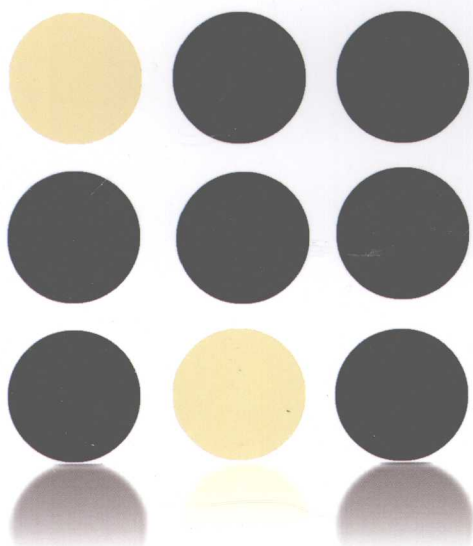


 高等院校计算机专业课程综合实验系列规划教材



丛书主编 何钦铭 陈根才

何钦铭 冯雁 陈越 著
廖明宏 主审

数据结构 课程设计



ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

高等学校计算机专业基础课系列教材系列规划教材

圖書在版編目(CIP)數據

廖明宏 主审

浙江大學出版社

内容提要

本书主要围绕数据结构的基本知识点,设计了8个大型综合性练习案例,通过相关背景知识的回顾、题目解析与实现要点的分析以及测试方法分析等,为学生完成综合性的数据结构实践提供参考。本书所附光盘内容为这8个课程设计案例的源程序及教材中描述的测试数据。本教材还提供了8个课程设计题目及其简要的提示。这些案例和题目大多取材于程序设计竞赛题,具有较好的趣味性和技巧性。

本教材可作为数据结构课程配套的实验教材,也适合于对C程序设计以及数据结构有初步基础的读者学习数据结构的设计方法和提高编程技巧。

图书在版编目(CIP)数据

数据结构课程设计 / 何钦铭,冯雁,陈越著. —杭州:
浙江大学出版社, 2007. 8
(高等院校计算机专业课程综合实验系列规划教材)
ISBN 978-7-308-05521-5

I. 数… II. ①何…②冯…③陈… III. 数据结构—课程
设计—高等学校—教材 IV. TP311.12

中国版本图书馆CIP数据核字(2007)第139221号

数据结构课程设计

何钦铭 冯 雁 陈 越 著
廖明宏 主审

丛书主编 何钦铭 陈根才
策 划 黄娟琴 希 言
责任编辑 黄娟琴 邹小宁
封面设计 氧化光阴
出版发行 浙江大学出版社
(杭州天目山路148号 邮政编码310028)
(E-mail: jsjsyb@zju.edu.cn)
(网址: <http://www.zjuxsj.com>)

排 版 浙江大学出版社电脑排版中心
印 刷 临安市曙光印务有限公司
开 本 787mm×1092mm 1/16
印 张 9.75
字 数 250千
版 次 2007年8月第1版 2007年8月第1次印刷
印 数 0001—3000
书 号 ISBN 978-7-308-05521-5
定 价 20.00元(附光盘)

版权所有 翻印必究 印装差错 负责调换

高等院校计算机专业课程

综合实验系列规划教材编委会

主 任

齐治昌 国防科技大学教授,教育部软件工程专业教学指导分委员会副主任

副主任

陈道蓄 南京大学计算机系教授,教育部计算机科学与技术专业教学指导分委员会副主任

蒋宗礼 北京工业大学计算机学院副院长,教授,教育部计算机科学与技术专业教学指导分委员会秘书长

委 员(按姓氏笔画排列)

