天津大学

数据结构上机实验报告

题目:奇偶交换排序

学生姓名 王雨朦

学生学号 2016229082

学院名称 国际工程师学院

专业 计算机

时间 2016/12/25

目　　录

第一章　需求分析

1.1 原题表述

奇偶交换排序是另一种交换排序。算法的基本思想如下：排序过程通过对文件 x[i]（1≤i≤n）的若干次扫描来完成，第奇数次扫描，对所有下标为奇数的记录 x[i]与其后面的记录 x[i+1]（1≤i≤n－1）相比较，若 x[i].KEY（记录 x[i]的 key 值）＞x[i+1].KEY，则交换 x[i]和 x[i+1]的内容；第偶数次扫描，对所有下标为偶数的记录 x[i]与其后面的记录 x[i+1]（2≤i≤n－1）相比较，若 x[i].KEY＞x[i+1].KEY，则交换 x[i]和 x[i+1]之内容。一趟奇数次扫描和一趟偶数次扫描被称为一趟奇偶扫描。重复上述过程直到排序完成为止。

设计奇偶排序函数，对于任意输入元素，输出排序过程。

1.2 解决方案

1) 建立一种新的排序方式，一共有两趟。其中，奇数那趟对数组中的所有奇数项扫描；偶数那趟对序列中的所有偶数项扫描；
2) 如果 a[i]>a[i+1]，那么交换两数；
3) 如此反复，最后得到排序好的数列为止，终止条件为进行一趟排序对比时，不出现交换的情况。

第二章　概要设计

2.1 抽象数据类型

ADT RecordNode {

数据对象：D = {a_i| a_i ∈ ElemSet, i = 1,2,3,…,n}

数据关系：S = {<a_i,a_{i+1}>| i = 1,2,3,…,n}

基本操作：

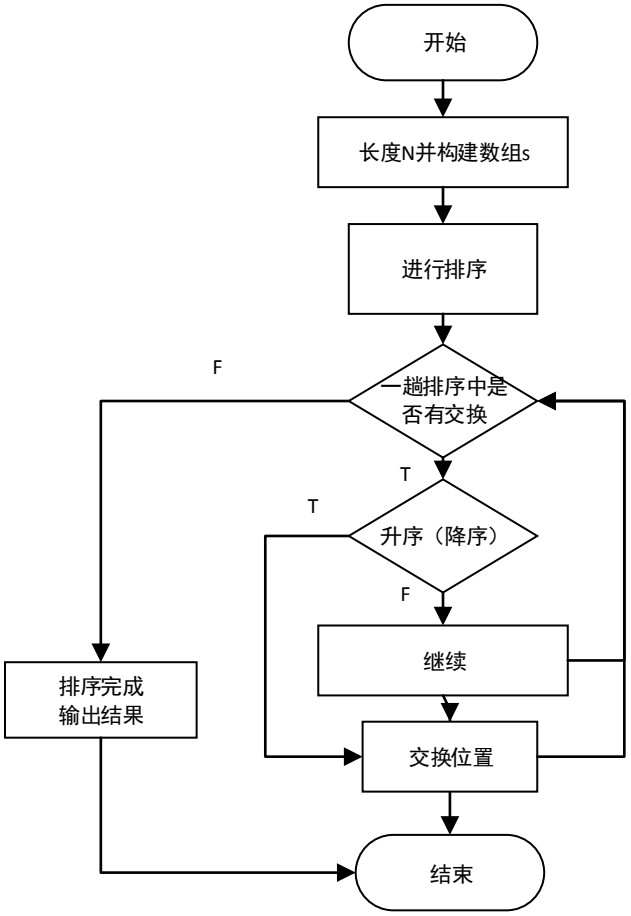
OddEvenSort(int s[], int n);

初始条件：输入的一个序列

操作条件：按照奇数和偶数趟进行排序，输出结果。

}

2.2 算法



第三章　详细设计

3.1 程序代码

```
#include <iostream>
#include <cstdlib>
using namespace std;
int OddEvenSort(int s[], int n)
{
    int i, exchange = 1, temp, num = 0;
    while(exchange != 0){
        exchange = 0;
        for(i = 1; i < n - 1; i += 2){
            if(s[i] > s[i+1])
            {
                exchange = 1;
                temp = s[i];
                s[i] = s[i+1];
                s[i+1] = temp;
            }
        }
        if(exchange){
            num++;
            exchange = 0;
            for(int w = 0; w < n; w++)
                cout<<s[w]<<" ";
            cout<<endl;
        }
        for(i = 0; i < n - 1; i += 2){
            if(s[i] > s[i+1])
            {
                exchange = 1;
                temp = s[i];
                s[i] = s[i+1];
                s[i+1] = temp;
            }
        }
        if(exchange){
            num++;
            for(int t = 0; t < n; t++)
                cout<<s[t]<<" ";
            cout<<endl;
```

```
        }
    }
    return num;
}
int main()
{
    int n;
    cout<<"请输入数组位数:";
    cin>>n;
    int s[n];
    cout<<"请依次输入数组数据："<<endl;
    for(int i = 0; i < n; i++)
    {
        int number;
        cin>>number;
        s[i] = number;
    }
    cout<<"未排序前数组数据为: "<<endl;
    for(int i = 0; i < n; i++)
        cout<<s[i]<<" ";
    cout<<endl;
    cout<<"排序过程: "<<endl;
    int sum = OddEvenSort(s, n);
    cout<<"排序后数组数据为: "<<endl;
    for(int i = 0; i < n; i++)
        cout<<s[i]<<" ";
    cout<<endl;
    cout<<"总排序趟数："<<sum<<endl;
}
```

