

新世纪计算机专业系列教材

数据结构 C++实现

习题解析与实验指导

缪淮扣 沈俊 顾训穰 编著

 科学出版社
www.sciencep.com

新世纪计算机专业系列教材

数据结构——C++ 实现

习题解析与实验指导

缪淮扣 沈俊 顾训穰 编著

科学出版社

北京

内 容 简 介

本书是与主教材《数据结构——C++实现》(科学出版社)配套的教学参考书,旨在指导、启发和帮助学生学好数据结构这门课程。本书对主教材的每一章给出了复习提要,并给出了主教材中全部习题的参考答案和分析。本书为主教材的每一章设计了一套上机实习题,并提供了可在计算机上运行的上机实习的实例。

此外,我们还对数据结构考试的题型做了介绍,并给出了3套模拟试卷。

本书是高等院校计算机专业“数据结构”课程的教学参考书,也可供参加硕士研究生入学考试的考生以及从事计算机开发和应用的工程技术人员阅读、参考。

图书在版编目(CIP)数据

数据结构——C++实现习题解析与实验指导 / 缪准扣等编著. —北京:科学出版社,2005

(新世纪计算机专业系列教材)

ISBN 7-03-015441-X

I. 数… II. 缪… III. ①数据结构-高等学校-教学参考资料②C语言-程序设计-高等学校-教学参考资料 N. ①TP311.12②TP312

中国版本图书馆 CIP 数据核字(2005)第 042766 号

责任编辑:陈晓萍 / 责任校对:刘彦妮

责任印制:吕春珉 / 封面设计:王 浩

科学出版社出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

新蕾印刷厂印刷

科学出版社发行 各地新华书店经销

*

2005年7月第一版 开本:787×1092 1/16

2005年7月第一次印刷 印张:15 3/4

印数:1—3 000 字数:367 000

定价:22.00元

(如有印装质量问题,我社负责调换(路通))

销售部电话 010-62136131 编辑部电话 010-62138978-8001(HI06)

新世纪计算机专业系列教材

编委会

顾问编委

施伯乐教授	复旦大学
白英彩教授	上海交通大学

主 任

左孝凌 教授	上海交通大学
--	--------

编 委

刘 璟教授	南开大学
宋方敏教授	南京大学
何炎祥教授	武汉大学
余雪丽教授	太原理工大学
阮家栋教授	上海工程技术大学
顾训穰教授	上海大学
徐汀荣教授	苏州大学
曾 明教授	西安交通大学
曹元大教授	北京理工大学
曹文君教授	复旦大学
陶树平教授	同济大学
缪淮扣教授	上海大学
谢康林教授	上海交通大学

总 序

20年来,计算机学科的发展日新月异,促使现代科学在各个领域突飞猛进。目前,计算机科学技术已应用在实时控制、信息处理、通信传输、企事业管理等领域,成为人们工作、学习、生活必不可少的工具。计算机技术的发展瞬息万变,具有以下三方面特点:

(一)传统的工、理、文、医、商、农在计算机的应用方面都有着各自专业的需要,例如,经济、艺术、法律、管理、医学等各种学科都需要依赖于计算机技术的应用。除了各自领域的专业实践外,应用计算机已是各个专业提高效率、发挥潜能、促进发展的必不可少的手段。因此现在很难用传统的工、理、文、医、商、农等去界定学科的分类。

(二)计算机网络改变了计算机通信的时空距离。计算机应用的发展是与计算机网络的发展紧密相连的。从最初的局域网(LAN)到广域网(WAN),以至用一种新的方法将LAN和WAN互联起来,即成为网际网(Internetwork)。这种网际网的实验原型Internetwork,通常缩写为Internet。计算机网络将计算机互连起来,从而使计算机之间可以交换信息,而且这种信息交换可以在几分钟内就影响到世界各地。计算机网络的发展,带动了计算机学科在很多领域的拓展。

