

高等学校教材·计算机科学与技术

可下载教学资料

<http://www.tup.tsinghua.edu.cn>

# 数据结构教程 上机实验指导

李春葆 编著



清华大学出版社

高等学校教材·计算机科学与技术

# 数据结构教程上机实验指导

李春葆 编著

清华大学出版社  
北京

## 内 容 简 介

本书是《数据结构教程》的配套上机实验指导书,两书章次一一对应,内容包括线性表、栈和队列、串、数组和稀疏矩阵、递归、树状结构、广义表、图、查找、内排序、外排序、文件等。书后附录中给出了VC++ 6.0环境下编写C程序所需的基本知识及学生应提交的实验报告的格式。本书包含全部上机实验题目,因此自成一体,可脱离《数据结构教程》单独使用。

本书适合高等院校计算机及其他相关系本科及研究生使用。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

### 图书在版编目(CIP)数据

数据结构教程上机实验指导/李春葆编著. —北京:清华大学出版社,2005.7

(高等学校教材·计算机科学与技术)

ISBN 7-302-10826-9

I. 数… II. 李… III. 数据结构—高等学校—教学参考资料 IV. TP311.12

中国版本图书馆CIP数据核字(2005)第033443号

出版者:清华大学出版社 地 址:北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编:100084

社总机:010-62770175 客户服务:010-62776969

组稿编辑:丁 岭

文稿编辑:孙建春

印装者:清华大学印刷厂

发行者:新华书店总店北京发行所

开 本:185×260 印张:20.5 字数:484千字

版 次:2005年7月第1版 2005年7月第1次印刷

书 号:ISBN 7-302-10826-9/TP·7203

印 数:1~3000

定 价:26.00元

高等学校教材·计算机科学与技术

编审委员会成员

(按地区排序)

清华大学	周立柱	教授
	覃征	教授
	王建民	教授
	刘强	副教授
	冯建华	副教授
北京大学	杨冬青	教授
	陈钟	教授
	陈立军	副教授
北京航空航天大学	马殿富	教授
	吴超英	副教授
	姚淑珍	教授
中国人民大学	王珊	教授
	孟小峰	教授
	陈红	教授
北京交通大学	阮秋琦	教授
北京信息工程学院	孟庆昌	教授
北京科技大学	杨炳儒	教授
石油大学	陈明	教授
天津大学	艾德才	教授
复旦大学	吴立德	教授
	吴百锋	教授
	杨卫东	副教授
华东理工大学	邵志清	教授
华东师范大学	杨宗源	教授
	应吉康	教授
东华大学	乐嘉锦	教授
上海第二工业大学	蒋川群	教授
浙江大学	吴朝晖	教授
	李善平	教授
南京大学	骆斌	教授
南京航空航天大学	秦小麟	教授
南京理工大学	张功萱	教授

南京邮电学院	朱秀昌	教授
苏州大学	龚声蓉	教授
江苏大学	宋余庆	教授
武汉大学	何炎祥	教授
华中科技大学	刘乐善	教授
中南财经政法大学	刘腾红	教授
华中师范大学	王林平	副教授
	魏开平	教授
武汉理工大学	李中年	教授
国防科技大学	赵克佳	教授
	肖 依	副教授
中南大学	陈松乔	教授
湖南大学	林亚平	教授
	邹北骥	教授
西安交通大学	沈钧毅	教授
	齐 勇	教授
西北大学	周明全	教授
长安大学	巨永峰	教授
西安石油学院	方 明	教授
西安邮电学院	陈莉君	副教授
哈尔滨工业大学	郭茂祖	教授
吉林大学	徐一平	教授
	毕 强	教授
长春工程学院	沙胜贤	教授
山东大学	孟祥旭	教授
	郝兴伟	教授
山东科技大学	郑永果	教授
中山大学	潘小轰	教授
厦门大学	冯少荣	教授
福州大学	林世平	副教授
云南大学	刘惟一	教授
重庆邮电学院	王国胤	教授
西南交通大学	杨 燕	副教授

改革开放以来,特别是党的十五大以来,我国教育事业取得了举世瞩目的辉煌成就,高等教育实现了历史性的跨越,已由精英教育阶段进入国际公认的大众化教育阶段。在质量不断提高的基础上,高等教育规模取得如此快速的发展,创造了世界教育发展史上的奇迹。当前,教育工作既面临着千载难逢的良好机遇,同时也面临着前所未有的严峻挑战。社会不断增长的高等教育需求同教育供给特别是优质教育供给不足的矛盾,是现阶段教育发展面临的基本矛盾。