王志英 国防科技大学计算机学院副院长,教授,教育部高等学校计算机科学与技术专业教学指导分委员会副主任

左保河 华南理工大学软件学院副教授,教育部高等学校软件工程专业教学指导分委员会委员

刘 强 清华大学副教授,教育部高等学校软件工程专业教学指导分委员会秘书长

孙吉贵 吉林大学计算机学院副院长,教授,教育部高等学校计算机科学与技术专业教学指导分委员会委员

庄越挺 浙江大学计算机学院副院长,教授,教育部计算机科学与技术专业教学指导分委员会委员

吴 跃 电子科技大学计算机学院教授,教育部计算机科学与技术专业教学指导分委员会委员

李 彤 云南大学软件学院副院长,教授,教育部计算机科学与技术专业教学指导分委员会委员

邹逢兴 国防科学技术大学教授,国家级教学名师

陈志刚 中南大学信息学院副院长,教授,教育部高等学校计算机科学与技术专业教学指导分委员会委员

岳丽华 中国科学技术大学教授,教育部高等学校计算机科学与技术专业教学指导分委员会委员

徐宝文 东南大学教授,教育部高等学校软件工程专业教学指导分委员会委员

廖明宏 哈尔滨工业大学计算机学院副院长,教授,教育部高等学校计算机科学与技术专业教学指导分委员会委员

管会生 兰州大学信息科学与工程学院副院长,教授,教育部高等学校理工类计算机基础课程教学指导分委员会秘书长

序 言

近 10 多年来,以计算机和通信技术为代表的信息技术迅猛发展,并已深入渗透到国民经济与社会发展的各个领域。信息技术成为国家产业结构调整 and 推动国民经济与社会快速发展的最重要的支撑技术。与此同时,深入掌握计算机专业知识、具有良好系统设计与分析能力的计算机高级专业人才在社会上深受欢迎。

计算机科学与技术是一门实践性很强的学科。良好的系统设计和分析能力的培养需要通过长期、系统的训练(包括理论和实践两方面)才能获得。高等学校的实践教学一般包括课程实验、综合性设计(课程设计)、课外科技活动、社会实践、毕业设计等,基本上可以分为三个层次:第一,是紧扣课堂教学内容,以掌握和巩固课程教学内容为主的课程实验和综合性设计;第二,是以社会体验和科学研究体验为主的社会实践和课外科技活动;第三,是以综合应用专业知识和全面检验专业知识应用能力的毕业设计。课程实践(含课程实验和课程设计)是大学教育中最重要也最基础的实践环节,直接影响后继课程的学习以及后继实践的质量。由于课程设计是以培养学生的系统设计与分析能力为目标,通过团队式合作、研究式分析、工程化设计完成较大型系统或软件的设计题目的,因此课程设计不仅有利于学生巩固、提高和融合所学的专业课程知识,更重要的是能够培养学生多方面的能力,如综合能力、动手能力、文献检索能力、团队合作能力、工程化能力、研究性学习能力、创新能力等。

浙江大学计算机学院在专业课程中实施课程设计(project)已有 10 多年的历史,积累了丰富的经验和资料。为全面总结专业课程设计建设的经验,推广建设成果,我们特别组织相关课程的骨干任课教师编写了这套综合实验系列教材。本系列教材的作者们不仅具有丰富的教学和科研经验,而且是浙江大学计算机学院和软件学院的教学核心力量。这支队伍目前已经获得了两门国家精品课程以及四门省部级精品课程,出版了几十部教材。

本套教材由《C 程序设计基础课程设计》、《软件工程课程设计》、《数据结构课程设计》、《数值分析课程设计》、《编译原理课程设计》、《逻辑与计算机设计基础实验与课程设计》、《操作系统课程设计》、《数据库课程设计》、《Java 程序设计课程设计》、《面向对象程序设计课程设计》、《计算机组成课程设计》、《计算机体系结构课程设计》和《计算机图形学课程设计》等十三门课程的综合实验教材所组成。该系列教材构思新颖、案例丰富,许多案例直接取材于作者多年教学、科研以及企业工程经验的积累,适用于作为计算机以及相关专业课程设计的实验教材;也适用于对计算机有浓厚兴趣的专业人士进一步提升计算机的系统设计与分析

能力。从实践的角度出发,大部分教材配备了随书光盘,以方便读者练习。

可以说,本套教材涵盖了计算机专业绝大部分必修课程和部分选修课程,是一套比较完整的专业课程设计系列教材,也是国内第一套由研究型大学计算机学院独立组织编写的专业课程设计系列教材。鉴于书中难免存在的谬误之处,敬请读者指正,以便不断完善。

主编 何钦铭、陈根才

2007年6月于求是园

前 言

数据结构课程是计算机专业最重要的基础课之一,主要研究分析计算机存储、组织数据的方式,使学生学会数据的组织方法和现实世界问题在计算机内部的表示方法,并能针对应用问题,选择合适的数据逻辑结构、存储结构及其算法。

