



高等学校计算机类“十三五”规划教材
COMPUTER

省级精品课程配套教材

数据结构与算法设计 实践与学习指导

主 编 齐爱玲 张小艳
副主编 李占利 王伯槐



西安电子科技大学出版社
<http://www.xduph.com>

高等学校计算机类“十三五”规划教材
省级精品课程配套教材

数据结构与算法设计 实践与学习指导

主 编 齐爱玲 张小艳
副主编 李占利 王伯槐

西安电子科技大学出版社

内 容 简 介

本书是作者多年讲授“数据结构”课程及指导学生实验的教学经验的集成,与西安电子科技大学出版社出版的《数据结构与算法设计》一书相配套。全书分为两部分:第一部分是实验指导,其中,第一章给出了实验安排和实验步骤,第二至六章内容均由实验目的、实验指导和实验题组成,精选了涵盖各种数据结构的典型实验,每个实验给出了在C语言环境下调试运行的结果;第二部分是学习指导,各章内容均由基本知识点、习题解析和自测题及参考答案组成,每组习题均与教材中的内容相对应。书末给出了两套考试试题及参考答案。

本书可以配合《数据结构与算法设计》一书使用,起到衔接课堂教学和指导实验教学的作用;也可作为高等院校学生学习“数据结构”课程的辅助教材及计算机学科研究生入学考试的辅导教材;对于从事计算机软件开发和应用的工程技术人员,本书也具有一定的参考价值。

图书在版编目(CIP)数据

数据结构与算法设计实践与学习指导/齐爱玲,张小艳主编. —西安:西安电子科技大学出版社,2016.2
高等学校计算机类“十三五”规划教材

ISBN 978-7-5606-4014-3

I. ① 数… II. ① 齐… ② 张… III. ① 数据结构—高等学校—教材
② 电子计算机—算法设计—高等学校—教材 IV. ① TP311.12 ② TP301.6

中国版本图书馆 CIP 数据核字(2016)第 021596 号

策 划 李惠萍

责任编辑 李惠萍

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2016年2月第1版 2016年2月第1次印刷

开 本 787毫米×1092毫米 1/16 印 张 13.5

字 数 319千字

印 数 1~3000册

定 价 25.00元

ISBN 978 - 7 - 5606 - 4014 - 3/TP

XDUP 4306001-1

如有印装问题可调换

前 言

数据结构是计算机程序设计重要的理论技术基础，它不仅是计算机学科的核心课程，而且已经成为计算机相关专业必要的选修课。

数据结构是学生从传统的形象思维转向科学的抽象思维的第一门课程，需要学生在思维认识上有一个转变，这使得数据结构成为一门公认的比较难学的课程。大多数学生在学习数据结构过程中普遍反映上课听得明白，遇到实际问题却无从下手。对于习题，即便能够解出，答案中也往往有错误；有时即使答案正确，也不“合格”，原因是算法太低效，思路不清，可读性较差等。究其原因，主要是对实验重视不够，导致动手能力太差。因此，在教学过程中，采用以实验加课堂演练为重点的计算思维教学模式，让学生在实验和适当的课堂演练中学习知识、消化知识，强化计算思维，进而培养学生的科学与工程计算能力，不失为一种有效的教学方法。

为此，我们以计算思维为导向，以科学与工程计算能力培养为目标，编写了这本《数据结构与算法设计实践与学习指导》教学用书。

本书是和《数据结构与算法设计》一书相配套的实践与学习指导和习题解析教材。全书分为两大部分，第一部分是实验指导，第二部分是学习指导。实验指导部分设计了具有趣味性的实践课题，并完整地介绍了数据抽象、表示及算法设计过程，而且均给出了 C 语言描述，并对所有算法进行了详尽的注释，有利于学生理解算法的基本思想。这样学生只要读懂算法，便能实现程序，激发了学生的学习兴趣，进一步提高了学生对数据的抽象能力和程序设计能力，使得计算思维能力的培养切实可行。学习指导部分详细给出了书中习题的解答思路和参考答案。

