**《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 | 王怡佳 |
| 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：95 | |
| 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  运行代码：  #include<stdio.h>  void print(int a[], int n)  {  for (int i = 0; i < n; i++)  {  printf("%d ", a[i]);  }  printf("\n");  }    void insertSort(int a[], int n)  {  int i, j;  for ( i = 1; i < n; i++)  {  int key = a[i];  //注意有等号则算法就不稳定了  for ( j = i - 1; j >= 0 && a[j] > key; j--)  {  a[j + 1] = a[j];    }  a[j + 1] = key;  }  }    int main()  {  printf("插入排序：\n");  int a[] = { 2,4,6,8,10,1,3,5,7,9 };  print(a, sizeof(a) / sizeof(a[0]));  insertSort(a, 10);  print(a, sizeof(a) / sizeof(a[0]));    return 0;    }    运行结果：    2.希尔排序  运行代码:  #include<stdio.h>  void print(int a[], int n)  {  for (int i = 0; i < n; i++)  {  printf("%d ", a[i]);  }  printf("\n");  }  void shellSort(int a[], int n)  {  int gap = 3;  int i, j;  while (gap--)  {  for ( i = gap; i < n; i++)  {  int key = a[i];  for ( j = i - gap; j >= 0 && a[j] > key; j-=gap)  {  a[j + gap] = a[j];  }  a[j + gap] = key;  }  }  }    int main()  {  int a[] = { 2,4,6,8,10,1,3,5,7,9 };  printf("希尔排序：\n");  print(a, sizeof(a) / sizeof(a[0]));  shellSort(a, 10);  print(a, sizeof(a) / sizeof(a[0]));    return 0;    }    运行结果：    3.堆排序  运行代码：  #include<stdio.h>  void print(int a[], int n)  {  for (int i = 0; i < n; i++)  {  printf("%d ", a[i]);  }  printf("\n");  }  void adjustDown(int a[], int size, int n)  {  int child = n \* 2 + 1;  while (child < size)  {  if (child + 1 < size && a[child + 1] > a[child])  {  child = child + 1;  }    if (a[child] > a[n])  {  int temp = a[child];  a[child] = a[n];  a[n] = temp;  n = child;  child = child \* 2 + 1;  }  else  {  return;  }    }  }    void heapSort(int a[], int n)  {  //建堆  for (int i = (n - 2) / 2; i >= 0; i--)  {  adjustDown(a, n, i);  }    //利用堆删除思想排序  int end = n - 1;  while (end)  {  int temp = a[0];  a[0] = a[end];  a[end] = temp;    //heapSort(a, end);  //上一次end的下标刚好是下一次要排的元素个数  adjustDown(a, end, 0);  end--;  }    }    int main()  {    int a[] = { 1,3,5,7,9,2,4,6,8,10 };  printf("堆排序：\n");  print(a, sizeof(a) / sizeof(a[0]));  heapSort(a, sizeof(a) / sizeof(a[0]));  print(a, sizeof(a) / sizeof(a[0]));  return 0;      }      运行结果：    分析：通过本次学习可以学到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）；理解排序过程；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； | | | | | | | |

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。