(三)现代计算机学科向综合性发展。计算技术发展伊始,每种学科均以软硬件分类,泾渭分明。但自网络发展以来,Internet软件中的两部分变得特别重要和特别具有开创性,即网际协议(Internet protocol, IP)和传输控制协议(transmission control protocol, TCP)。这些协议是必不可少的软件系统。但是在网络系统中,网络的互连必须依靠路由器、服务器、接口插座、调制解调器等硬件设施,所以计算机网络很难归结为软件或硬件的单一体系。

随着计算机技术的发展,计算机与通讯、视频、声音等密不可分;随着多媒体的发展和应用,计算机科学已经愈来愈成为与数字传输、视频、声、光、电等综合的学科。

尽管计算机技术的发展如此神速、新异,但像一切新学科的发展一样,计算机教育水平仍滞后于计算机技术的发展。为了适应计算机教学改革的需要,我们国内部分重点院校的教授、学者,在科学出版社的积极鼓励和支持下,成立了新世纪计算机专业教材编委会。自2000年10月以来,我们群策群力,多次探讨了当前教育与技术进展之间的差距,并且仔细研讨了美国ACM/IEEE-CS公布的*Computing Curricula 2001*的优点与不足,结合我国计算机教育的实际情况,提出了编著一套适用于计算机本科专业的励精图治的教材计划。这套教材的选题、定位乃至作者的遴选,都得到了国内很多著名教授和学者的认同,并且有很多选题都争取到了一些著名教授亲自参与编写。这套教材立意着重基础,反映导向,注重实践。

因此我们在基础课目方面,首先列选了数据库原理、操作系统、编译程序原理、智能基础等基础教程。这些基础课教材都由一些国内著名学者执笔,论述内容既注意打好扎实基础,又注意要反映最新导向,高屋建瓴,使读者迅速接近最新领域。

同时,为了反映导向,我们抓住网络课程作为计算机专业学生的应用基础,编写了一本实用性极强的《计算机网络教程》。这本教材的编著思想是以基础—理论—应用为主线,通信是基础,协议是核心,互连是重点,应用是目标。

其次,为了拓展学生的网络应用本领,我们还安排了电子商务、多媒体应用以及 Web 数据库技术三门应用课程。电子商务和多媒体应用是计算机应用中最为热门的课程,也是拓展性极广的计算机应用领域,应用前景极为广阔。

Web 数据库技术是一种随着互联网技术发展起来的应用技术。它涉及网络、HTTP 协议、Script 语言、动态网页开发平台、远程数据访问技术等各种网络应用技术。目前国内还无适合教材,因此,编写 Web 数据库技术的教材,可以说是填补了应用领域的一个空白。

在研究美国公布的“计算 2001-CS 教程”中,我们仔细探讨了数据结构这一课程的变化。在“计算 1991 教程”中,数据结构内容明确放在算法与数据结构之中,而“2001-CS 教程”却无数据结构的课程名称,代之以程序设计基础(programming fundamentals)。文件中提到了基本数据结构和抽象数据类型以及面向对象的程序设计等内容。从这里可以看出,数据结构是以程序设计基础作为研究对象的。另外该教程把算法与复杂性作为一个单独课程列出,这一方面说明算法是一种问题求解的策略,另一方面也说明基本算法及复杂性的讨论对于程序设计是多么重要。

为此在这套丛书中我们安排了一个软件课程系列,即开设从语言、数据结构、算法到软件工程的课程。首先我们从面向对象的 C++ 语言入手,进一步讲解语言学概论。主要内容是分析语法结构,掌握语言构成规律,读懂语言文本。任何计算机语言均可触类旁通,这种从结构规律来学会应用的方法,就是以不变应万变,因为从根本上说,尽管计算机语言千变万化,但万变不离其宗。在搞通语言基础上,我们组编了数据结构,或者说是研究程序设计基础。然后是学习基本算法,也就是为了程序设计需要,而进行问题求解,即进行常用算法讨论。为了使开发软件遵循工程管理方法,软件工程的学习将是计算机专业学生规范软件开发的必不可少的训练课程。