编写第一部分的出发点是通过一些典型的实例练习，使学生掌握如何利用数据结构知识去分析和解决实际问题。本书安排了五个实践单元，各单元的训练重点在于基本数据结构和经典算法的训练。较大的实验题比平时的实验题要复杂得多，也更接近实际。各实践单元与教科书的各章具有基本的对应关系，每一章实验题均包含实验目的、实验指导和实验题。其中，实验指导中包含的验证性实验和综合实验都是精选的实验题目，每一道实验题都给出了重点语句的注释，并给出了调试结果，以帮助读者更好地理解程序。每章的选做实验中安排有难度不等的若干个实验题，并给出了部分

实验的简要分析,以帮助读者解题。在编写过程中,我们将计算思维的能力培养融入其中。计算思维的本质是抽象和自动化。通过数据抽象进行数据的选取和组织,应用约简、嵌入、转化、仿真等方法,实现问题的求解。在实验教学环节采用计算思维教学模式,让学生在实验中学习知识、消化知识,以战代练,强化计算思维。最终,使学生将计算思维转化成为自己的认识论和方法论,并逐渐形成科学的思维方法和严谨的科学态度。

编写第二部分的目的是通过对习题的分析与解答,使学生充分掌握数据结构的原理以及求解数据结构问题的思路和方法,从而编写出符合数据结构规范的算法,进一步加深对基本概念的理解,提高学生分析问题和解决问题的能力。第二部分共分为九章,每一章都包含基本知识点、习题解析、自测题及参考答案。基本知识点帮助读者回顾本章的基本知识和重要知识点。每一章习题解析都与《数据结构与算法设计》一书中的内容相对应,对教科书中的每道习题给出了比较详尽的解答,希望帮助学生解决一些学习上的疑难之处,给他们一点启发。每组习题之后附加了自测题,建议大家先自己做,然后再看答案,这样有利于学习、思考、掌握,否则,既容易妨碍自己的思路,又可能造成依赖性。另外,本书的解答只作为参考,希望大家能有更好的解答。

本书实验指导部分的第四、五、六章,学习指导部分的第五、六、七、八章及附录内容由齐爱玲编写;实验指导部分的第一、二章,学习指导部分的第二章内容由张小艳编写;实验指导部分的第三章以及学习指导部分的第三、四章内容由李占利编写;学习指导部分的第一、九章内容由王伯槐编写。全书由齐爱玲统稿、修改、定稿。张小艳、李占利审稿。

限于编者水平,书稿虽几经修改仍难免存在不足之处,敬请广大读者批评指正。

编 者
2016年1月



目 录

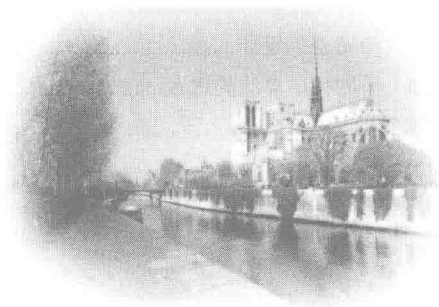
第一部分 实验指导

第一章 实验规范指导	1	4.1 实验目的	42
1.1 基于计算思维的数据结构实验教学	1	4.2 实验指导	42
1.2 本书实验安排	2	4.2.1 串基本操作的实现	42
1.3 实验步骤	3	4.2.2 用三元组表实现稀疏矩阵的基本 操作	46
第二章 线性表及其应用	6	4.2.3 KMP 算法的实现	49
2.1 实验目的	6	4.2.4 输出魔方阵	51
2.2 实验指导	6	4.3 实验题	55
2.2.1 顺序表的应用	6		
2.2.2 单链表的应用	9	第五章 树、图及其应用	56
2.2.3 约瑟夫环问题	14	5.1 实验目的	56
2.2.4 狐狸逮兔子实验	17	5.2 实验指导	56
2.3 实验题	19	5.2.1 二叉树的基本运算实现	56
第三章 栈与队列及其应用	21	5.2.2 图遍历的演示	60
3.1 实验目的	21	5.2.3 电文的编码和译码	63
3.2 实验指导	21	5.2.4 拓扑排序实验	68
3.2.1 顺序栈的基本操作实现	21	5.3 实验题	75
3.2.2 链栈的基本操作实现	25		
3.2.3 循环队列的基本操作实现	28	第六章 查找、排序及其应用	77
3.2.4 后缀表达式求值	30	6.1 实验目的	77
3.2.5 八皇后问题	34	6.2 实验指导	77
3.2.6 模拟服务台前的排队问题	37	6.2.1 静态查找表	77
3.3 实验题	40	6.2.2 动态查找表	80
第四章 串、数组及其应用	42	6.2.3 哈希表的设计	82
		6.2.4 不同排序算法的比较	86
		6.3 实验题	92

