

哈尔滨工业大学(深圳)

# 《数据结构》实验报告

## 实验四

### 排序、查找及其应用

学 院: 计算机科学与技术学院  
姓 名: 金正达  
学 号: 220110515  
专 业: 计算机类  
日 期: 2023-05-27

## 一、问题分析

4001 题目要求根据输入的还钱总额、每天还钱额度及天数来确定最大还款利率，可以通过二分搜索和题目提供的检查函数来逐步缩小范围确定该利率；

4002 题目要求对学生成绩排序，优先根据加权计算出的总成绩进行排序，若总成绩相等则按照科目成绩优先级排序，若仍然相等则按学号降序排列但名次相同，对总成绩先进行稳定排序，而后根据进一步排序的要求逐个筛选排查；

4003 题目要求找出一组数中逆序对的个数，在排序过程中需要交换的即使一对逆序对，根据此思想在排序算法中加入计算即可。

## 二、详细设计

### 2.1 设计思想

4001 直接进行二分搜索，左边界定为 0，右边界定为 2，在循环中先计算中间值。由于边界值为浮点数，故在题目给出的检查函数返回值为 1 时将左边界设置为中间值，否则将右边界设置为中间值，直到左右边界相等时退出循环，返回左边界或者右边界值。对浮点数判等时判断其差的绝对值是否小于某个精度。

4002 对 student 结构添加总成绩 sum 和排名 rank 变量，直接对 student 数组操作，再将结果复制进入 rank 数组中更为方便。首先，根据输入的各科成绩和权重计算各个学生总分并储存与 student 的 sum 变量中，接着，按照 sum 的值对学生进行预排序，并按照排序结果分配 rank 值，然后，利用插入排序的思想，对于每一个学生，遍历排序在其前面的所有学生，两两比较，若相等，则按照科目优先级来排序，通过计数和遍历优先级数组来寻找当前最高优先级所对应的科目，若各科目成绩均相等，则比较其学号来排序，并且两者排名相同，最后，循环遍历将学生信息填入 rank 数组即可。

4003 题目要求高效的算法，可以使用归并排序，在归并排序中归并两个已经有序的数组时，使用了两个指针，一个指向左子数组的起始位置，一个指向右子数组的起始位置。比较左右两个指针所指元素的大小，如果左边的元素小于等于右边的元素，则将左边的元素放入合并后的数组，并将左指针向后移动一位。如果左边的元素大于右边的元素，则说明存在逆序对，此时将右边的元素放入合并后的数组，并将右指针向后移动一位。在这个过程中，如果存在逆序对，逆序对的数量就是当前左指针到左子数组末尾的元素个数。归并完成时，即可得到逆序对的数量。

## 2.2 存储结构及操作

(1) 存储结构（一般为自定义的数据类型，比如单链表，栈等。）

4001 无自定义数据类型

4002 定义了学生和排名，

```
typedef struct student {  
    int Code;//学号  
    int Score[5];//各科目成绩  
    double sum;//加权总成绩  
    int rank;//排名  
} Student;
```

```
typedef struct rank {  
    int Rank;//排名  
    int Code;//学号  
} Rank;
```

4003 无自定义数据类型

(2) 涉及的操作（一般为自定义函数，可不写过程，但要注明该函数的含义。）

4001

double BinarySearch(double x, double y, int t, double lo, double hi)

在输入边界 lo 和 hi 之间进行二分搜索，返回搜索结果。

4002

void CalcAndWriteRanks()

对输入的学生成绩排序。

4003

void merge(int\* a, int lo, int mid, int hi)