教育部一直十分重视高等教育质量工作。2001年8月,教育部下发了《关于加强高等学校本科教学工作,提高教学质量的若干意见》,提出了十二条加强本科教学工作提高教学质量的措施和意见。2003年6月和2004年2月,教育部分别下发了《关于启动高等学校教学质量与教学改革工程精品课程建设工作的通知》和《教育部实施精品课程建设提高高校教学质量和人才培养质量》文件,指出“高等学校教学质量和教学改革工程”是教育部正在制订的《2003—2007年教育振兴行动计划》的重要组成部分,精品课程建设是“质量工程”的重要内容之一。教育部计划用五年时间(2003—2007年)建设1500门国家级精品课程,利用现代化的教育信息技术手段将精品课程的相关内容上网并免费开放,以实现优质教学资源共享,提高高等学校教学质量和人才培养质量。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学

科体系有实质性的改革和发展、顺应并符合新世纪教学发展的规律、代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。首批推出的特色精品教材包括:

(1) 高等学校教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 高等学校教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 高等学校教材·电子信息——高等学校电子信息相关专业的教材。

(4) 高等学校教材·软件工程——高等学校软件工程相关专业的教材。

(5) 高等学校教材·信息管理与信息系统

清华大学出版社经过近 20 年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材经过 20 多年的精雕细刻,形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

E-mail: dingl@tup.tsinghua.edu.cn

**本**书是《数据结构教程》(以下简称为《教程》)的配套上机实验指导书。

全书分为 14 章,内容均为上机实验题解析。第 1 章为绪论;第 2 章为线性表;第 3 章为栈和队列;第 4 章为串;第 5 章为数组和稀疏矩阵;第 6 章为递归;第 7 章为树状结构;第 8 章为广义表;第 9 章为图;第 10 章为查找;第 11 章为内排序;第 12 章为外排序;第 13 章为文件;第 14 章为综合实验题。各章次与《教程》的章次相对应。

另外,书后给出了两个附录,附录 A 较系统地给出在 VC++ 6.0 环境下编写 C 程序所需要的基本知识。附录 B 为学生应提交的实验报告的格式。

书中所有程序都在 VC++ 6.0 环境下调试通过。

书中列出了全部的上机实验题目,因此自成一体,可以脱离《教程》单独使用。

由于水平所限,尽管编者不遗余力,仍可能存在错误和不足之处,敬请教师 and 同学们批评指正。

编 者

2004.7.30 于珞珈山



<b>第 1 章 绪论——上机实验题 1 解析</b> .....	1
实验题 1.1 求素数 .....	1
实验题 1.2 求一个正整数的各位数字之和 .....	2
实验题 1.3 求一个字符串是否为回文 .....	4
<b>第 2 章 线性表——上机实验题 2 解析</b> .....	6
实验题 2.1 实现顺序表各种基本运算的算法 .....	6
实验题 2.2 实现单链表各种基本运算的算法 .....	10
实验题 2.3 实现双链表各种基本运算的算法 .....	15
实验题 2.4 实现循环单链表各种基本运算的算法 .....	21
实验题 2.5 实现循环双链表各种基本运算的算法 .....	26
实验题 2.6 求集合(用有序单链表表示)的并、交和差运算 .....	33
实验题 2.7 求两个多项式相加运算 .....	37
<b>第 3 章 栈和队列——上机实验题 3 解析</b> .....	41
实验题 3.1 实现顺序栈各种基本运算的算法 .....	41
实验题 3.2 实现链栈各种基本运算的算法 .....	44
实验题 3.3 实现顺序队列各种基本运算的算法 .....	48
实验题 3.4 实现链队各种基本运算的算法 .....	52
实验题 3.5 求解迷宫问题的所有路径及最短路径程序 .....	56
实验题 3.6 病人看病模拟程序 .....	59
实验题 3.7 停车场管理程序 .....	62
<b>第 4 章 串——上机实验题 4 解析</b> .....	69
实验题 4.1 实现顺序串各种基本运算的算法 .....	69