第二部分 学习指导

第一章 绪论	93	1.1 基本知识点	93
--------------	----	-----------------	----

1.2 习题解析.....	93		
1.3 自测题及参考答案.....	98		
第二章 线性表	100	第六章 二叉树与树	149
2.1 基本知识点.....	100	6.1 基本知识点.....	149
2.2 习题解析.....	101	6.2 习题解析.....	150
2.3 自测题及参考答案.....	123	6.3 自测题及参考答案.....	160
第三章 栈与队列	127	第七章 图	162
3.1 基本知识点.....	127	7.1 基本知识点.....	162
3.2 习题解析.....	127	7.2 习题解析.....	163
3.3 自测题及参考答案.....	133	7.3 自测题及参考答案.....	174
第四章 串	135	第八章 查找表	176
4.1 基本知识点.....	135	8.1 基本知识点.....	176
4.2 习题解析.....	135	8.2 习题解析.....	176
4.3 自测题及参考答案.....	139	8.3 自测题及参考答案.....	183
第五章 数组和广义表	141	第九章 排序	186
5.1 基本知识点.....	141	9.1 基本知识点.....	186
5.2 习题解析.....	141	9.2 习题解析.....	186
5.3 自测题及参考答案.....	147	9.3 自测题及参考答案.....	193
附录一 硕士研究生入学考试试题及答案(一)	196		
附录二 硕士研究生入学考试试题及答案(二)	203		
参考文献	210		





第一部分 实验指导

第一章 实验规范指导

1.1 基于计算思维的数据结构实验教学

“数据结构与算法”是一门研究非数值计算的程序设计问题中有关计算机的操作对象以及它们之间的关系和操作等的学科。在学习数据结构与算法课程的基础上,通过对不同数据结构及其算法的上机实践,以加强对相关理论知识的理解与掌握,同时使学生具备针对数学和工程问题按照计算机求解问题的基本方式解决问题的能力,进而能够构建出相应的算法和基本程序。因此,在数据结构课程教学过程中实验环节起着至关重要的作用。本课程重在培养学生的数据抽象能力和复杂程序设计能力,进而培养学生的科学与工程计算能力。

为了促进学习者计算思维能力的养成,在实验教学中,需要为每个知识点提供不同的实验题目,以便不同专业方向的学习者可以自主选择,通过学习者的自主实验培养其计算思维能力和对数据结构知识点的掌握。因此本书的实验材料具有多样化特点。编者精心设计了多种由浅入深的实验题目,这些实验包括以下三个方面:

(1) 单个知识点实验。设计该知识点的典型问题,让学习者模仿该类问题的求解方法,初步掌握计算思维方法。

(2) 知识点综合实验和课程综合实验。这部分实验引导学生将不同的知识点和方法综合应用到该实验问题的解决中,提高学生综合应用所学知识的能力,使学生可以对问题进行分解,提出该问题的解决方案;训练学生的计算思维,提高综合运用计算思维方法的能力。

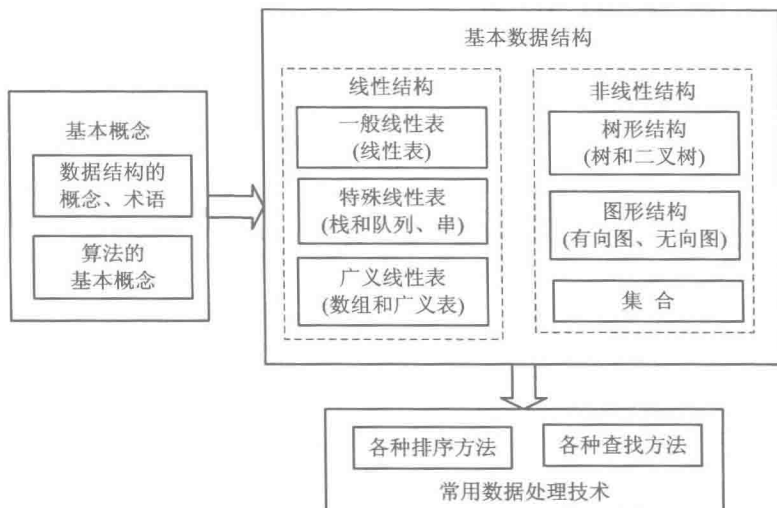
(3) 自选实验。通过课堂理论教学以及前两方面由浅入深的实践教学,学生具备了一定的解决数据结构问题的能力,设计自选实验的目的是使学生能够自己综合运用数据结构所学知识和前期积累的实践经验,达到自己动手解决问题的能力,是对学生计算思维能力培养结果的一个检验。