第四章　调试分析

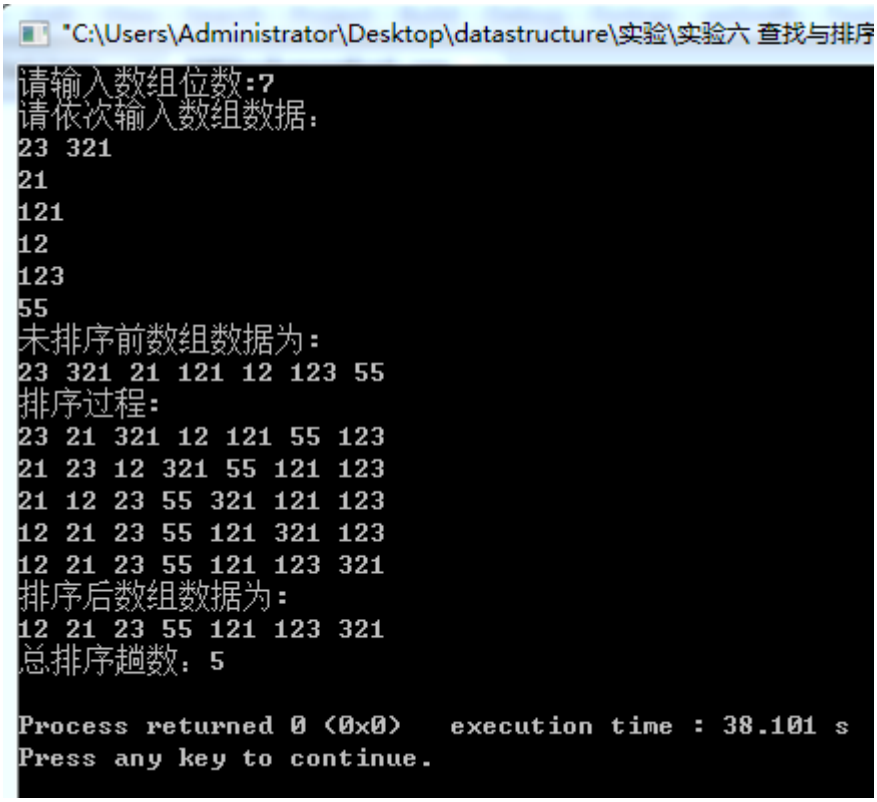
4.1 调试过程

Bug 名称	数据结构选择不合适
Bug 描述	编写程序解决问题时由于选择结构不正确导致出现麻烦
Bug 原因	没有选择正确的数据结构
Bug 解决方案	改变数据的存储方式
Bug 总结	数据结构运用不够熟练

第五章　测试结果

5.1 测试过程

测试编号	1
测试对象	排序功能
测试输入参数	请输入数组位数:7 请依次输入数组数据： 23 321 21 121 12 123 55
测试步骤	先输入数组，再查看结果
测试预期结果	未排序前数组数据为: 23 321 21 121 12 123 55 排序过程: 23 21 321 12 121 55 123 21 23 12 321 55 121 123 21 12 23 55 321 121 123 12 21 23 55 121 321 123 12 21 23 55 121 123 321 排序后数组数据为: 12 21 23 55 121 123 321 总排序趟数：5
测试输出结果	未排序前数组数据为: 23 321 21 121 12 123 55 排序过程: 23 21 321 12 121 55 123 21 23 12 321 55 121 123 21 12 23 55 321 121 123 12 21 23 55 121 321 123 12 21 23 55 121 123 321 排序后数组数据为: 12 21 23 55 121 123 321 总排序趟数：5
测试分析	程序正确


"C:\Users\Administrator\Desktop\datastructure\实验\实验六 查找与排序
请输入数组位数:7
请依次输入数组数据:
23 321
21
121
12
123
55
未排序前数组数据为:
23 321 21 121 12 123 55
排序过程:
23 21 321 12 121 55 123
21 23 12 321 55 121 123
21 12 23 55 321 121 123
12 21 23 55 121 321 123
12 21 23 55 121 123 321
排序后数组数据为:
12 21 23 55 121 123 321
总排序趟数: 5
Process returned 0 (0x0) execution time : 38.101 s
Press any key to continue.