数据结构课程的学习离不开实践。针对数据结构的程序设计实践不仅可以加深对课程内容的理解,更重要的是可以通过实践进一步掌握程序设计的技能与方法,初步感受软件开发过程的项目管理方法与规范,为更进一步的学习打下基础。

数据结构的课程实践可分一般性的实验和综合性的课程设计。在传统的课程教学中,往往使用一般性的实验作为课程实践的主要内容,即向学生布置直接针对课堂教学内容的小型练习题,由学生独立进行程序设计与上机实现;而综合性的课程设计更强调知识的整合、问题分析与求解能力以及团队合作能力的培养。因而,课程设计更能培养学生综合运用所学理论知识解决复杂问题的实践能力、研究性学习能力和团队合作能力。

本教材主要围绕数据结构的基本知识点,设计若干个大型综合性练习案例,通过相关背景知识的回顾、题目解析与实现要点的分析以及测试方法分析等,为学生完成综合性的数据结构实践提供参考。本教材还提供了一系列课程设计题目及其简要的提示。这些案例和题目大多取材于程序设计竞赛题,具有较好的趣味性和技巧性。

全书共分6章。第1章简要介绍了数据结构课程设计的组织与评分方法,并给出了课程设计实验报告的基本内容;第2章至第5章,分别针对栈与树结构、图结构、排序与动态查找、算法设计等4方面的内容,各给出了两个课程设计案例(共8个案例),对每个案例均从基本知识回顾、设计题目、设计分析、设计实现、测试方法、评分要点等几个方面进行了详细的分析;第6章为读者提供了8个课程设计习题,并给出了简要的提示。本书配光盘,光盘内容为第2至第5章的8个课程设计案例的源程序及教材中描述的测试数据。

本教材可作为数据结构课程配套的实验教材,也适合于对C程序设计以及数据结构有初步基础的读者学习数据结构的设计方法和提高编程技巧。

本教材编写过程中得到了研究生车延轍、何安、王小燕的许多帮助,特别是案例的实现与测试。刘耀庭、薛在岳提供了两道精彩的课程设计习题。在此向他们表示衷心的感谢。

由于作者水平所限,对书中存在的谬误之处,敬请读者指正。

作 者

2007年7月

目 录

第 1 章 数据结构课程设计概要	1
1.1 课程设计的意义	1
1.2 课程设计实验报告撰写的基本要求	2
1.3 课程设计的组织与评分方法	3
第 2 章 栈与树结构案例详解	5
2.1 栈的应用:表达式求值	5
2.1.1 基本知识回顾	5
2.1.2 设计题目	8
2.1.3 设计分析	9
2.1.4 设计实现	10
2.1.5 测试方法	23
2.1.6 评分要点	24
2.2 树的遍历:文件目录结构的显示	24
2.2.1 基本知识回顾	24
2.2.2 设计题目	29
2.2.3 设计分析	30
2.2.4 设计实现	31
2.2.5 测试方法	38
2.2.6 评分要点	40
第 3 章 图结构案例详解	41
3.1 最短路径:拯救 007	41
3.1.1 基本知识回顾	41
3.1.2 设计题目	43
3.1.3 设计分析	44
3.1.4 设计实现	46
3.1.5 测试方法	55
3.1.6 评分要点	57

3.2 深度与广度优先搜索:迷宫问题	58
3.2.1 基本知识回顾	58
3.2.2 设计题目	61
3.2.3 设计分析	61
3.2.4 设计实现	63
3.2.5 测试方法	67
3.2.6 评分要点	69
第4章 排序与动态查找案例详解	70
4.1 快速排序详析	70
4.1.1 基本知识回顾	70
4.1.2 设计题目	73
4.1.3 设计分析	73
4.1.4 设计实现	74
4.1.5 测试方法	78
4.1.6 评分要点	81
4.2 散列表的应用:插队买票	82
4.2.1 基本知识回顾	82
4.2.2 设计题目	85
4.2.3 设计分析	87
4.2.4 设计实现	89
4.2.5 测试方法	94
4.2.6 评分要点	95
第5章 算法设计案例详解	96
5.1 算法分析:搜索算法效率比较	96
5.1.1 基本知识回顾	96
5.1.2 设计题目	98
5.1.3 设计分析	99
5.1.4 设计实现	101
5.1.5 测试方法	103
5.1.6 评分要点	104
5.2 贪心算法:任务调度问题	105
5.2.1 基本知识回顾	105
5.2.2 设计题目	108
5.2.3 设计分析	109
5.2.4 设计实现	111
5.2.5 测试方法	113
5.2.6 评分要点	114

