] = arr[k];  }  arr[k + increase] = temp;  }  }  } while (increase > 1);  }  void ShellSort\_Down(int\* arr, int length)  {  int increase = length;  int i, j, k, temp;  do  {  increase = increase / 3 + 1;    for (i = 0; i < increase; i++)  {    for (j = i + increase; j < length; j += increase)  {  temp = arr[j];  for (k = j - increase; k >= 0 && temp > arr[k]; k -= increase)  {  arr[k + increase] = arr[k];  }  arr[k + increase] = temp;  }  }  } while (increase > 1);  }  void PrintArr(int\* arr, int length)  {  for (int i = 0; i < length; i++)  {  printf("%d ", arr[i]);  }  printf("\n");  return;  }  long GetSysTime()  {  struct timeb tb;  ftime(&tb);  return tb.time \* 1000 + tb.millitm;  }  int main(int argc, char\* argv[])  {  srand((size\_t)time(NULL));  int arr[MAXSIZE] = { 0 };  int arr2[MAXSIZE] = { 0 };  for (int i = 0; i < MAXSIZE; i++)  {  int num = rand() % MAXSIZE;  arr[i] = num;  arr2[i] = num;  }  printf("排序前:\n");  PrintArr(arr, MAXSIZE);  printf("希尔排序升序:\n");  ShellSort\_Up(arr, MAXSIZE);  PrintArr(arr, MAXSIZE);  printf("希尔排序降序:\n");  ShellSort\_Down(arr, MAXSIZE);  PrintArr(arr, MAXSIZE);  return 0;  }  r   1. 冒泡排序   #include<stdio.h>  void main()  {  int n[10] = { 25,35,68,79,21,13,98,7,16,62 };  int i, j, k, temp;  for (i = 1; i <= 9; i++)  {  for (j = 0; j <= 9 - i; j++)  {  if (n[j] > n[j + 1])  {  temp = n[j];  n[j] = n[j + 1];  n[j + 1] = temp;  }  }  printf("第%d趟排序完成后的数据排序:\n", i);  for (k = 0; k < 10; k++)  printf("%-4d", n[i]);  printf("\n");  }  printf("排序过后的数顺序:\n");  for (i = 0; i < 10; i++)  printf("%-4d", n[i]);  printf("\n");  }  uu  分析：  学习掌握了各种排序过程，并实现了多种类型的排序算法。  掌握了排序的概念，能够对比各种算法的优劣及使用场景。 | | | | | | | |

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。