实验题 4.2	实现链串各种基本运算的算法 .....	74
实验题 4.3	顺序串的各种模式匹配运算 .....	81
实验题 4.4	文本串加密和解密程序 .....	85
实验题 4.5	求一个串中出现的第一个最长重复子串 .....	87
<b>第 5 章</b>	<b>数组和稀疏矩阵——上机实验题 5 解析 .....</b>	<b>90</b>
实验题 5.1	求 $5 \times 5$ 阶螺旋方阵 .....	90
实验题 5.2	求一个矩阵的马鞍点 .....	92
实验题 5.3	求两个对称矩阵之和与乘积 .....	94
实验题 5.4	实现稀疏矩阵(采用三元组表示)的基本运算 .....	97
<b>第 6 章</b>	<b>递归——上机实验题 6 解析 .....</b>	<b>103</b>
实验题 6.1	输出求解 $akm(2,1)$ 的递推和求值过程 .....	103
实验题 6.2	求解 $n$ 皇后问题 .....	104
实验题 6.3	求解背包问题 .....	107
实验题 6.4	采用递归和非递归方法求解 $F(n)$ .....	110
<b>第 7 章</b>	<b>树状结构——上机实验题 7 解析 .....</b>	<b>113</b>
实验题 7.1	实现二叉树各种基本运算的算法 .....	113
实验题 7.2	实现二叉树各种遍历算法 .....	119
实验题 7.3	求二叉树中从根结点到叶子结点的路径 .....	126
实验题 7.4	由遍历序列构造二叉树 .....	130
实验题 7.5	实现中序线索化二叉树 .....	134
实验题 7.6	构造哈夫曼树 .....	138
实验题 7.7	用二叉树来表示代数表达式 .....	142
<b>第 8 章</b>	<b>广义表——上机实验题 8 解析 .....</b>	<b>145</b>
实验题 8.1	实现广义表的各种基本运算算法 .....	145
实验题 8.2	实现广义表求表头和表尾的运算 .....	149
实验题 8.3	实现广义表的相关运算算法(一) .....	152
实验题 8.4	实现广义表的相关运算算法(二) .....	156
<b>第 9 章</b>	<b>图——上机实验题 9 解析 .....</b>	<b>162</b>
实验题 9.1	实现图的邻接矩阵和邻接表存储 .....	162
实验题 9.2	实现图的遍历算法 .....	166
实验题 9.3	求有向图的简单路径 .....	170

实验题 9.4 求无向图中满足约束条件的路径 .....	175
实验题 9.5 求无向图的深度优先生成树和广度优先生成树 .....	178
实验题 9.6 采用普里姆算法求最小生成树 .....	181
实验题 9.7 采用克鲁斯卡尔算法求最小生成树 .....	184
实验题 9.8 采用狄克斯特拉算法求有向带权图的最短路径 .....	187
实验题 9.9 采用弗洛伊德算法求有向带权图的最短路径 .....	190
<b>第 10 章 查找——上机实验题 10 解析 .....</b>	<b>194</b>
实验题 10.1 实现顺序查找的算法 .....	194
实验题 10.2 实现二分查找的算法 .....	195
实验题 10.3 实现分块查找的算法 .....	197
实验题 10.4 实现二叉排序树的基本运算算法 .....	200
实验题 10.5 统计一个字符串中出现的字符及其次数 .....	205
实验题 10.6 实现二叉平衡树的相关运算算法 .....	207
实验题 10.7 实现 B-树的相关运算算法 .....	216
实验题 10.8 实现哈希表的相关运算算法 .....	225
<b>第 11 章 内排序——上机实验题 11 解析 .....</b>	<b>230</b>
实验题 11.1 实现直接插入排序算法 .....	230
实验题 11.2 实现希尔插入排序算法 .....	232
实验题 11.3 实现冒泡排序算法 .....	234
实验题 11.4 实现快速排序算法 .....	235
实验题 11.5 实现直接选择排序算法 .....	237
实验题 11.6 实现堆排序算法 .....	239
实验题 11.7 实现二路归并排序算法 .....	242
实验题 11.8 实现基数排序算法 .....	244
实验题 11.9 实现可变长度的字符串序列快速排序算法 .....	247
实验题 11.10 实现英文单词按字典序排列的基数排序算法 .....	250
<b>第 12 章 外排序——上机实验题 12 解析 .....</b>	<b>254</b>
实验题 12.1 实现置换-选择算法 .....	254
<b>第 13 章 文件——上机实验题 13 解析 .....</b>	<b>260</b>
实验题 13.1 实现索引文件建立和查找算法 .....	260
<b>第 14 章 综合实验题解析 .....</b>	<b>267</b>
综合实验题 14.1 链表综合算法设计 .....	267