我们筹组这套丛书时,希望每本教材都有创意,能引起共鸣,能被关注,能被采纳,能被推广。但是我们也注意到,由于各个学校情况不同,各人观点不同,理解角度也有所不同,所以对教材的选用和编著,不易一致认同。不过我们希望这套教材能够反映当前学校动向,在促进学以致用等方面有所促进、有所推动,更希望兄弟院校的教师、学者能够积极使用,参与讨论,以使本套丛书能够不断修改,日臻完善。

最后我要感谢科学出版社的领导对本套丛书的列选、报审、出版所给予的鼓励和支持。

左孝凌

2001 年 7 月 30 日

前 言

数据结构课程的教学目的是使学生学会分析研究计算机所要加工处理的数据的特征,掌握组织数据、存储数据和处理数据的基本方法,并加强在实际应用中选择合适的数据结构和相应算法的训练。

面向对象技术是软件工程领域中的重要技术,它不仅是一种程序设计方法,更重要的是一种对真实世界的抽象思维方式。目前,面向对象的软件分析和设计技术已发展成为软件开发的主流方法。用面向对象的方式来描述数据结构及其算法成为一种趋势。

数据结构是一门知识性和实践性很强的课程,它内容丰富,学习量大;隐藏在各部分内容中的方法和技术多,贯穿于全书的动态链表存储结构和递归技术令不少初学者望而生畏。要学好这门课程必须付出极大的努力。除了上课听讲,看书理解之外,还有两个环节不可忽视:一是做书面练习,二是上机实习。只有做大量的习题和上机实习才能掌握数据结构的知识,提高算法设计的能力。

本书是科学出版社出版的“新世纪计算机专业系列教材”中的《数据结构——C++实现》一书的配套教学参考书,旨在指导、启发和帮助学生学好数据结构这门课程。

本书给出了主教材中的所有习题的参考答案和分析。对于学习数据结构这门课程的学生来讲,这些解答只可作为参考,切不可完全依赖于它。如果未做习题之前就先看答案,那就与作者编写本书的初衷背道而驰了。有关数据结构程序的习题可以有多种解答,本书提供的并不一定是唯一的,有的也不一定是最好的。学生可以设计多个程序,并加以比较。

为了加强实习环节,本书为主教材的每一章设计了一套上机实习题,并提供了可在计算机上运行的上机实习的实例。

此外,我们还对数据结构考试的题型做了介绍,并给出了3套模拟试卷。

在本书的写作过程中,上海大学教务处和计算机学院给予了很大支持,研究生岳香芬和费立志做了部分习题的录入和部分程序的调试工作,在此表示感谢。

本书是作者在多年教学的基础上,收集了各种比较典型的习题而编写的。由于时间仓促和作者水平有限,本书一定存在着疏漏和缺点,敬请广大读者批评指正。

缪淮扣

2005年3月于上海

目 录

1 绪论	1
1.1 复习提要	1
1.2 习题解析	1
1.3 上机实习题	7
2 抽象数据类型和 C++ 类	9
2.1 复习提要	9
2.2 习题解析	9
2.3 上机实习题	22
3 线性表	25
3.1 复习提要	25
3.2 习题解析	25
3.3 上机实习题	44
4 栈、队列和递归	46
4.1 复习提要	46
4.2 习题解析	46
4.3 上机实习题	61
5 串、数组和广义表	64
5.1 复习提要	64
5.2 习题解析	65
5.3 上机实习题	88
6 树和森林	94
6.1 复习提要	94
6.2 习题解析	95
6.3 上机实习题	113
7 图	119
7.1 复习提要	119
7.2 习题解析	120
7.3 上机实习题	148
8 查找	151
8.1 复习提要	151
8.2 习题解析	152
8.3 上机实习题	172

9 排序	175
9.1 复习提要	175
9.2 习题解析	176
9.3 上机实习题	194
10 考试指导	196
10.1 题型介绍.....	196
10.2 模拟试卷 1	203
10.3 模拟试卷 2	212
10.4 模拟试卷 3	221
11 综合设计实验与实验示例	229
11.1 综合设计实验.....	229
11.2 实验示例——商品货架管理.....	231
参考文献	239