第 6 章 课程设计习题	115
6.1 二叉搜索树:各种搜索树效率比较	115
6.1.1 题目要求	115
6.1.2 分工要求	115
6.1.3 简要提示	115
6.2 并查集:检查网络	116
6.2.1 题目要求	116
6.2.2 分工要求	117
6.2.3 简要提示	117
6.3 网络流:宇宙旅行	117
6.3.1 题目要求	117
6.3.2 分工要求	118
6.3.3 简要提示	119
6.4 最小生成树:室内布线	119
6.4.1 题目要求	119
6.4.2 分工要求	120
6.4.3 简要提示	120
6.5 分治法:最小套圈设计	121
6.5.1 题目要求	121
6.5.2 分工要求	122
6.5.3 简要提示	122
6.6 动态规划:商店购物	123
6.6.1 题目要求	123
6.6.2 分工要求	125
6.6.3 简要提示	125
6.7 熊猫烧香	126
6.7.1 题目要求	126
6.7.2 分工要求	127
6.7.3 简要提示	128
6.8 神秘国度的爱情故事	129
6.8.1 题目要求	129
6.8.2 分工要求	129
6.8.3 简要提示	130
附录 课程设计实验报告样例	132
参考文献	141

第1章

数据结构课程设计概要

1.1 课程设计的意义

数据结构是计算机存储、组织数据的方式。选择合适的数据结构更容易设计出更高运行或存储效率的算法;反之,选择了特定的算法后也需要设计合适的数据结构与之配合,以达到最佳效果。所以,在进行程序设计时必须将数据结构和与之相关的算法结合起来考虑。

数据结构课程是计算机专业最重要的基础课之一。通过学习,学生应掌握解决复杂问题的程序设计方法和技术,即学会数据的组织方法和现实世界问题在计算机内部的表示方法,并针对问题的应用背景分析,选择最佳的数据结构与算法。

计算机科学与技术是个实践性很强的专业。数据结构专业基础课程的学习同样也离不开实践。针对数据结构的程序设计实践不仅可以加深对课程内容的理解,更重要的是可以通过实践进一步掌握程序设计的技能与方法,初步感受软件开发过程的项目管理方法与规范,为更进一步的学习打下基础。

数据结构的课程实践可分一般性的实验和综合性的课程设计。在传统的课程教学中,往往使用一般性的实验作为课程实践的主要内容,即向学生布置直接针对课堂教学内容的小型练习题,由学生独立进行程序设计与上机实现。而综合性的课程设计则更强调知识的整合、问题分析与求解能力以及团队合作能力的培养。因而,课程设计更能培养学生综合运用所学理论知识解决复杂问题的实践能力、研究性学习能力和团队合作能力。

本教材主要围绕数据结构的基本知识点,设计若干个大型综合性练习案例,通过相关背景知识的回顾、题目解析与实现要点的分析以及测试方法分析等,为学生完成综合性的数据结构实践提供参考。本教材还提供了一系列课程设计题目及其简要的提示。这些案例和题目大多取材于程序设计竞赛题,具有较好的趣味性和技巧性。如果读者想更多地了解这类题目,可访问浙江大学程序设计竞赛网站(<http://acm.zju.edu.cn>)。

课程设计不仅仅是以实现相应的程序为目标,更重要的是在完成课程设计的过程中逐步培养今后从事软件开发所需要的各种能力与素质。其中很重要的一种能力就是软件文档的写作能力。因此,在课程设计实施中,不仅需要完成程序并进行测试,还需要撰写相应的课程设计实验报告。课程设计实验报告不仅是对课程设计的总结,也是对软件文档写作能力的初步训练。

