**实 验 报 告 （ 4 ）**

|  |  |
| --- | --- |
| **实验名称**：排序算法 | **实验地点**：信息楼318 |
| **所使用的工具软件及环境：Win10, Java** | |
| **一、实验目的：**  理解各类排序算法的设计思想，灵活应用排序方法解决实际问题。 | |
| **二、实验内容描述：**（填写题目内容及输入输出要求）  1. 使用快速排序算法，对下列给定数据进行排序。  数据：6 1 2 7 9 3 4 5 10 8  要求：在实验报告中详细说明快速排序的整个过程。 | |
| **三、程序运行结果（说明设计思路，解释使用的数据结构，计算时间复杂度）**  **第1题**   1. 实验运行结果截图      1. 数据结构   采用数组存储输入，利用quicksort递归函数进行排列。     1. 设计思路   根据输入设置基准位temp，从数组第一个元素arr[low]和最后一个元arr[high]与基准位temp进行比较，保证值小于基准位的位于数组前端，值大于基准位的位于数组后端，满足条件则向中间移动，如果不满足条件，则将arr[low]与arr[high]进行交换。并递归调用左半数组、右半数组。  public static void quickSort(int[] arr,int low,int high){  int i,j,temp,t;  if(low>high){  return;  }  i=low;  j=high;  //temp就是基准位  temp = arr[low];  while (i<j) {  //先看右边，依次往左递减  while (temp<=arr[j]&&i<j) {  j--;  }  //再看左边，依次往右递增  while (temp>=arr[i]&&i<j) {  i++;  }  //如果满足条件则交换  if (i<j) {  t = arr[j];  arr[j] = arr[i];  arr[i] = t;  }  }  //最后将基准为与i和j相等位置的数字交换  //如果是偶数个数字，出循环时也还是i=j，  //因为里面有两个while：(temp<=arr[j]&&i<j) 、(temp>=arr[i]&&i<j)  arr[low] = arr[i];  arr[i] = temp;  //递归调用左半数组  quickSort(arr, low, j-1);  //递归调用右半数组  quickSort(arr, j+1, high);  }   1. 时间复杂度   执行quicksort递归函数的时间复杂度为O(nlogn)。 | |

2023年6月6日