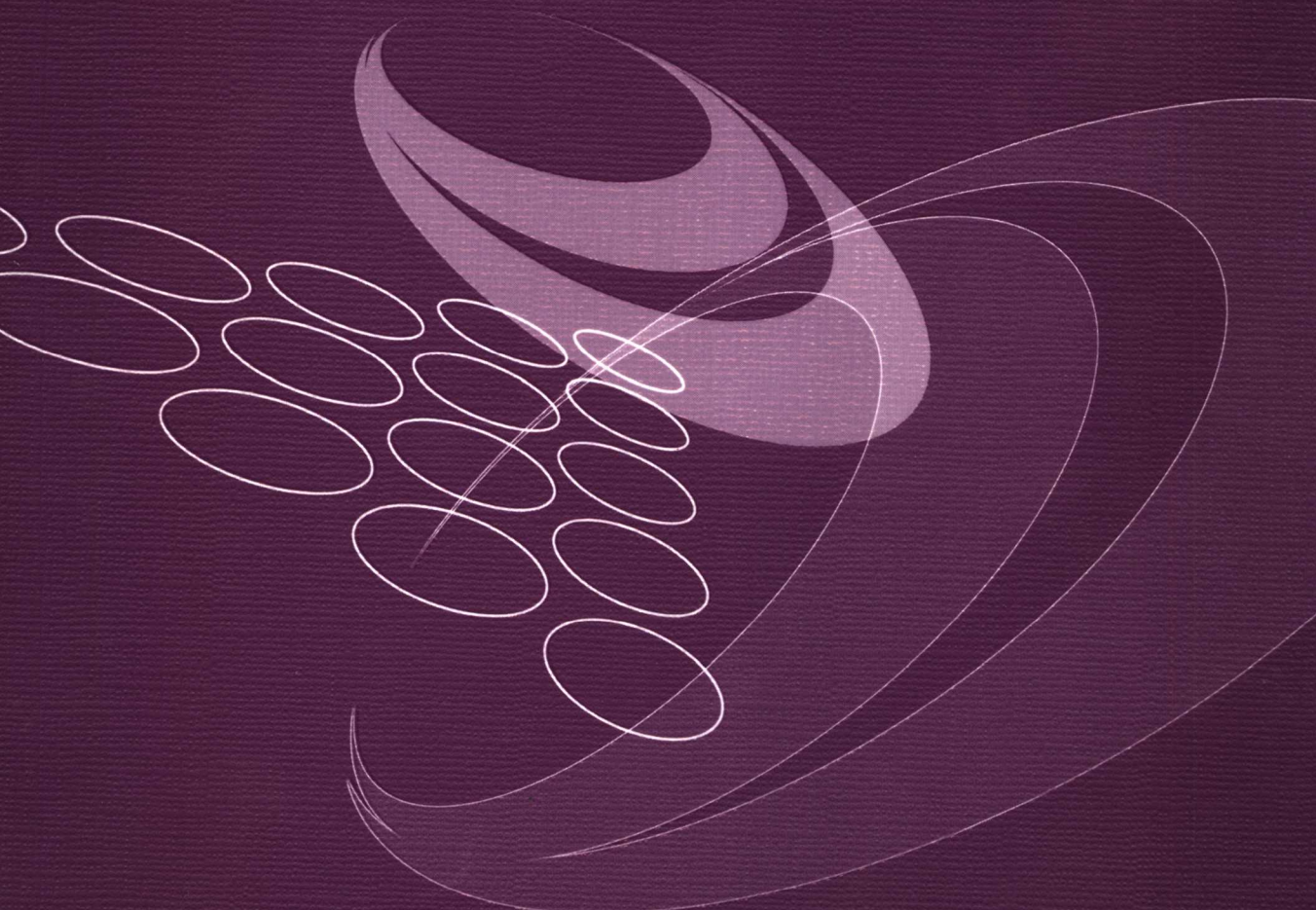


普通高等院校
计算机专业(本科)实用教程系列

数据结构实用教程

(第二版) 习题参考解答

徐孝凯 编著



清华大学出版社

普通高等院校计算机专业(本科)实用教程系列

数据结构实用教程(第二版)

习题参考解答

徐孝凯 编著

清华大学出版社
北京

内 容 简 介

本书是徐孝凯编著的《数据结构实用教程（第二版）》的配套辅助教材，与主教材同时修订。本书提供了主教材中绝大部分习题的参考解答，并对较难的习题进行了详细分析。在每章的习题中，还给出了选择题和填空题，通过这些练习能够加深对基本概念的理解和认识。

本书含有丰富的题型和例题，许多算法都包含在可以直接运行的程序中。本书可以作为学习数据结构课程的辅助教材或考研复习的参考书。

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

数据结构实用教程(第二版)习题与参考解答/徐孝凯编著. —北京:清华大学出版社,2006.9

(普通高等院校计算机专业(本科)实用教程系列)

ISBN 7-302-13398-0

I. 数… II. 徐… III. 数据结构—高等学校—解题 IV. TP311.12-44

中国版本图书馆 CIP 数据核字(2006)第 078064 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: 郑寅堃

文稿编辑: 赵晓宁

印 刷 者: 北京市清华园胶印厂

装 订 者: 三河市新茂装订有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 9.75 字数: 236 千字

版 次: 2006 年 9 月第 1 版 2006 年 9 月第 1 次印刷

书 号: ISBN 7-302-13398-0/TP·8411

印 数: 1~5000

定 价: 13.00 元

读者意见反馈

亲爱的读者：

感谢您一直以来对清华版计算机教材的支持和爱护。为了今后为您提供更优秀的教材，请您抽出宝贵的时间来填写下面的意见反馈表，以便我们更好地对本教材做进一步改进。同时如果您在使用本教材的过程中遇到了什么问题，或者有什么好的建议，也请您来信告诉我们。