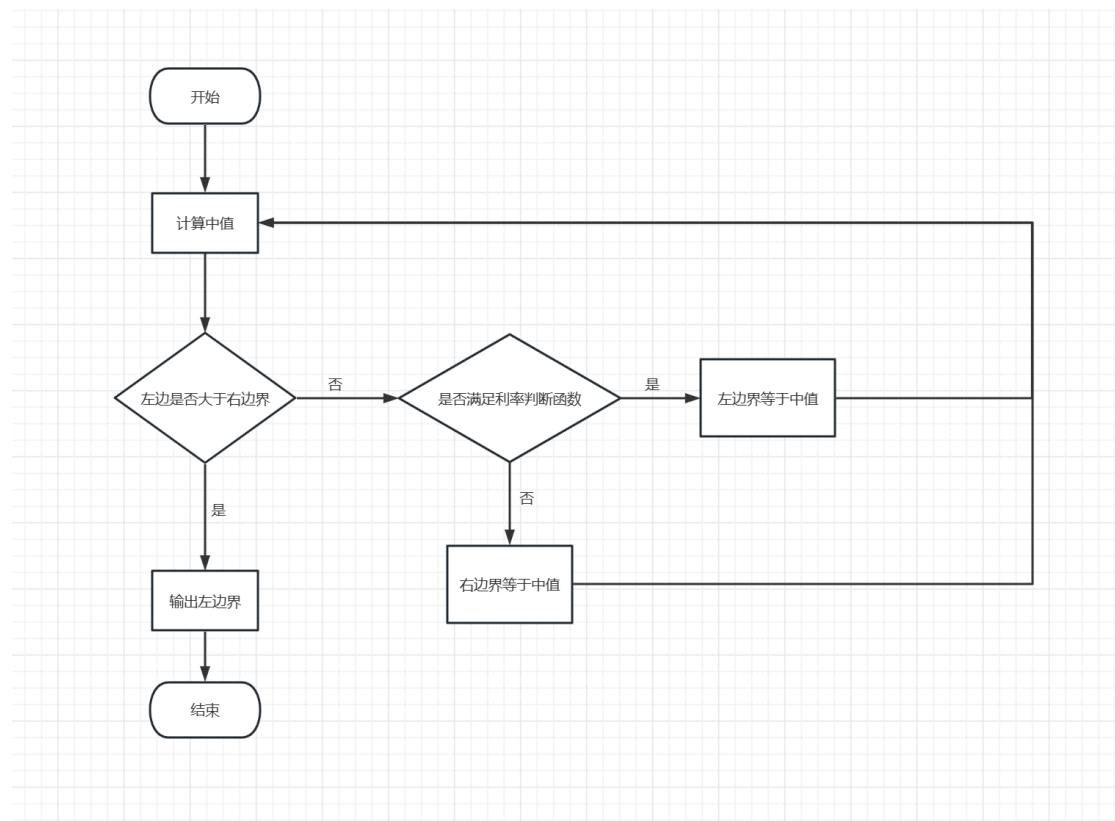
对输入数组的指定下标进行合并。

void mergeTD(int\* a, int lo, int hi)

