

排序实验报告

课程名称：数据结构与算法

学 院： 计算机与信息安全学院

专业班级：19003603

学 号：1900301517

姓 名：陆洪业

报告日期：2021年 01 月 10 日

目录

**[1. 利用队列实现密码加密解密实验目的](#_Toc1831915961_WPSOffice_Level1)** **[1](#_Toc1831915961_WPSOffice_Level1)**

[1.1. 选用储存结构](#_Toc647397264_WPSOffice_Level2) [1](#_Toc647397264_WPSOffice_Level2)

[1.2. 实验代码](#_Toc1130979933_WPSOffice_Level2) [1](#_Toc1130979933_WPSOffice_Level2)

[1.3. 系统测试与结果](#_Toc1068613576_WPSOffice_Level2) [7](#_Toc1068613576_WPSOffice_Level2)

[1.4. 实验小结](#_Toc1598252537_WPSOffice_Level2) [8](#_Toc1598252537_WPSOffice_Level2)

# 排序实验目的

完成直接插入排序法\二分插入排序法\表插入排序\shell排序

## 实验代码

#include <stdio.h>

#include <stdlib.h>

/\*数据结构定义\*/

typedef int DataType;

typedef struct

{

DataType \*data; //用于存放待排序关键字的起始地址

int NUM; //待排序关键字的个数

} SortObject;

typedef struct node //用于表插入排序的数据结构

{

DataType info;

struct node \*next;

} linkObject;

//输出顺序表

void print(SortObject \*p)

{

for(int i=0;i<p->NUM;i++)

printf("%d ",p->data[i]);

printf("\n");

}

//输出链表

void printLink(linkObject \*Head)

{

linkObject \*p = Head->next ;

while(p)

{

printf("%d ",p->info);

p = p->next;

}

printf("\n");

}

/\*第一关

此处请填写代码实现递增序进行直接插入排序,

要求每趟排序后 调用print函数，输出关键字的排列情况\*/

//input 8 49 38 65 97 76 113 27 49

//output the result of listSort:

//38 49 65 97 76 113 27 49

//38 49 65 97 76 113 27 49

//38 49 65 97 76 113 27 49

//38 49 65 76 97 113 27 49

//38 49 65 76 97 113 27 49

//27 38 49 65 76 97 113 49

//27 38 49 49 65 76 97 113

void insertSort( SortObject \*Rec )

{

/\*----begin------\*/

int i, j;

DataType tmp;

DataType \* data = Rec->data;

for (i = 1; i < Rec->NUM; ++i) {

tmp = data[i];

// 腾出一个位置插入

for (j = i - 1; tmp < data[j] && j >= 0; --j)

data[j+1] = data[j];

if (j != i-1)

data[j+1] = tmp;

print (Rec);

}

/\*-----end------\*/

}

/\*第二关

此处请填写代码实现递增序进行二分插入排序,

实质是在已经有序的表中采用二分法查找插入位置

要求每趟排序后 调用print函数，输出关键字的排列情况\*/

void binInsertSort( SortObject \*Rec )

{

/\*----begin------\*/

int i, j, l, r, mid;

DataType tmp;

DataType \* data = Rec->data;

for (i = 1; i < Rec->NUM; ++i) {

tmp = data[i];

l = 0;

r = i - 1;

while (l <= r) {

mid = (l+r)/2;

/\* 最后一次mid一定会等于l,

\* 此时如果tmp小于data[l],

\* 则r向左移, tmp仍然小于data[l]

\* 如果tmp大于或等于data[l],

\* 则l向右移

\* 最终tmp一定会恰好小于data[l]

\*/

if (tmp < data[mid])

r = mid - 1;

else

l = mid + 1;

}

// 腾出一个位置插入

for (j = i - 1;j >= l; --j)

{

data[j+1] = data[j];

}

if (j != i-1)

data[j+1] = tmp;

print (Rec);

}

/\*-----end------\*/

}

/\*第三关

此处请填写代码实现递增序进行表插入排序,

返回值是关键字比较次数

Head是表头结点，不存放数据，info是待插入数据

要求每趟排序后 调用printLink函数，输出关键字的排列情况

\*/

void listSort(linkObject \*plist )

{

/\*----begin------\*/

linkObject \* p = plist->next;

linkObject \* q;

linkObject \* tmp;

while (NULL != p->next) {

// 腾出一个位置插入

for (q = plist; q->next != p->next && q->next->info <= p->next->info; q = q->next); // q->next恰好大于p->next, 将p->next插在q和q->next之间

if(q->next == p->next) {

p = p->next;