1.2 课程设计实验报告撰写的基本要求

每个课程设计对学生的要求不仅仅是编写代码,而且还要按照科学论文的基本要求完成一篇完整的实验报告,从而全面锻炼学生做研究与设计的总结能力。

实验报告首先要求有一个清晰醒目的报告标题,例如:《数据结构课程设计实验一:搜索算法效率比较》。此外,至少要求具备以下6部分内容。

一、简介

这一部分需简单介绍题目内容,即该实验到底要做什么。如果涉及明确的算法,最好再简单介绍一下算法产生的背景。

基本要求:实验内容必须完全覆盖。

二、算法说明

这一部分需详细描述解决问题所需要用到的算法和重要的数据结构,即该实验到底应该怎么做。

基本要求:处理问题中所用到的关键算法都要描述清楚,而不是仅描述主函数。算法和数据结构可用伪码和图示描述,不要只写源代码和注释。

这一部分的目的是让读者在短时间内清楚地理解作者解决问题的整体思路,表达方式必须比源代码更通俗易懂。如果读者感觉还不如直接读源代码来得明白,这一部分内容就失去了意义。

三、测试结果

这一部分内容需要紧扣课程设计的题目类型和要求,设计提供相应的测试方法和结果。

对于需要比较不同算法性能优劣的题目,应设计并填写一张性能比较表格,列出不同算法在同一指标下的性能表现。仅仅罗列出一堆数据是不够的,还应将数字转化为图形、曲线等方式,帮助读者更直观地理解测试结果。

对于需要利用某算法解决某问题的题目,应设计并填写一张测试用例表。每个测试用例一般应包括下列内容:

- 测试输入:设计一组输入数据;
- 测试目的:设计该输入的目的在于测试程序在哪方面可能存在漏洞;
- 正确输出:对应该输入,若程序正确,应该输出的内容;
- 实际输出:该数据输入后,实际测试得到的输出内容;
- 错误原因:如果实际输出与正确输出不符,需分析产生错误的可能原因;
- 当前状态:分为“通过”(实际输出与正确输出相符)、“已改正”(实际输出与正确输出不符,但现在已修改正确)、“待修改”(实际输出与正确输出不符,且尚未改正)三种状态。

需要注意的是,测试员的态度,不是提供几组简单的数据让程序员容易通过,从而宣称该程序是正确的;而应该是千方百计设计“刁难”的数据,想办法让所测试的程序暴露出问

题,这样才能真正帮助程序员完成正确的程序,最后通过严格的裁判数据测试。

四、分析与探讨

这一部分应是整篇报告中最令读者感兴趣的部分,分为以下两方面内容:

- 测试结果分析。需详细解释测试策略,对得到的数据进行分析,总结出算法的时空复杂度,得出自己对算法性能等方面分析的结论。
- 不局限于题目要求使用的算法,探讨更多解决问题的途径,或者提出自己的见解,给出改进算法以得到更好结果的建议。

附录:源代码

源代码列在附录中,要求程序风格清晰易理解,有充分的注释。有意义的注释行少于代码的30%将不能得分。

任务分配:

- 程序员:×××
- 测试员:×××
- 文档员:×××

这一部分说明合作完成课程设计的程序员,测试员和文档员的姓名及主要任务。

实验报告完成日期:yyyy-mm-dd

请注意,以上只是课程设计实验报告的基本要求,不同的题目还会有不同的具体要求。在本教材的附录中,提供了一个课程设计实验报告的样本,供读者参考。

1.3 课程设计的组织与评分方法

课程设计实验的每一道题目建议按照3个人一组,课后4个学时左右完成设计。每题的总分为100分,可以按如下比例分配给3个人。

- 程序员:50分,负责完成源代码。
- 测试员:30分,负责设计测试用程序、产生充分测试数据,最后完成实验报告的第三、四部分内容,即测试结果与分析探讨部分。
- 文档员:20分,负责撰写实验报告的第一、二部分内容,即实验内容简介、算法与数据结构描述。同时完成整个文档的整合,使整篇报告排版、文字风格统一,而不是对3个人工作的简单拼凑。

