

华中科技大学

课程实验报告

课程名称： 数据结构实验

专业班级 CS2210

学 号 U202115473

姓 名 刘欣逸

指导教师 郑渤龙

报告日期 2023 年 5 月 29 日

计算机科学与技术学院

目 录

1	基于顺序存储结构的线性表实现.....	1
1.1	问题描述.....	1
1.2	系统设计.....	2
1.3	系统实现.....	4
1.4	系统测试.....	4
1.5	实验小结.....	4
2	基于二叉链表的二叉树实现	5
2.1	问题描述.....	5
2.2	系统设计.....	5
2.3	系统实现.....	5
2.4	系统测试.....	5
2.5	实验小结.....	5
3	课程的收获和建议	6
3.1	基于顺序存储结构的线性表实现	6
3.2	基于链式存储结构的线性表实现	6
3.3	基于二叉链表的二叉树实现.....	6
3.4	基于邻接表的图实现.....	6
4	附录 A 基于顺序存储结构线性表实现的源程序	6
5	附录 B 基于链式存储结构线性表实现的源程序.....	6
6	附录 C 基于二叉链表二叉树实现的源程序	6
7	附录 D 基于邻接表图实现的源程序.....	6

1 基于顺序存储结构的线性表实现

1.1 问题描述

要求构造一个具有菜单的功能演示系统。该演示系统实现多个线性表管理。其中，在主函数中准备函数调用所需实参值、显示函数执行结果，并给出适当的操作提示。

实现依据最小完备性和常用性相结合的原则确定的初始化表、销毁表、清空表、判定空表、求表长和获得元素等 12 种基本运算，具体运算功能定义如下。

1. 初始化表：函数名称是InitaList(L)；初始条件是线性表L不存在已存在；操作结果是构造一个空的线性表。
2. 销毁表：函数名称是DestroyList(L)；初始条件是线性表L已存在；操作结果是销毁线性表L。
3. 清空表：函数名称是ClearList(L)；初始条件是线性表L已存在；操作结果是将L重置为空表。
4. 判定空表：函数名称是ListEmpty(L)；初始条件是线性表L已存在；操作结果是若L为空表则返回TRUE, 否则返回FALSE。
5. 求表长：函数名称是ListLength(L)；初始条件是线性表L已存在；操作结果是返回L中数据元素的个数。
6. 获得元素：函数名称是GetElem(L,i,e)；初始条件是线性表L已存在， $1 \leq i \leq \text{ListLength}(L)$ ；操作结果是用e返回L中第i个数据元素的值。
7. 查找元素：函数名称是LocateElem(L,e,compare())；初始条件是线性表L已存在；操作结果是返回L中第一个与e满足关系compare()关系的数据元素的位序，若这样的数据元素不存在，则返回值为0。
8. 获得前驱：函数名称是PriorElem(L,cur_e,pre_e)；初始条件是线性表L已存在；操作结果是若cur_e是L的数据元素，且不是第一个，则用pre_e返回它的前驱，否则操作失败，pre_e无定义。
9. 获得后继：函数名称是NextElem(L,cur_e,next_e)；初始条件是线性表L已存在；操作结果是若cur_e是L的数据元素，且不是最后一个，则用next_e返回它的后继，否则操作失败，next_e无定义。
10. 插入元素：函数名称是ListInsert(L,i,e)；初始条件是线性表L已存在，

$1 \leq i \leq \text{ListLength}(L)+1$ ；操作结果是在L的第i个位置之前插入新的数据元素e。

11. 删除元素：函数名称是`ListDelete(L,i,e)`；初始条件是线性表L已存在且非空， $1 \leq i \leq \text{ListLength}(L)$ ；操作结果：删除L的第i个数据元素，用e返回其值。
12. 遍历表：函数名称是`ListTraverse(L,visit())`，初始条件是线性表L已存在；操作结果是依次对L的每个数据元素调用函数`visit()`。

此外，还在基础功能的基础上实现了附加功能：

1. 最大连续子数组和：函数名称是`MaxSubArray(L)`；初始条件是线性表L已存在且非空，请找出一个具有最大和的连续子数组（子数组最少包含一个元素），操作结果是其最大和；
2. 和为K的子数组：函数名称是`SubArrayNum(L,k)`；初始条件是线性表L已存在且非空，操作结果是该数组中和为k的连续子数组的个数；
3. 顺序表排序：函数名称是`sortList(L)`；初始条件是线性表L已存在；操作结果是将L由小到大排序；
4. 实现线性表的文件形式保存：其中，
 - 需要设计文件数据记录格式，以高效保存线性表数据逻辑结构(D,{R})的完整信息；
 - 需要设计线性表文件保存和加载操作合理模式。
5. 实现多个线性表管理：设计相应的数据结构管理多个线性表的查找、添加、移除等功能。

实验目的：

1. 加深对线性表的概念、基本运算的理解；
2. 熟练掌握线性表的逻辑结构与物理结构的关系；
3. 物理结构采用顺序表，熟练掌握线性表的基本运算的实现。

1.2 系统设计

本系统提供一个采用顺序存储方式的线性表及其操作实现。系统可供选择的操作有：

- 基本操作：初始化线性表、销毁表、清空表、判定空表，求表长、获得元素、查找元素、获得某元素的前驱、获得某元素的后继、插入元素、删除元素、遍历线性表。
- 附加功能：最大连续子数组和、和为K的子数组、顺序表排序、实现线性表的文件形式保存、实现多个线性表管理。

1.2.1 头文件、宏和类型定义

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <limits.h>
5
6  #define TRUE 1
7  #define FALSE 0
8  #define OK 1
9  #define ERROR 0
10 #define INFEASIBLE -1
11 #define OVERFLOW -2
12 #define ISEMPY -3
13 #define LIST_INIT_SIZE 1000
14 #define LISTINCREMENT 10
15 #define max(i,j) ((i)>(j)?(i):(j))
16
17 typedef int status;
18 typedef int ElemType; //数据元素类型定义
19 typedef struct{ //顺序表（顺序结构）的定义
20     ElemType * elem;
21     int length;
22     int listsize;
23 } SqList;
24
```

```
25 typedef struct THELISTS{ //顺序表的管理表定义
26     struct ALIST{
27         char name[30];
28         SqList L;
29     } elem[50];
30     int length = 0;
31     int listssize = 50;
32 } LISTS;
```

1.3 系统实现

1.4 系统测试

1.5 实验小结

2 基于二叉链表的二叉树实现

2.1 问题描述

2.2 系统设计

2.3 系统实现

2.4 系统测试

2.5 实验小结

3 课程的收获和建议

3.1 基于顺序存储结构的线性表实现

3.2 基于链式存储结构的线性表实现

3.3 基于二叉链表的二叉树实现

3.4 基于邻接表的图实现

4 附录 A 基于顺序存储结构线性表实现的源程序

5 附录 B 基于链式存储结构线性表实现的源程序

6 附录 C 基于二叉链表二叉树实现的源程序

7 附录 D 基于邻接表图实现的源程序