} else {

tmp = p->next;

p->next = p->next->next;

tmp->next = q->next;

q->next = tmp;

}

printLink (plist);

}

/\*-----end------\*/

}

/\* 第四关

此处请填写代码实现递增序进行shell排序,

要求每趟排序后 调用print函数，输出关键字的排列情况

\*/

void shellSort( SortObject \*Rec,int d )

{

int i, j, inc;

DataType tmp;

DataType \* data = Rec->data;

for(inc = d; inc > 0; inc /= 2) {

for (i = inc; i < Rec->NUM; ++i) {

tmp = data[i];

for (j = i - inc; j >= 0 && tmp < data[j]; j -= inc)

data[j + inc] = data[j];

if (j != i - inc)

data[j + inc] = tmp;

}

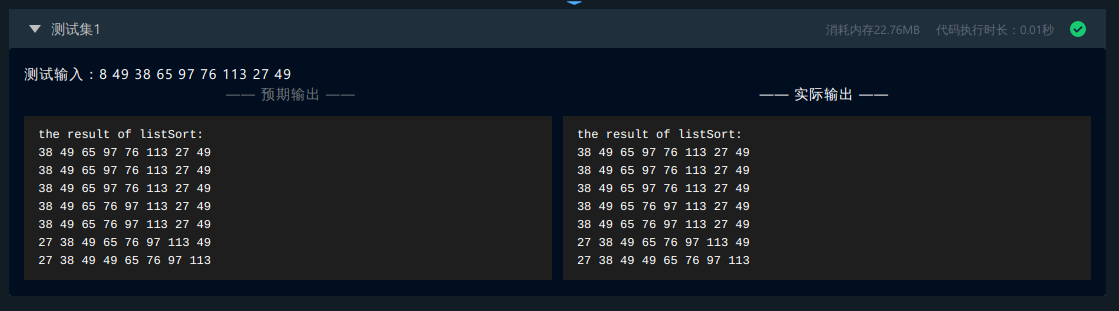
print (Rec);

}

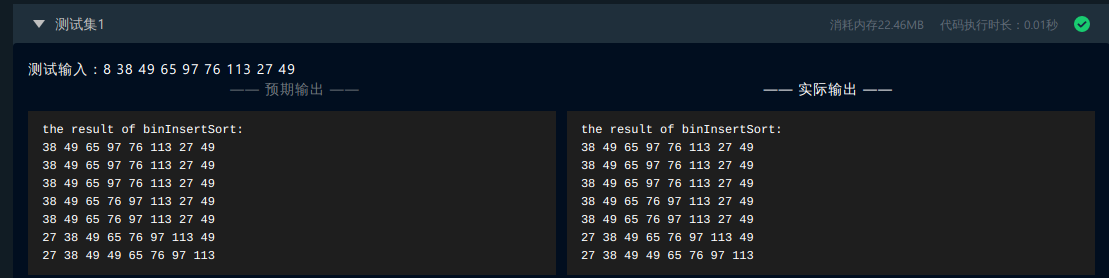
}

## 系统测试与结果

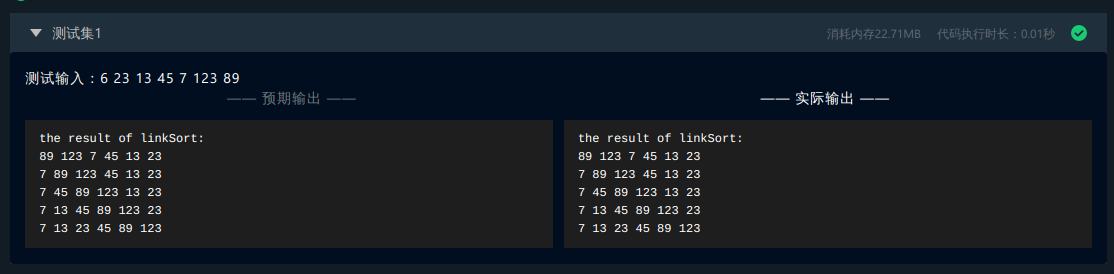
直接插入排序



二分插入排序



表插入排序



上述测试及结果证明了系统测试通过,各项操作函数能正常执行,并达到了实验要求的目标,实现了各种排序.

## 实验小结

通过本次实验, 我学会了各种排序.