通过不同层次的实验,学生不仅可以验证一些理论知识的正确性,同时还可以通过灵



活多样性的实验题目提高上机编程能力,进一步培养学生按照计算思维方法对所研究的问题进行抽象、归纳、建立数学模型,或确定问题求解模型、步骤,或确定算法,达到培养学生的计算思维能力,为学生将来应用科学与工程计算从事科学研究、解决工程实践问题奠定坚实的理论和方法论基础的目的。

数据结构课程的内容大致可分为基本概念、基本数据结构、常用的数据处理技术三大部分,其多维知识架构如实验图 1.1 所示。通过该知识点架构图我们可以清楚地看出数据结构课程的脉络。数据结构实验正是围绕数据结构的知识点展开,是对理论知识点的深化和实践。通过单个知识点的实验加深读者对每种数据结构的理解,通过综合应用理解每种数据结构在实际中的应用。实验中对常用的数据处理技术进行了强化训练。



实验图 1.1 数据结构知识点架构图

1.2 本书实验安排

“数据结构与算法”是一门研究非数值计算的程序设计问题中有关计算机的操作对象以及它们之间的关系和操作等的学科。本课程重在培养学生逻辑思维能力,培养学生的数据抽象能力和复杂程序的设计能力。因此,数据结构课程教学过程中实验环节起着至关重要的作用。在学习数据结构课程的基础上,通过对不同数据结构及其算法的上机实践,可以加强对相关理论知识的掌握。通过数据结构实践教学,不仅能使学生系统掌握数据结构这门课的主要内容,而且培养学生分析问题和解决问题以及编程等实际动手能力。

本课程教学大纲要求总学时为 64~80,理论教学为 48~64 学时。根据教学内容和教学目标,建议实验课共开设 16 个学时,其中验证性实验 4 学时,综合实验 12 学时。要求在教学相关章节同时进行专业实验并写出实验报告。实验课分班进行,每个实验班 35 人左右,配备一名实验指导教师。

本书共给出了 22 个实践设计实例,其中包括验证性实验和综合实验,见实验表 1.1 所



示。并在每部分后给出了选作实验，供学生选择练习。

实验表 1.1 实验项目与内容提要

序号	实验项目名称	实验学时	实验类型	内 容 提 要
1	线性表及其应用	4	验证	1. 顺序表的应用 2. 链表的应用
			综合	1. 约瑟夫环问题 2. 狐狸逮兔子实验
2	栈与队列及其应用	2	验证	1. 顺序栈的基本操作实现 2. 链栈的基本操作实现 3. 循环队列的基本操作实现
			综合	1. 后缀表达式求值 2. 八皇后问题 3. 模拟服务台前的排队问题
3	串、数组及其应用	2	验证	1. 串基本操作的实现 2. 用三元组表实现稀疏矩阵的基本操作
			综合	1. KMP 算法的实现 2. 输出魔方阵
4	树、图及其应用	4	验证	1. 二叉树的基本运算实现 2. 图遍历的演示
			综合	1. 电文的编码和译码 2. 拓扑排序实验
5	查找、排序及其应用	4	验证	1. 静态查找表 2. 动态查找表
			综合	1. 哈希表的设计 2. 不同排序算法的比较