建议对整门课程发布3或6个课程设计实验题目,使每个人都有机会把3项不同的工作做1~2遍,以达到全面锻炼能力的目的。

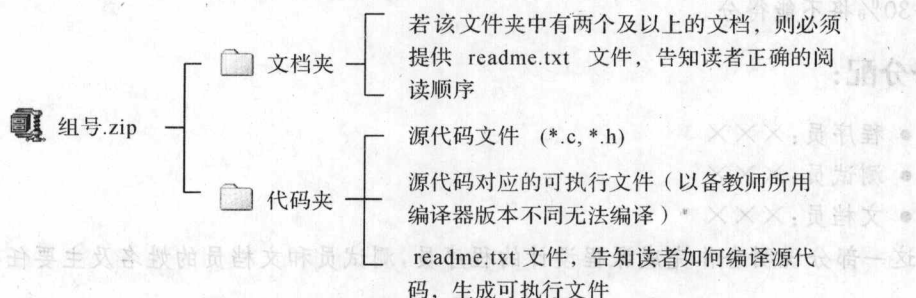
由于每个人的能力水平高低不同,为公平起见,应使组内每个人的成绩相对独立,不依赖于其他组员的表现而得分。例如,程序员只要正确完成了自己的程序,且注释充分,就可

以得满分;测试员的测试计划和测试用例设计都不依赖于源代码,即使程序员无法提供程序,但测试员只要完成了测试计划的设计,对算法进行了理论上的分析,就可以得到满分;文档员的工作相对独立,只要完成了自己负责的部分,并将手头所有资料整合为一体,就可以获得满分。

不过,在程序员提供了源代码的情况下,测试员的分数跟程序员的分数是相关的——若程序有漏洞,被裁判(如教师)发现却没有被测试员发现,则测试员要和程序员一起被扣分;若该漏洞已经被测试员发现,而程序员没有改正,则只有程序员被扣分。

当然,分组实验的另一个目的是锻炼学生的沟通与团队合作能力。这个目的可以通过教师的激励来达到,即教师在点评每道题目的完成情况时,对个别优秀的组提出表扬,而只有3项工作都做得完美的组,才会获此殊荣。

学生提交的全部资料,建议按如下方式打包:



每道题目发布时,应有明确的截止日期。对超过截止日期的提交,应有相应的迟交罚扣。例如,每迟交24小时罚扣应得分数的10%。

第2章

栈与树结构案例详解

线性表、树结构和图结构是三类最主要的数据结构。在线性表数据结构中,最常用、最典型的是栈和队列。本章通过表达式求值和文件系统的目录结构显示这两个课程设计方案分别回顾了栈与树结构的基本知识和基本方法,并对两个课程设计的具体设计与实现进行了详细的分析。

2.1 栈的应用:表达式求值

2.1.1 基本知识回顾

1. 栈的定义

栈(Stack)又称堆栈,它是一种运算受限的线性表,之所以受限是因为仅允许在表的一端进行插入和删除。人们把此端(一般为表的末端)称为栈顶(top),另一端则称为栈底(bottom),栈顶的第一个元素被称为栈顶元素。对一个栈最基本的操作包括进栈(Push)和出栈(Pop),即对栈的插入和删除。因为栈只能在栈顶进行进栈和出栈的操作,即栈的修改是按照后进先出的原则进行的,所以栈又称为后进先出表(Last In First Out, LIFO)。

不含任何数据元素的栈称为空栈,对一个空栈进行出栈操作会产生错误;同样,对一个没有剩余空间的满栈进行进栈操作也会产生错误。

2. 栈的抽象数据类型

栈的抽象数据类型(Abstract Data Type, ADT)包括数据部分和操作部分。栈的数据部分可以采用顺序存储结构或链接存储结构;而操作部分包括元素进栈、出栈、栈的初始化、读取栈顶元素、检查栈是否为空等。为了简单起见,在本章中,栈的数据部分的元素类型都以 int 为例。下面给出栈的抽象数据类型的定义:

ADT Stack {

数据对象:

采用顺序或链接方式存储的栈

基本操作:

Stack CreateStack()

操作结果:构造一个空栈 S

int IsEmpty(Stack S)