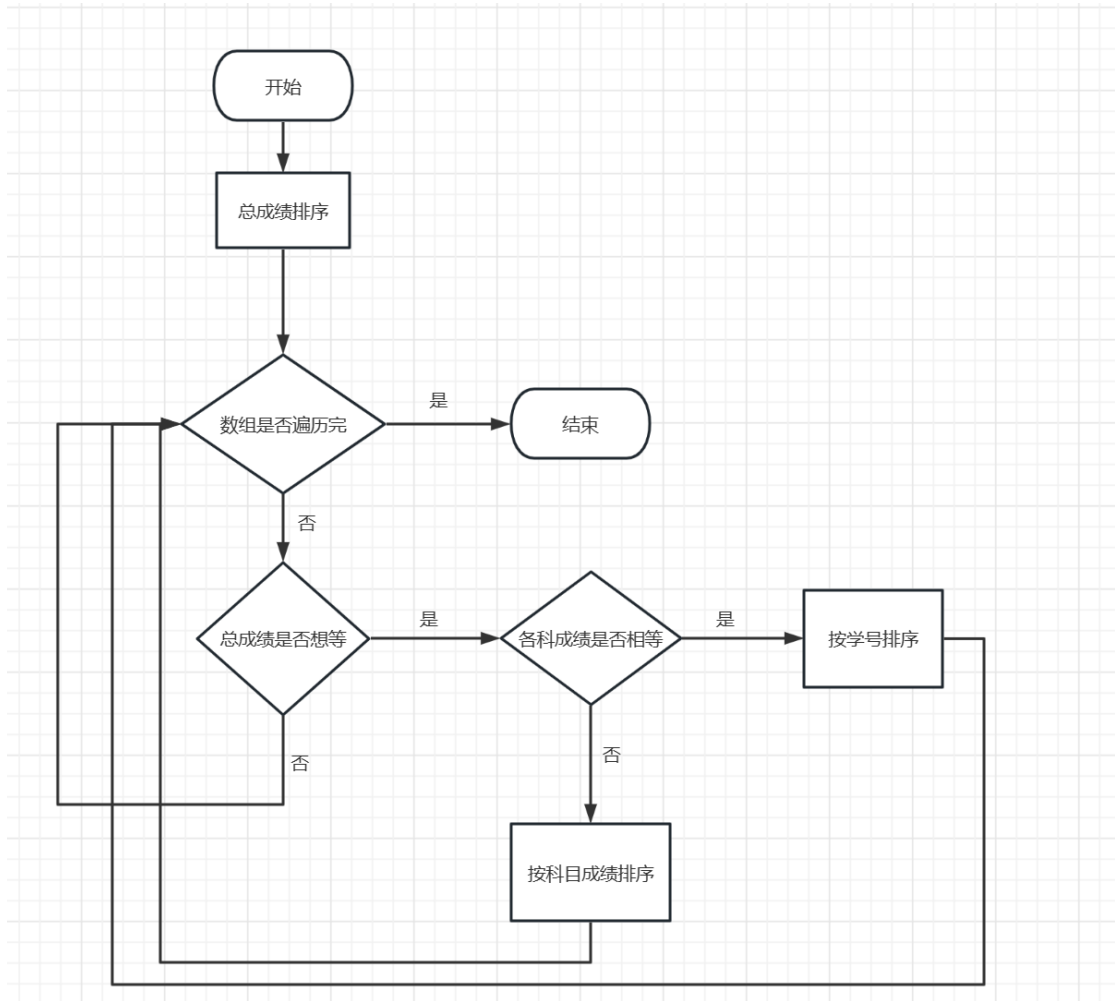
对输入数组进行归并排序。

## 2.3 程序整体流程

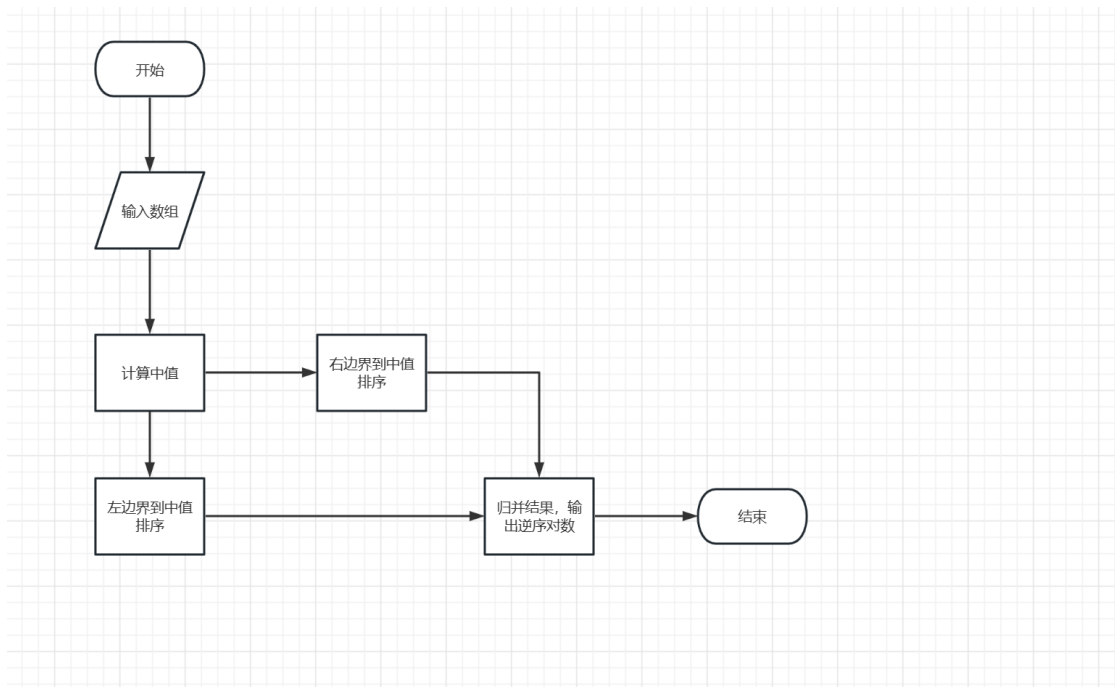
4001



4002



4003



## 二、运行结果

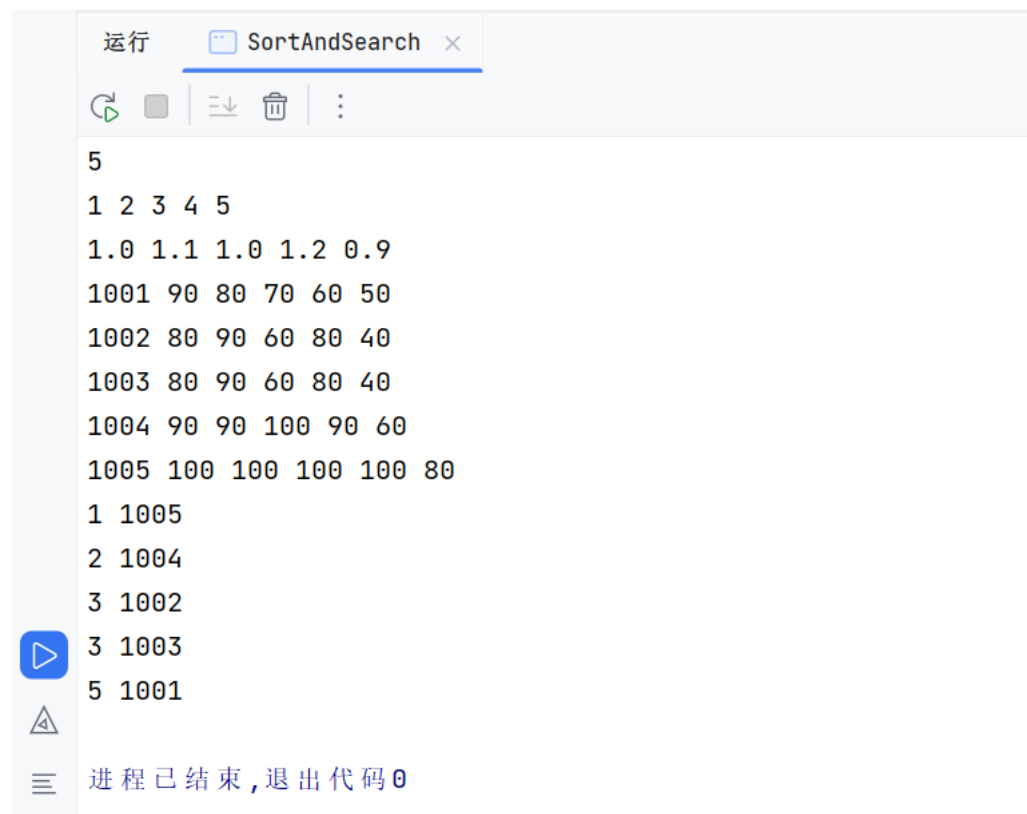
4001



```
运行  SortAndSearch x
4
1919810 114514 24
3.09%
1919810 114514 36
4.90%
1919810 114514 48
5.51%
1919810 114514 60
5.76%

进程已结束,退出代码0
```

4002



```
运行  SortAndSearch x
5
1 2 3 4 5
1.0 1.1 1.0 1.2 0.9
1001 90 80 70 60 50
1002 80 90 60 80 40
1003 80 90 60 80 40
1004 90 90 100 90 60
1005 100 100 100 100 80
1 1005
2 1004
3 1002
3 1003
5 1001

进程已结束,退出代码0
```

4003



```
运行 SortAndSearch x
6
5 4 2 6 3 1
11
进程已结束,退出代码0
```

### 三、总结

涉及的数据结构和算法：主要涉及插入排序、二分搜索和归并排序这几种算法。

遇到的问题：在 4001 中，我开始没有认识到对浮点数进行二分查找应该有不同的具体处理，仍然向对整数处理那样逐步累加，导致结果出错。在 4002 中，我对成绩相等后的排序只考虑了相邻两项，后来才意识到要按照插入排序的思想将前面的每一项和该项相比。在 4003 中，在计算每次归并有多少逆序对时花费了较多时间。

收获：经历了一些困难，我对基本的排序和查找算法更加熟悉了，提高了自己用计算机解决问题的能力。