1.3 实验步骤

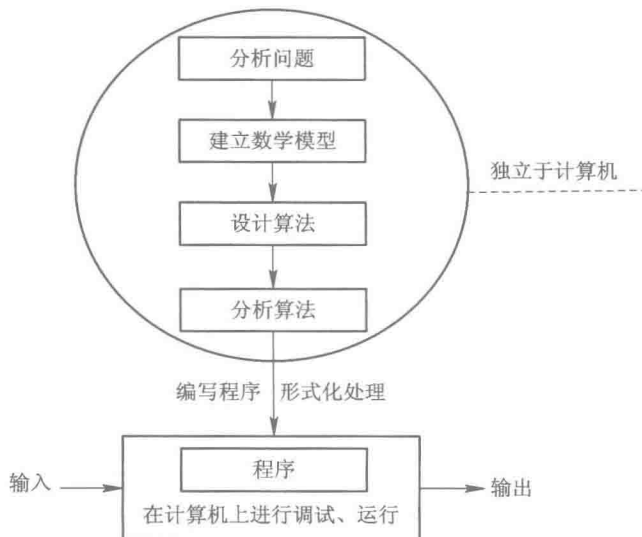
人们解决问题的过程是：观察问题→分析问题→脑中收集信息→根据已有的知识、经验进行判断、推理→采用某种方法和步骤解决。人们用计算机解决问题时是将采用的方法和步骤利用计算机能够识别的计算机语言编制代码，“告诉”计算机进行处理。其过程如实验图 1.2 所示。

1. 分析问题

在进行设计之前，首先应该充分地分析和理解问题，明确问题要求做什么，限制条件是什么，也就是对所需完成的任务作出明确的描述。例如：输入数据的类型、值的范围及输入的形式；输出数据的类型、值的范围及输出的形式；若是会话式的输入，则结束标志



是什么, 是否接受非法的输入, 对非法输入的回答方式是什么; 等等。用科学规范的语言对所求解的问题做准确的描述。



实验图 1.2 利用计算机解决问题的过程

2. 数学建模(数据及其之间关系的抽象)及操作集合的定义

在分析问题的基础上, 抽象出问题相关的数据及其之间的关系, 也就是建立相应的数学模型。这里的数学模型可以是线性表、树、图。这些模型可以用 C 语言中的数据类型描述。之后, 根据具体问题所需的处理逻辑抽象出操作集合。

3. 算法设计

依据第二步得到的数学模型及操作集合进行算法设计。针对问题列出指令序列, 这里可以用伪码语言表示。

4. 算法分析

算法分析一般包括正确性分析、时间及空间效率分析。

5. 编码实现

运用熟悉的程序设计语言进行编码, 在编码的过程中需注意以下几点:

- (1) 对函数功能和重要变量进行注释。
- (2) 一定要按格式书写程序, 分清每条语句的层次, 对齐括号, 这样便于发现语法错误。
- (3) 控制 if 语句连续嵌套的深度, 分支过多时应考虑使用 Switch 语句。

6. 上机准备和上机调试

上机准备包括以下几个方面:

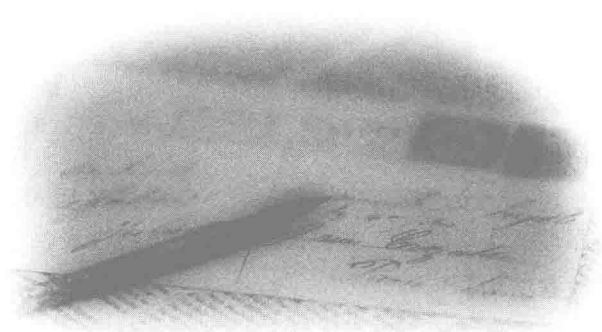
- (1) 熟悉 C 语言用户手册或程序设计指导书。
- (2) 注意 Turbo C、VC 与标准 C 语言之间的细微差别。
- (3) 熟悉机器的操作系统和语言集成环境的用户手册, 尤其是最常用的命令操作, 以便顺利进行上机操作的基本活动。



(4) 掌握调试工具，考虑调试方案，设计测试数据并手工得出正确结果。“磨刀不误砍柴工”。学生应该熟练运用高级语言的程序调试器 **DEBUG** 调试程序。

上机调试程序时要带一本高级语言教材或手册。调试最好分模块进行，自底向上，即先调试低层过程或函数，必要时可以另写一个调用驱动程序。这种表面上看来麻烦的工作实际上可以大大降低调试所面临的复杂性，提高调试工作效率。

