

# 浙江大学实验报告

专业：计算机科学与技术

姓名：蔡雨谦

学号：3210102466

日期：2022.06.02

课程名称：C 程序设计专题 指导老师：翁恺 成绩：\_\_\_\_\_

实验名称：作业 4：快速排序和归并排序比较

## 一、实验题目要求

比较自己实现的归并和快排的性能，注意比较性能时应使用正确的计时方法，并有一定的数据量，而且要设计数据的特性。

## 二、实验思路和过程描述

### 1、快速排序

quicksort (int a[],int l,int r) 取数组中间的元素下标为 pivot，与最左端 (a[l]) 交换。pivot 表示最后放置作为 pivot 的数的位置的下标。从第二个元素起，依此与 a[l] 比较，比 a[l] 小的与 pivot 下一个交换，pivot 右移。遍历一轮后，从第二位起以下标 pivot 为界分成比元素 pivot 小的和比元素 pivot 大的两边，此时下标 pivot 指的数是小于元素 pivot 的数的最后一个。将 a[l] (即元素 pivot) 与 a[pivot] 交换。再对 pivot 两边的数组进行 quicksort 操作，直至 l=r。

### 2、归并排序

将两个有序数组合并的函数 comb(int a[],int l,int m,int r)，：l 至 m,m+1 至 r 各为一个升序数组，通过新建数组 temp[r-l+1],依此比较两数组中元素大小，填满 temp。

归并的函数 mergesort(int a[],int l,int r)：m=(l+r)/2,对 m 两侧的数组分别进行 mergesort 操作，直至数组长度为 1 (l=r),最后 comb 左右两侧的有序数组。

### 3、计时

借助 PTA 网站例题测速。

## 三、实验代码解释

1、声明数组中间元素的下标时，用 m=l+(r-l)/2 而不用 m=(l+r)/2 的原因是 l+r 容易溢出。

```
if (l < r)
{
    int m = l + (r - l) / 2;
    mergesort(a, l, m);
    mergesort(a, m + 1, r);
    comb(a, l, m, r);
}
```

## 四、实验结果与心得

	快速排序	归并排序
无序<1000	3ms, 312kb	3ms,324kb
顺序<1000	3ms,184kb	2ms,196kb
逆序<1000	3ms,320kb	3ms,328kb
无序 10000~1000000	124ms,5932kb	119ms,7416kb
顺序 10000~1000000	72ms,3884kb	65ms,5184kb
逆序 10000~1000000	49ms,2732kb	44ms,3636kb

根据实验结果，我设计的快速排序与归并排序在排序时效率相当，归并排序稍快于快速排序。排大量数据用时都比少量数据耗时长。快速排序占用的空间小于归并排序。排顺序和逆序数据时，用时和占用空间都小于无序数据。快速排序和归并排序的时间复杂度都是  $O(n \log n)$ ，都比冒泡排序和选择排序快。