1

绪 论

1.1 复习提要

“数据结构”是计算机科学与技术专业本科生的一门专业基础课,根据教学大纲的要求,主要讨论在软件开发中如何进行数据的组织、数据的表示和数据的处理,它不仅为操作系统、编译原理、数据库系统、计算机网络等后续课程提供必要的知识基础,而且也为学生者提供必要的技能训练,所以它是一门知识性、实践性都很强的课程。不少单位在招聘计算机人才时,许多高校和科研机构在招收计算机专业研究生时往往将“数据结构”作为考核的课程之一,足见其重要性。

本章主要介绍了贯穿和应用整个“数据结构”课程始终的基本概念和算法性能分析方法。

传统的“数据结构”概念从数据的逻辑结构(数据的组织)、物理结构——存储结构(数据的表示)和相关操作(数据的处理)等三方面进行讨论,反映了数据结构设计的不同层次。按照面向对象建模技术的要求,在软件开发中做数据结构设计时不但要设计对象——类、类的属性、类的操作,还要建立类的实例,即对象之间的关系。所以,把数据结构定义为数据对象及对象中数据成员(元素)之间关系的集合是合适的。

操作—算法的设计属于面向过程的开发模式,即传统的“输入—计算—输出”模式。为比较解同一问题的不同算法的优劣,进而设计出好的算法,对算法的性能分析是必要的。

本章的基本要求是:

(1) 需要理解什么是数据、数据对象、数据元素(数据成员)、数据结构、数据的逻辑结构与物理结构(存储结构)。

(2) 需要理解算法的定义、算法的特性、算法的时间复杂度和空间复杂度。

数据结构设计及算法的编写与分析都是不容易的,往往讲课时容易接受,但课下自学难度大,特别是做练习题时常常无从下手。需要付出极大的努力,勤学多练。多总结、多思考、多做题、自主做题、勤上机、勤交流,这是学好“数据结构”这门课程的关键。

1.2 习题解析

1. 简述下列术语:数据、数据元素、数据对象和数据类型。

【解答】

数据是信息的载体,是描述客观事物的数、字符、图形、图像、声音以及所有能输入到计算机中并被计算机程序识别和处理的符号的集合。

例如,一个解代数方程的程序处理的对象只是整数和实数,一个程序设计语言的编译

程序处理的对象是由 ASCII 码字符组成的字符串等。

数据元素是数据的基本单位。一个数据元素可由若干个数据项组成,数据项是数据的最小单位。

例如,在一个关于学生成绩的登记表中,每个学生的姓名、班级、各课程成绩组成了一个数据元素,其中的姓名是一个数据项,班级是一个数据项,每一门课程都是一个数据项。

数据对象是具有相同特性的数据元素所构成的集合。数据对象是数据的子集。例如,所有的“数”构成了“数据”集合,而正整数集合 $N = \{1, 2, \dots\}$ 是“数”的数据对象;所有的字符是数据,字母集合 $AS = \{A, B, \dots, Z, a, b, \dots, z\}$ 是该数据的数据对象。

数据类型是一个和数据结构密切相关的概念,它在高级程序设计语言中用以刻画程序的操作对象的特性。数据类型是一个值的集合和定义在这个集上的一组操作的总称。可以看成是数据结构的实现。

2. 什么是数据结构? 什么是数据的逻辑结构? 什么是数据的物理结构?