综合实验题 14.2 求复杂表达式的值 .....	273
综合实验题 14.3 用二叉树实现家谱的相关运算 .....	287
<b>附录</b> .....	296
附录 A 使用 VC++ 6.0 系统 .....	296
附录 B 实验报告格式 .....	315

## 绪论——上机实验题 1 解析

## 实验题 1.1 求素数

设计一个程序,输出所有小于等于  $n$  ( $n$  为一个大于 2 的正整数)的素数。要求:(1)每行输出 10 个素数;(2)尽可能采用较优的算法。

**解:** 本项目 proj1\_1 的组成结构如图 1.1 所示。本程序的模块结构图如图 1.2 所示,图中方框表示函数,方框中指出函数名,箭头方向表示函数间的调用关系,虚线方框表示文件的组成,即指出该虚线方框中的函数存放在哪个文件中。

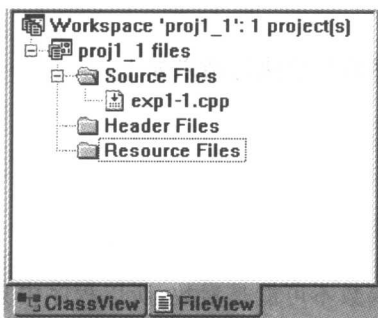


图 1.1 proj1\_1 项目组成

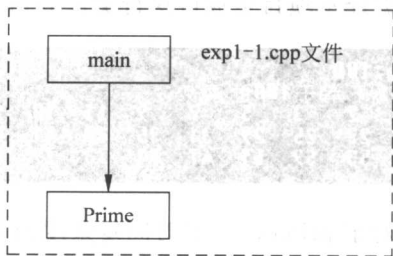


图 1.2 proj1\_1 项目的程序结构图

exp1-1.cpp 文件包含如下函数。

Prime( $x$ ): 其功能是判断正整数  $x$  是否为素数。采用的方法是,若  $x$  是素数,则  $x$  不能被  $2 \sim \sqrt{x}$  的任何数整除。

函数代码如下(设计思路详见代码中的注释):

```
/* 文件名: exp1-1.cpp */
#include <stdio.h>
#include <math.h>
int prime(int x)
{
    /* 判断正整数 x 是否为素数 */
```

```

int i;
for (i = 2; i < (int)sqrt(x); i++)
    if (x % i == 0) return 0;          /* 若 x 不是素数,则退出并返回 0 */
return 1;
}
void main()
{
    int n,i,j=0;                       /* j 用于累计素数个数 */
    printf("n: ");
    scanf("%d",&n);
    printf("小于等于 %d 的素数: \n",n);
    if (n>2)
    { printf("%4d",2);
      j++;
    }
    for (i = 3; i <= n; i += 2)
        if (prime(i) == 1)
        { printf("%4d",i);
          if (j != 0 && ++j % 10 == 0)    /* 每行最多显示 10 个素数 */
              printf("\n");
        }
    printf("\n");
}

```

编译本项目生成可执行文件 proj1\_1.exe。程序的一次执行结果如下：

```

n:100
小于等于100的素数:
 2  3  5  7  9 11 13 15 17 19
23 25 29 31 35 37 41 43 47 49
53 59 61 67 71 73 79 83 89 97

```

对于  $\text{prime}(x)$ , 其时间复杂度为  $O(\sqrt{x})$ , 由于偶数不可能是素数, 所以程序中只对奇数进行素数的判断。因此, 上述程序的时间复杂度较低。

## 实验题 1.2 求一个正整数的各位数字之和

编写一个程序, 计算任一输入的正整数的各位数字之和, 并分析算法的时间复杂度。

**解:** 本项目 proj1\_2 的组成结构如图 1.3 所示。本程序的模块结构图如图 1.4 所示, 图中方框表示函数, 方框中指出函数名, 箭头方向表示函数间的调用关系, 虚线方框表示文件的组成, 即指出该虚线方框中的函数存放在哪个文件中。

