**《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 | 王梦鑫 |
| 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：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   **插入排序**  #include <stdio.h>  void insertion\_sort(int arr[], int n) {  int i, j, key;  for (i = 1; i < n; i++) {  key = arr[i];  j = i - 1;  while (j >= 0 && arr[j] > key) {  arr[j + 1] = arr[j];  j--;  }  arr[j + 1] = key;  }  }  int main() {  int arr[] = { 8, 3, 4, 2, 12,11 };  int n = sizeof(arr) / sizeof(arr[0]);  insertion\_sort(arr, n);  printf("Sorted array: ");  for (int i = 0; i < n; i++) {  printf("%d ", arr[i]);  }  printf("\n");  return 0;  }  **冒泡排序**  #include <stdio.h>  void bubble\_sort(int arr[], int n) {  int i, j, temp;  for (i = 0; i < n - 1; i++) {  for (j = 0; j < n - i - 1; j++) {  if (arr[j] > arr[j + 1]) {  temp = arr[j];  arr[j] = arr[j + 1];  arr[j + 1] = temp;  }  }  }  }  int main() {  int arr[] = { 12, 2, 1, 16, 9, 3 };  int n = sizeof(arr) / sizeof(arr[0]);  bubble\_sort(arr, n);  printf("Sorted array: ");  for (int i = 0; i < n; i++) {  printf("%d ", arr[i]);  }  printf("\n");  return 0;  }    **快速排序**  #include <stdio.h>  void swap(int\* a, int\* b) {  int temp = \*a;  \*a = \*b;  \*b = temp;  }  int partition(int arr[], int low, int high) {  int pivot = arr[high];  int i = low - 1;  for (int j = low; j <= high - 1; j++) {  if (arr[j] < pivot) {  i++;  swap(&arr[i], &arr[j]);  }  }  swap(&arr[i + 1], &arr[high]);  return i + 1;  }  void quick\_sort(int arr[], int low, int high) {  if (low < high) {  int pi = partition(arr, low, high);  quick\_sort(arr, low, pi - 1);  quick\_sort(arr, pi + 1, high);  }  }  int main() {  int arr[] = { 15, 12, 14, 16, 21, 13 };  int n = sizeof(arr) / sizeof(arr[0]);  quick\_sort(arr, 0, n - 1);  printf("Sorted array: ");  for (int i = 0; i < n; i++) {  printf("%d ", arr[i]);  }  printf("\n");  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。