在调试过程中可以不断借助 **DEBUG** 的各种功能提高调试效率。调试中遇到的各种异常现象往往是预料不到的，此时不应“苦思冥想”，而应借助系统提供的调试工具确定错误。调试正确后，认真整理源程序及其注释，打印出带有完整注释且格式良好的源程序清单和结果。





第二章 线性表及其应用

2.1 实验目的

线性表是最简单的基本数据结构，在解决实际问题中有着广泛的应用。本次实验的主要目的在于熟悉线性表的基本运算在两种存储结构上的实现，巩固对线性表逻辑结构的理解，掌握线性表的顺序存储结构和链式存储结构及基本操作的实现，为应用线性表解决实际问题奠定良好的基础。通过本章的实验，学会分析问题、解决问题，可从具体问题中抽象出解决问题的方法，选择合理的存储结构，设计合理的解决方法。通过本次实验还可以帮助学生复习 C 语言。

2.2 实验指导

线性表的存储结构有两种：顺序存储结构和链式存储结构。这两种存储结构各有其特点。

顺序存储结构具有按元素序号随机访问的特点，而且方法简单，不用为表示结点间的逻辑关系而增加额外的存储开销。但它也有缺点，即在做插入、删除操作时，平均需要移动大约表中一半的元素，因此对 n 较大的顺序表的操作效率低下。

链式存储结构不需要用地址连续的存储单元来实现，它是通过“链”建立起数据元素之间的逻辑关系，因此对线性表的插入、删除运算不需要移动数据元素。

实验内容围绕着线性表的两种存储结构展开，使学生能够充分理解线性表的特点及其两种存储结构的特点。

2.2.1 顺序表的应用

【问题描述】

建立一个递增有序的顺序表，实现将 X 插入到线性表的适当位置上，以保持线性表的有序性；实现插入、求表长以及输出该线性表。

【数据结构】

本设计使用顺序表实现。

【算法设计】

顺序表结构定义包含两部分：一部分是用于存储数据的一维数组 `list`，一部分是记录线



性表中最后一个元素在数组中位置的变量 last。为简单起见,线性表的数据元素为 int。程序中设计了六个函数:

➤ 函数 InitList()用来初始化一个空的线性表;

➤ 函数 PutseqList()用来输入一个线性表;

注意:线性表的第一个元素存储在数组 list 的 0 号单元, last 存储数组最后一个元素的存储位置为 n-1。

➤ 函数 LengthList()用来求线性表的长度;

➤ 函数 PositionList()用来判断 X 的插入位置;

注意:这里返回的位置值为找到的数组下标 + 1, 因为下标 j 表示第 j+1 个元素, 所以返回 j+1。

➤ 函数 InsertList()用来插入数据 X;

➤ 函数 OutputSeqList()用来输出线性表。

【程序实现】

```
#include<stdio.h>
#define MAXSIZE 100
typedef int ElemType;
/*定义线性表*/
typedef struct
{
    ElemType list[MAXSIZE];
    int last;
}SeqList;
/*创建空表*/
void InitList(SeqList *L)
{
    L->last = -1;
}
/*输入递增有序顺序表*/
void PutseqList(SeqList *L, int n)
{
    int i;
    for(i = 0; i<n; i++)
        scanf("%d", &(L->list[i]));
    L->last = L->last+n;
}
/*求表长*/
int LengthList(SeqList *L)
{
    int Len;
```



```
        Len=L->last+1;
        return Len;
    }
    /*判断插入位置*/
    int PositionList(SeqList *L, int X)
    {
        int j;
        for(j = 0; j<=L->last; j++)
            if(X<L->list[j]) /*如果找到位置就返回地址，否则直到循环结束再返回最后一个地址*/
                return j+1; /*注意这里要加 1，下标 j 表示第 j+1 个元素，所以返回 j+1*/
        return (L->last+2); /*这里不能用 else，否则就属于 for 循环里面的了*/
    }
    /*插入元素*/
    int InsertList(SeqList *L, int i, int e)
    {
        int k;
        if((i<1)||i>L->last+2))
        {
            printf("插入位置不合理");
            return(0);
        }
        if(L->last>=MAXSIZE-1)
        {
            printf("表已满无法插入");
            return(0);
        }
        for(k=L->last; k>=i-1; k--)
            L->list[k+1] = L->list[k];
        L->list[i-1] = e;
        L->last++;
        return(1);
    }
    /*输出元素*/
    int OutputSeqList(SeqList *L)
    {
        int i;
        printf("输出结果为: ");
        for(i=0; i<=L->last; i++)
            printf("%d", L->list[i]);
```



```
        return(L->list[i]);
    }
void main()
{
    int s, c;
    SeqList L;
    InitList(&L);
    printf("请输入顺序表长度: ");
    scanf("%d", &s);
    printf("请输入递增顺序表: ");
    PutseqList(&L, s);
    LengthList(&L);
    printf("表长为%d\n", LengthList(&L));
    printf("请输入要插入的元素: ");
    scanf("%d", &c);
    InsertList(&L, PositionList(&L, c), c);
    OutputSeqList(&L);
    printf("\n");
}
```

