

前 言

数据结构是计算机程序设计的重要理论技术基础，它是计算机学科的核心课程。通过实验帮助学生消化理解课程内容，提高自身编程能力。实验目的和任务是使实验者较全面地掌握各种常用的数据结构，为学习后续软件课程提供必要的基础，提高运用数据结构解决实际问题的能力。通过上机实验加深对课程内容的理解，增加感性认识，提高软件设计、编写及程序调试的能力。我们要求所编制的程序能正确运行，并提交实验报告。

数据结构的内容具体包括：

- 1、线性结构；
- 2、树状结构；
- 3、图状结构；
- 4、查找；
- 5、排序。

本实验讲义是针对数据结构中的操作编写，提出了上机实验的要求，并且具体安排了 8 个方面实验（其中设计性题目 5 个，验证性实验 1 个，综合设计性题目 2 个），每个实验 3 学时，共 24 学时。由于篇幅和课时的限制，教材和课堂讲授中只能介绍一些典型的例题。建议同学们尽可能多做例题和习题，以掌握数据结构特点及编程思路，另外需要说明的是本讲义对学生上机仅提供一个参考，学生可不局限于讲义讲的那些操作，同学可以多做或是做自己感兴趣的操作。

本实验所依据的教材是：《数据结构》，严蔚敏等编著，清华大学出版社。上机环境是 C 语言上机环境。

[实验题目一] 顺序表的基本操作

一、实验目的

- 1、掌握顺序表的结构特点。
- 2、主要掌握顺序表的插入和删除算法。
- 3、学会分析算法的时间复杂度。

二、预习内容

- 1、C 的上机环境；
- 2、教材 2.2 节线性表的顺序表示和实现。

三、实验内容

顺序表的基本操作有以下 2 个实验内容：

- 1、已知顺序表 L，在第 i 个元素前插入元素 e。

A、首先要定义顺序表的数据结构，可参考教材 SqList 类型定义；

B、定义一个函数能够初始化一个顺序表，可参考教材算法 2.3；

C、编写一个函数, 例如：

```
Status ListInsert_Sq(SqList *L, int i, int e) {  
    //已知顺序表 L，在第 i 个元素前插入元素 e。  
    .....  
}
```

上边算法可参考教材算法 2.4

D、读懂函数 (PrintList)，它能够输出一个顺序表中的所有元素的值。

E、编写 ListInsert_Sq 函数，并在 main 中进行测试

F、程序编制上的部分提示，仅供参考：

```
无标题 - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

#include <stdio.h>
#include <stdlib.h>
#define OK 1
#define ERROR 0
#define OVERFLOW -2
#define LIST_INIT_SIZE 100
#define LISTINCREMENT 10
typedef int Status;
typedef struct {
    int *elem;
    int length; //当前长度
    int listsize; //当前分配的最大的容量
} SqList;
Status InitList_Sq(SqList *L) {
    //功能注释, 构造一个空的线性表L
    L->elem=(int *)malloc(LIST_INIT_SIZE *sizeof(int));
    if(!L->elem) exit(OVERFLOW);
    L->length=0;
    L->listsize=LIST_INIT_SIZE;
    return OK;
} //InitList_Sq 时间复杂度O(1)
void CreateList(SqList *L, int n) {
    //创建顺序表, 含有n个数据元素
    int i;
    if(n<0) return;
    printf("请为顺序表输入%d个整数:\n", n);
    for(i=0; i<n; i++)
        scanf("%d", &L->elem[i]);
    L->length=n;
}
```

```
无标题 - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

void PrintList(SqList L) {
    //输出顺序表中的数据
    int i;
    printf("顺序表中的数据依次为:\n");
    for(i=0; i<L.length; i++)
        printf("%d ", L.elem[i]);
    printf("\n");
}
void main() {
    SqList L;
    InitList_Sq(&L); //初始化顺序表L
    CreateList(&L, 5); //向顺序表中输入5个数据
    PrintList(L); //将顺序表中的元素依次输出
    free(L.elem); //释放所要的空间
}
```

注：如果做过了内容 1，那么可以做下面这个题目

已知非递减顺序表 L（录入数据时可以手动保证 L 是非递减的），
插入一个元素 e 后，L 仍然是非递减的。

2、已知顺序表 L，从第 i 个元素起（包括第 i 个）连续删除 k 个元素。

A、在上边的程序(main 函数的上边)，增加编写一个函数, 例如：

```
int ListDel_Sq(SqList *L, int i, int k) {  
    /*顺序表 L，从第 i 个元素起连续删除 k 个元素。  
    删除成功返回 1，否则返回 0，如果 i 或 k 不合理，那么一个元素也不删除。*/  
    .....  
} //本算法可参考教材算法 2.5
```

B、扩充上边主函数(main) 对初始的顺序表 L（含 n 个数）调用

ListDel_Sq (&L, 2, 3) 函数，表示从 L 中第 2 个元素起连续删除 3 个数，并输出调用 ListDel_Sq 后的 L 中的各个元素的值。

提示：ListDel_Sq 这个函数如果是高效的，那么它的时间复杂度应为

$O(n)$, 如果时间复杂度为 $O(n^2)$ ，那么需要对算法进行改进。

3、在顺序表 L 中查找一个元素（选做） 本算法可参考教材算法 2.6

4、两个非递减的顺序表合并问题（选做） 本算法可参考教材算法 2.7

注：选做的题目无需写到实验报告上。

四、实验报告要求

- 1、写好班级、姓名、学号和实验题目；
- 2、写上实验的内容及程序源代码；
- 3、写上程序测试的结果；
- 4、写个实验小结。