【解答】

数据结构是数据元素的集合以及该集合中的数据元素之间关系的集合。根据数据元素之间关系的不同,数据结构分为两大类:线性结构和非线性结构。

数据的逻辑结构描述的是数据元素之间的逻辑关系,它属于用户视图,是用户所看到的数据结构,是面向问题的,它不考虑数据的存储。

数据的物理结构,又称存储结构,是数据的逻辑结构在计算机中的物理存储方式,它属于具体实现的视图,是面向计算机的。

数据的逻辑结构和物理结构是密切相关的两个方面。一般来说,算法设计是基于数据的逻辑结构,而算法实现则基于数据的物理结构。

3. 对下列测试题填空。

(1) _____ 是描述客观事物的数、字符以及所有能输入到计算机且被计算机程序加工处理的符号集合。

(2) _____ 是数据的基本单位,有时一个 _____ 由若干个 _____ 组成,在这种情况下,称 _____ 为记录, _____ 是数据的最小单位,而由记录组成的线性表称为 _____。

(3) 被计算机加工的 _____ 不是孤立无关的,它们彼此之间一般存在着某种联系,通常将 _____ 之间的这种联系称为 _____。

(4) 在 C++ 中建立参数类型和个数不同的同名函数是可能的,这称为函数 _____。

(5) 运算符 _____ 动态分配一个对象。

(6) 假定 a 和 b 是整数变量,可构造一个表达式 $a+b$,再假定 c 和 d 是浮点数变量,也可构造一个表达式 $c+d$ 。这两个表达式中的 + 运算符显然是用于不同的类型。C++ 中的这种性质称为 _____。

(7) 本章给出的 C++ 中预定义的两类流对象是 _____ 和 _____。

(8) _____ 限定符用来声明只读变量。

【解答】

(1) 数据

- (2) 数据元素 数据元素 数据项
 数据元素 数据项 文件
- (3) 数据元素 数据元素 结构
- (4) 重载
- (5) new
- (6) 运算符重载
- (7) cin cout
- (8) const

4. 在对多行文本使用块风格的注释时,使用 C++ 的哪一种注释界定符更简洁?

【解答】

多行注释的方法是采用定界符“/*”和“*/”,包含在定界符“/*”和“*/”之间的所有文本内容都是注释。也可以采用单行注释的方法对多行文本进行注释,单行注释定界符为“//”,在符号“//”之后至本行末的所有文本内容均为注释。

应该说,在对多行文本使用块风格的注释时,采用定界符“/*”和“*/”更简洁。

5. 为什么说 C++ 中的运算符“<<”和“>>”是重载的运算符?

【解答】

因为运算符“>>”既是右移位运算符,又是流提取运算符(根据上下文确定运算符的性质);运算符“<<”既是左移位运算符,又是流插入运算符(根据上下文确定运算符的性质)。

所以说 C++ 中的运算符“<<”和“>>”是重载的运算符。

6. 编写如下要求的 C++ 程序段。

- (1) 把字符串“enter your string”写入标准输出流 cout。
- (2) 从标准输入流 cin 读取变量 salary。

【解答】

关键字 cin 用于 C++ 中的输入,操作符“>>”用于分开输入的变量。空白(即 Tab 键、回车或空格键)用于在标准输入设备上将不同变量的值分开。

关键字 cout 用于输出到一个标准输出设备。cout 和将被输出的每一内容之间用操作符“<<”分开。要输出的内容从左向右被输出到标准输出设备上。

本题要输入和输出的变量都只有一个,比较简单。

- (1) cout << "enter your string";
- (2) cin >> salary

7. 编写一个完整的 C++ 函数,用输入/输出流和内联函数 CircleArea 提示用户输入圆的半径,然后计算并打印出圆的周长和面积。

【解答】

```
# include <iostream.h>
```

```
const float PI=3.14159;    //定义浮点类型的常量 PI
```

```
inline float CircleArea(const float r) {  
    return PI * r * r;}  
    //声明一个内联函数,它计算半径为 r 的面积  
main ()  
{  
    float radius;    //声明一个作为半径的变量  
    cout << "Enter the length of the radius of your circle:";  
        //提示输入圆的半径的值  
    cin >> radius; //输入圆的半径的值  
    cout << "Area of circle with radius" << radius <<  
        "is" << CircleArea (radius) << '\n';  
    return 0;  
}
```

8. 编写一个完整的 C++ 程序,分别用下面的三个函数,对在 main 中定义的变量 count 乘以 2,然后比较一下这三种方法的优劣。这三个函数是:

(1) doubleCallByValue 通过传值调用传递 count 的拷贝,把 count 值的拷贝乘以 2 后返回新值。

(2) 函数 doubleByPointer 用指针模拟传引用调用,用引用运算符“*”对 main 中 count 的原始拷贝乘以 2。

(3) 函数 doubleByReference 利用引用参数实现真正的引用调用,用 count 的别名(即引用参数)将其原始值的拷贝乘以 2。

【解答】

这是一个区别传值调用、传指针调用和传引用调用的程序:

```
#include <iostream.h>  
int doubleCallByValue(int a);  
void doubleCallByPointer(int * bPtr );  
void doubleCallByReference(int & cRef);  
  
main ()  
{  
    int count = 3,count1= count;  
    cout << "count =" << count << "before doubleCallByValue\n"  
        << "Value returned by doubleCallByValue:"  
        << doubleCallByValue(count) //传值调用  
        << "\n count =" << count << "after doubleCallByValue\n\n";  
  
    count = count1;  
    cout << "count =" << count << "before doubleCallByPointer\n";  
    doubleCallByPointer (&count) ;
```

```

//传指针调用
cout << " count=" << count << "after doubleCallByValue\n\n";

count = count1;
cout << "count = " << count << "before doubleCallByReference\n";
doubleCallByReference (count) ;
//传引用调用
cout << "count=" << count << "after doubleCallByReference\n";
return 0;
}

int doubleCallByValue(int a) //传值调用的函数的实现
{
    return a * 2; // 不修改调用者的参数
}

void doubleCallByPointer(int * bPtr) //传指针调用的函数的实现
{
    * bPtr * 2; // 修改调用者的参数
}

void doubleCallByReference(int &cRef) //传引用调用的函数的实现
{
    \cRef * 2; // 修改调用者的参数
}

```

输出结果:

```

count = 3 before doubleCallByValue
Value returned by doubleCallByValue: 6
count = 3 after doubleCallByValue

count = 3 before doubleCallByPointer
count = 6 after doubleCallByPointer

count = 3 before doubleCallByReference
count = 6 after doubleCallByReference

```

传值调用 `doubleCallByValue` 不修改调用者的参数 `count`, 而传指针调用 `doubleCallByPointer` 和传引用调用 `doubleCallByReference` 均修改调用者的参数 `count`。建议用指针传递那些允许被调用函数修改的参数, 而用对常量的引用来传递那些不需修改的大型参数。

注意, 因为被调用函数的函数体内只通过名字使用引用参数, 所以程序员可能会不小心把引用参数当作传值调用参数。如果该参数修改了变量的原始拷贝, 会导致意想不到的副作用。

9. 什么是算法? 算法的 5 个特性是什么? 请解释算法与程序的区别。

【解答】

通常,算法被定义为一个用于实现某个特定任务的有穷指令集,这些指令规定了一个运算序列。算法应当具有以下特性:

(1) 输入性: 一个算法必须具有零个或多个输入量。

(2) 输出性: 一个算法应有一个或多个输出量,输出量是算法计算的结果。

(3) 确定性: 算法中的每一条指令应含义明确,无歧义,即对每一种情况、需要执行的动作都应严格地、清晰地规定。

(4) 有穷性: 算法中的指令执行序列是有穷的,即算法无论在什么情况下都应在执行有穷步后终止。

(5) 有效性: 每条指令必须是足够基本的。就是说,它们原则上都能精确到可用笔和纸来模拟实现其指令过程。对于每一步操作,仅有(3)中的确定性是不够的,它还必须是可行的。

算法的描述可以依需要有一定的抽象程度。算法可以用某种语言和表示法来描述,它并不要求一定要在计算机上执行。程序用某种程序设计语言描述,可在计算机上执行。

一个程序与一个算法对于上述(4)是有重大区别的。一个程序可以不满足特性(4)。例如,以用户名和口令来进行登录的系统就是一个处于“键入用户名、键入口令,检查是否已在系统中注册过并且匹配,如不满足,则重新输入”的循环之中,直至键入的用户名和口令匹配,这样的程序可能会不终止。