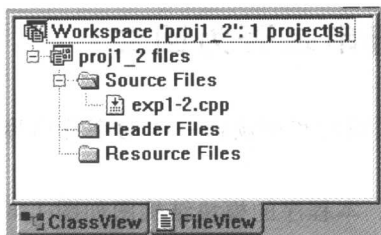


图 1.3 proj1\_2 项目组成

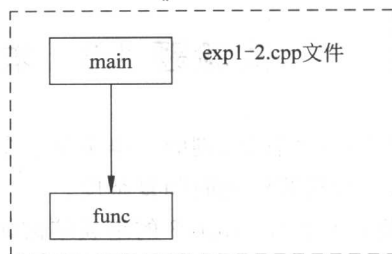


图 1.4 proj1\_2 项目的程序结构图

exp1-2.cpp 文件包含如下函数。

int func(int num): 分解 num 的各位数字, 返回这些数字之和。采用的方法是对 num 边分解边进行累加, 直到分解完毕。

函数代码如下(设计思路详见代码中的注释):

```

/* 文件名: exp1-2.cpp */
#include <stdio.h>
int func(int num)                                     /* 分解 num 的各位数字, 返回其和 */
{
    int s = 0;
    do
    {
        s + = num % 10;                                /* 累计各位数字之和 */
        num / = 10;                                    /* 求下一数值位 */
    } while(num);
    return(s);
}
void main()
{
    int n;
    printf("\n");
    printf("输入一个整数: ");
    scanf("% d", &n);
    printf("各位数字之和: % d\n", func(n));
    printf("\n");
}

```

编译本项目生成可执行文件 proj1\_2.exe。程序的一次执行结果如下:

```

输入一个整数:12345678
各位数字之和:36

```

func( $n$ )的时间复杂度为  $O(\text{len}(n))$ ,  $\text{len}(n)$  为正整数  $n$  的位数。程序的时间复杂度也为  $O(\text{len}(n))$ 。

### 实验题 1.3 求一个字符串是否为回文

编写一个程序,判断一个字符串是否为“回文”(顺读和倒读都一样的字符串称为“回文”),并分析算法的时间复杂度。

解:本项目 proj1\_3 的组成结构如图 1.5 所示。本程序的模块结构图如图 1.6 所示,图中方框表示函数,方框中指出函数名,箭头方向表示函数间的调用关系,虚线方框表示文件的组成,即指出该虚线方框中的函数存放在哪个文件中。

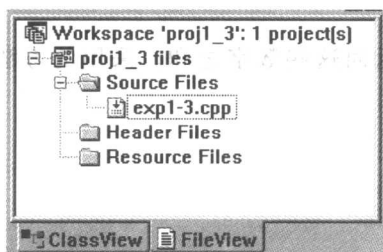


图 1.5 proj1\_3 项目组成

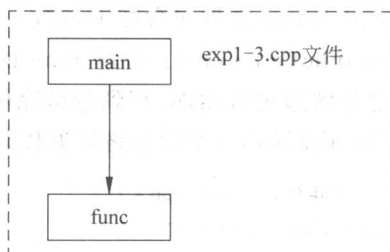


图 1.6 proj1\_3 项目的程序结构图

exp1-3.cpp 文件包含如下函数。

func(s): 判断串 s 是否为回文。采用的方法是:用 flag 表示是否为回文,其初值为 1。用 i 从左向右扫描字符串 s,用 j 从右向左扫描字符串 s,若 s[i]与 s[j]不相等,则 flag=0(表示不是回文)并退出循环,否则,继续比较直到 i<j 不成立。

函数代码如下(设计思路详见代码中的注释):

```

/* 文件名: exp1-3.cpp */
#include <stdio.h>
#include <string.h>
#define MAX 100
int func(char s[])
{
    int flag = 1;
    int i, j, slen = strlen(s);
    for (i = 0, j = slen - 1; i < j; i++, j--)
        if (s[i] != s[j])
        {
            flag = 0;
            break;
        }
    return(flag);
}

void main()
{
    char s[MAX];
    /* 字符串的最大长度 */
    /* 判断串 s 是否为回文 */
    /* slen 为字符串 s 的长度 */
}

```



```
printf("输入一字符串:");  
scanf("%s",s);  
if (func(s) == 1)  
    printf("%s 字符串是回文\n",s);  
else  
    printf("%s 字符串不是回文\n",s);  
}
```

编译本项目生成可执行文件 proj1\_3.exe。程序的一次执行结果如下：

```
输入一字符串:abcdba  
abcdba字符串是回文
```

在 func(s)算法中,for 循环语句的执行次数为  $\frac{n}{2}$  ( $n$  为字符串  $s$  的长度),则它的时间复杂度为  $O(n)$ 。程序的时间复杂度亦为  $O(n)$ 。