初始条件:栈 S 已存在

操作结果:若 S 为空栈,则返回 TRUE,否则返回 FALSE

```
int IsFull(Stack S)
```

初始条件:栈 S 已存在

操作结果:若 S 已满,则返回 TRUE,否则返回 FALSE

```
int Pop(Stack S)
```

初始条件:栈 S 已存在且非空

操作结果:删除 S 的栈顶元素,并返回其值

```
void Push(int X, Stack S)
```

初始条件:栈 S 已存在且不满

操作结果:插入元素 X 为新的栈顶元素

```
}ADT Stack
```

3. 栈的存储结构

由于栈是线性表,因而栈的存储结构可采用顺序和链式两种形式。顺序存储的栈称为顺序栈,链式存储的栈称为链栈。

(1)栈的顺序存储结构:通常由一个一维数组和一个记录栈顶元素位置的变量组成。习惯上将栈底放在数组下标小的那端。假设用一维数组 $S[\text{MaxSize}]$ (下标 $0 \sim \text{MaxSize}-1$) 表示一个栈, MaxSize 为栈中可存储数据元素的最大个数,即栈的最大长度。栈顶位置用一个整型变量 top 记录当前栈顶元素的下标值。当 top 指向 -1 时,表示空栈。当 top 指向 $\text{MaxSize}-1$ 时,表示满栈。用 C 语言描述顺序栈如下:

```
#define MaxSize<储存数据元素的最大个数>
typedef struct {
    int data[MaxSize];
    int top;
}Stack;
```

(2)栈的链式存储结构:栈的链式实现以某种形式的链表作为栈的存储结构。链栈的组织形式与单链表类似,但其运算受限制,插入和删除运算只能在链栈的栈顶进行。栈顶指针也就是链表的头指针,如图 2.1 所示。由于 C 函数参数值传递的特点,为了使 Push 和 Pop 操作能方便地处理栈空与不空两种情况,链栈通常带一表头节点。表头节点后面的第一个节点就是链栈栈顶节点, top 称为栈顶指针,它唯一地确定一个链栈。栈中的其他节点通过它们的 next 链接起来。栈底节点的 next 为 NULL。

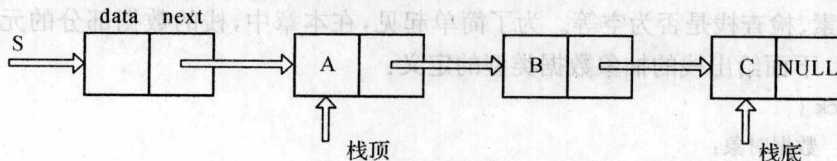


图 2.1 链栈示意图

用 C 语言描述链栈如下:

```
typedef struct Node{
    int data;
    struct Node * next;
```

```

}LinkStack;
LinkStack * Stack;

```

4. 栈运算实现

上文介绍了栈的两种存储实现方法,分别是顺序存储结构和链式存储结构。针对这两种结构,栈的运算实现是不同的。下文给出了这两种栈最主要的栈操作实现方式,例如 Pop 和 Push 等。

(1) 栈的操作在顺序存储结构上的实现。

```

int Pop(Stack * S)                                /* 删除栈顶元素,并将其返回 */
{
    if(S->top == -1)
    {
        Error("Stack is empty!"); /* 若栈为空,则终止运行 */
        exit(1);
    }
    return S->data[S->top--]; /* 返回原栈顶元素的值,并将栈顶前移一个位置 */
}

```

```

void Push(int X, Stack * S) /* 元素 X 进栈 */
{
    if(S->top == MaxSize - 1)
    {
        Error("Stack overflow!"); /* 若栈已满,则终止运行 */
        exit(1);
    }
    S->data[++(S->top)] = X; /* 栈顶后移一个位置,并将 X 的值赋给新的栈顶位置 */
}

```

(2) 栈的操作在链式存储结构上的实现。

```

LinkStack * CreateStack() /* 初始化栈,并将其置为空 */
{

```

```

    LinkStack * S;
    S = (LinkStack *) malloc(sizeof(LinkStack));
    if(S == NULL)
        FatalError("Out of space!"); /* 空间不足 */
    S->next = NULL;
    return S;
}

```

```

int Pop(LinkStack * S) /* 删除栈顶元素,并返回之 */
{
    LinkStack * FirstCell;
    if(S->next == NULL)
        Error("Empty stack!"); /* 若栈为空,则终止运行 */
}

```