# 北京邮电大学软件学院

# 2017-2018学年第1学期实验报告

**课程名称： 算法与数据结构**

**实验名称： 排序**

**实验完成人：**

**姓名：**\_\_徐朗朗\_\_\_\_**学号：**\_2016211965\_\_\_**成绩：**\_\_\_\_\_\_\_\_

**指导教师：**\_\_\_\_\_\_\_\_\_李晶\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**日 期： 2018 年 1 月 11 日**

1. **实验目的**

本次实验旨在集中对几个专门的问题作较为深入的探讨和理解，不强调对某些特定的编程技术的训练。

1. **实验内容**
2. **内部排序算法比较**

**[问题描述]**

　　各种内部排序算法的时间复杂度分析结果只给出了算法执行时间的阶，或大概执行时间。试通过随机的数据比较各算法的关键字比较次数和关键字移动次数，以取得直观感受。

**[基本要求]**

　　（1） 对以下9种常用的内部排序算法进行比较：直接插入排序；折半折入排序；二路插入排序；希尔排序；起泡排序；快速排序；简单选择排序；堆排序；归并排序。

　　（2） 待排序表的表长不少于100；其中的数据要用伪随机数产生程序产生；至少要用5组不同的输入数据作比较；比较的指标为有关键字参加的比较次数和关键字移动次数（关键字交换计为3次移动）。

**[测试数据]**

　　由随机产生器决定。

**[实现提示]**

　　主要工作是设法在程序中适当的地方插入计数操作。程序还可以包括计算几组数据得出结果波动大小的解释。注意分块调试的方法。

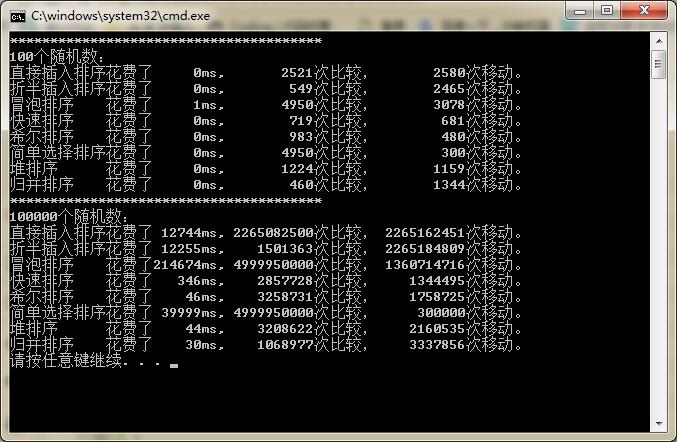
1. **实验环境**

VS2017

1. **实验结果**

完成了全部必做内容

1. **附录**
2. 程序运行结果展示



1. 性能分析

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 算法名称 | 直接插入排序 | 简单选择排序 | 冒泡排序 | 折半插入排序 | 快速排序 | 希尔排序 | 堆排序 | 归并排序 |
| 时间复杂度 | O(n²) | O(n²) | O(n²) | O(n²) | O（） | O() | O（） | O（） |
| 空间复杂度 | O(1) | O(1) | O(1) | O(1) | O（ | O(1) | O(1) | O(n) |
| 稳定性 | 稳定 | 不稳定 | 稳定 | 稳定 | 不稳定 | 不稳定 | 不稳定 | 稳定 |

1. 部分代码展示

//快速排序

int QuickSort(int a[], int n, int left, int right) {

int i, j, pivot, times = 0;

if (left >= right) {

// print(a, n);

return times;

}

i = left;

j = right;

pivot = a[left];//将最左边元素作为枢纽

cmpt++;

while (i != j) {

while (a[j] >= pivot&&i < j)//从右边开始找

j--, cmpt++;

while (a[i] <= pivot&&i < j)

i++, cmpt++;

swap(a[i], a[j]);

movt += 3;

}

swap(a[left], a[i]);//将枢纽元素归位，左边小于它，右边大于它

movt += 3;

QuickSort(a, n, left, i - 1);

QuickSort(a, n, i + 1, right);

}

1. 比较次数和移动次数的计算：

如上述代码示例，在涉及数组元素比较的语句后使cmpt自增1

涉及数组赋值的语句后movt自增1，数组元素之间交换一次movt自增3