10. 设 n 为正整数,分析下列各程序段中加下划线的语句的执行次数:

(1)

```
for (int i=1; i<=n; i++)
for (int j=1; j<=n; j++) {
    c[i][j]=0.0;
    for (int k=1; k<=n; k++)
        c[i][j] = c[i][j]+a[i][k] * b[k][j];
}
```

(2)

```
x=0; y=1;
for (int i=1; i<=n; i++)
for (int j=1; j<=i; j++)
    for (int k=1; k<=j; k++)
        x = x+y;
```

(3)

```
int i=1, j=1;
while (i<=n && j<=n) {
    i = i+1; j = j+i;
}
```

【解答】

(1) 这是一个带有三重循环的程序段, 每一个循环的次数都是 n , 加横线的语句处在最里面的循环中, 所以该语句的执行次数为

$$\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n 1 = n^3$$

(2) 这是一个带有三重循环的程序段, i 循环的计数是 $1 \sim n$, j 循环的计数是 $1 \sim i$, k 循环的计数是 $1 \sim j$, 加横线的语句处在最里面的循环中, 所以该语句的执行次数为

$$\begin{aligned} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^j 1 &= \sum_{i=1}^n \sum_{j=1}^i j = \sum_{i=1}^n \left(\frac{i(i+1)}{2} \right) = \frac{1}{2} \sum_{i=1}^n i^2 + \frac{1}{2} \sum_{i=1}^n i \\ &= \frac{1}{2} \frac{n(n+1)(2n+1)}{6} + \frac{1}{2} \frac{n(n+1)}{2} = \frac{n(n+1)(n+2)}{6} \end{aligned}$$

(3) i 为 1 时, $i=2, j=j+i=1+2=2+1$

i 为 2 时, $i=3, j=j+i=(2+1)+3=3+1+2$

i 为 3 时, $i=4, j=j+i=(3+1+2)+4=4+1+2+3$

i 为 4 时, $i=5, j=j+i=(4+1+2+3)+5=5+1+2+3+4$

.....

i 为 k 时, $i=k+1, j=j+i=(k+1)+(1+2+3+4+\dots+k)$

因为 $j=(k+1) + \sum_{i=1}^k i \leq n$, 所以

$$(k+1) + \frac{k(k+1)}{2} = \frac{k^2 + 3k + 2}{2} \leq n$$

解出满足上述不等式的 k 的最大值, 即为语句 $i = i + 1$ 的程序步数。

1.3 上机实习题



实验目的

熟练掌握 C++ 语言的基本结构及相关语句。



实验内容

(一) 输入输出语句

[问题描述]

分别输入圆柱体的半径和高, 求其体积; 输入球半径, 求其表面积; 输入长方体的长、宽、高, 求其体积。

[基本要求]

依次输入相应的数据进行计算, 然后输出计算结果。

[测试数据]

输入圆柱体的半径和高: 32 44

输出圆柱体的体积: 141548

输入球半径:30

输出球表面积:11309.7

输入长方体的长、宽、高:32 12 15

输出长方体体积:5760

学生还可以设定其他的测试数据。

(二) 赋值语句、条件语句和循环语句

[问题描述]

编一个程序,对给定的两个整数变量 a 和 b,求其最大公约数,并输出之。

[基本要求]

在程序中提示输入两个整数,输出计算结果。

[测试数据]

由学生自己确定。

(三) 函数调用

[问题描述]

某工厂生产汽车,1月份生产两辆,2月份是1月份产量减去半辆,再翻一番;3月份是2月份产量减去半辆,再翻一番。如此下去。求从2月份开始,每月的汽车产量。

[基本要求]

先编写一个计算给定月份的汽车产量的函数,然后在主函数中用一个循环语句来调用该函数。在程序中输出从2月份开始,每月的汽车产量。