【运行与测试】

运行如下:

```
请输入顺序表长度: 10
请输入递增顺序表: 2 6 10 15 20 25 30 39 50 68
表长为10
请输入要插入的元素: 25
输出结果为: 2 6 10 15 20 25 25 30 39 50 68
```

2.2.2 单链表的应用

【问题描述】

完成两个多项式的相加操作: 已知有两个多项式 $P_m(x)$ 、 $Q_m(x)$, 设计算法实现 $P_m(x) + Q_m(x)$ 运算, 而且对加法运算不重新开辟存储空间。要求用链式存储结构实现。例如: $P_m(x) = 5x^3 + 2x + 1$, $Q_m(x) = 3x^3 + x^2 - 2x - 3$, 其计算输出结果为: $8x^3 + 1x^2 - 2$ 。

【数据结构】

本设计使用单链表实现。

【算法设计】

程序中设计了四个函数:

- 函数 Init() 用来初始化一个空链表;
- 函数 CreateFromTail() 用来创建一个链表, 这里是用尾插法来创建链表;
- 函数 Polyadd() 用来实现两个多项式相加算法;
- 函数 Print() 用来输出多项式。



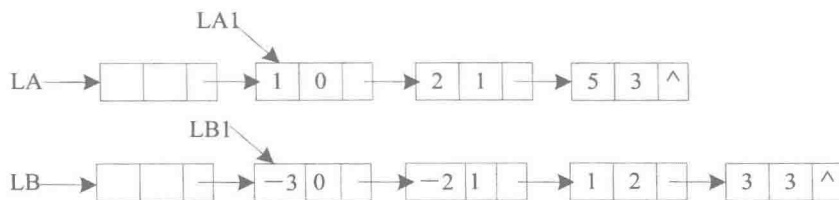
两个多项式相加算法的实现，首先是将两个多项式分别用链表进行存放。可以设置两个指针 LA1 和 LB1 分别从多项式 $P_m(x)$ 和 $Q_m(x)$ 的首结点移动，比较 LA1 和 LB1 所指结点的指数项，则可以分下面三种情况进行处理：

(1) 若 $LA1 \rightarrow \text{exp} < LB1 \rightarrow \text{exp}$ ，则 LA1 所指结点为多项式中的一项，LA1 指针在原来的基础上向后移动一个位置。

(2) 若 $LA1 \rightarrow \text{exp} = LB1 \rightarrow \text{exp}$ ，将对应项的系数相加，然后分两种情况处理：如果系数项的和为零，则释放 LA1 和 LB1 所指向的结点；如果系数项的和不为零，则修改 LA1 所指向结点的系数域，释放 LB1 结点。

(3) 若 $LA1 \rightarrow \text{exp} > LB1 \rightarrow \text{exp}$ ，则 LB1 所指结点为多项式中的一项，LB1 指针在原来的基础上向后移动一个位置。

实验图 2.1 为两个多项式链表的示意图。



实验图 2.1 由两个多项式建立的两个线性链表的示意图

【程序实现】

```

#include<stdio.h>
#include<malloc.h>
#include<stdlib.h>
typedef struct poly
{
    int exp;           /*指数幂*/
    int coef;          /*系数*/
    struct poly *next; /*指针域*/
}PNode, *PLinklist;
int Init(PLinklist *head) /*链表初始化*/
{
    *head=(PLinklist)malloc(sizeof(PNode));
    if(*head)
    {
        (*head)->next = NULL;
        return 1;
    }
    else
        return 0;
}
  
```