地址：北京市海淀区双清路学研大厦 A 座 602 室 计算机与信息分社营销室 收
邮编：100084 电子邮箱：jsjjc@tup.tsinghua.edu.cn
电话：010-62770175-4608/4409 邮购电话：010-62786544

教材名称：数据结构实用教程（第二版）习题参考解答

ISBN：7-302-13398-0/TP·8411

个人资料

姓名：_____ 年龄：_____ 所在院校/专业：_____

文化程度：_____ 通信地址：_____

联系电话：_____ 电子信箱：_____

您使用本书是作为：指定教材 选用教材 辅导教材 自学教材

您对本书封面设计的满意度：

很满意 满意 一般 不满意 改进建议_____

您对本书印刷质量的满意度：

很满意 满意 一般 不满意 改进建议_____

您对本书的总体满意度：

从语言质量角度看 很满意 满意 一般 不满意

从科技含量角度看 很满意 满意 一般 不满意

本书最令您满意的是：

指导明确 内容充实 讲解详尽 实例丰富

您认为本书在哪些地方应进行修改？（可附页）

您希望本书在哪些方面进行改进？（可附页）

电子教案支持

敬爱的教师：

为了配合本课程的教学需要，本教材配有配套的电子教案（素材），有需求的教师可以与我们联系，我们将向使用本教材进行教学的教师免费赠送电子教案（素材），希望有助于教学活动的开展。相关信息请拨打电话 010-62776969 或发送电子邮件至 jsjjc@tup.tsinghua.edu.cn 咨询，也可以到清华大学出版社主页（<http://www.tup.com.cn> 或 <http://www.tup.tsinghua.edu.cn>）上查询。

普通高等院校计算机专业(本科)实用教程系列

主教材

- 信息技术基础实用教程(樊孝忠 等编著)
- 数字逻辑实用教程(王玉龙 编著)
- 计算机组成原理实用教程(第二版)(幸云辉 等编著)
- C++语言基础教程(徐孝凯 编著)
- 数据结构实用教程(第二版)(徐孝凯 编著)
- 面向对象程序设计实用教程(张海藩 等编著)
- 操作系统实用教程(第二版)(任爱华 等编著)
- 数据库实用教程(第二版)(丁宝康 等编著)
- 计算机网络实用教程(第二版)(刘云 等编著)
- 微机接口技术实用教程(艾德才 等编著)
- JAVA 2 实用教程(第二版)(耿祥义 等编著)
- 离散数学结构(王家廐 编著)
- 微型计算机技术实用教程(Pentium 版)(艾德才 等编著)
- 编译原理实用教程(温敬和 等编著)
- JAVA 2 实用教程(修订版)(耿祥义 等编著)
- Java 语言最新实用案例教程(杨树林 等编著)
- 信息技术英语阅读(王栋 等编著)

辅助教材

- 数据结构课程实验(徐孝凯 编著)
- 数据结构实用教程(第二版)习题参考解答(徐孝凯 编著)
- 数据库实用教程(第二版)习题解答(丁宝康 等编著)
- 面向对象程序设计实用教程习题解答与应用实例(配光盘)(牟永敏 等编著)
- 操作系统实验指导(任爱华 等编著)
- 离散数学结构习题与解答(王家廐 编)
- JAVA 2 实用教程(第二版)实验指导与习题解答(耿祥义 等编著)
- Java 课程设计(耿祥义 等编著)

选修教材

- JSP 实用教程(耿祥义 等编著)

普通高等院校计算机专业（本科）实用教程系列 编委会

主任 孙家广（清华大学教授，中国工程院院士）
成员（按姓氏笔画为序）

王玉龙（北方工业大学教授）

艾德才（天津大学教授）

刘云（北方交通大学教授）

任爱华（北京航空航天大学教授）

杨旭东（北京邮电大学副教授）

张海藩（北京信息工程学院教授）

徐孝凯（中央广播电视大学教授）

徐培忠（清华大学出版社编审）

樊孝忠（北京理工大学教授）

丛书策划 徐培忠 徐孝凯

序 言

时光更迭、历史嬗递。中国经济以她足以令世人惊叹的持续高速发展驶入了一个新的世纪，一个新的千年。世纪之初，以微电子、计算机、软件和通信技术为主导的信息技术革命给我们生存的社会所带来的变化令人目不暇接。软件是优化我国产业结构、加速传统产业改造和用信息化带动工业化的基础产业，是体现国家竞争力的战略性产业，是从事知识的提炼、总结、深化和应用的高智型产业；软件关系到国家的安全，是保证我国政治独立、文化不受侵蚀的重要因素；软件也是促进其他学科发展和提升的基础学科；软件作为 20 世纪人类文明进步的最伟大成果之一，代表了先进文化的前进方向。美国政府早在 1992 年“国家关键技术”一文中提出“美国在软件开发和应用上所处的传统领先地位是信息技术及其他重要领域竞争能力的一个关键因素”，“一个成熟的软件制造工业的发展是满足商业与国防对复杂程序日益增长的要求所必需的”，“在很多国家关键技术中，软件是关键的、起推动作用（或阻碍作用）的因素”。在 1999 年 1 月美国总统信息技术顾问委员会的报告“21 世纪的信息技术”中指出“从台式计算机、电话系统到股市，我们的经济与社会越来越依赖于软件”，“软件研究为基础研究方面最优先发展的领域。”而软件人才的缺乏和激烈竞争是当前国际的共性问题。各国、各企业都对培养、引进软件人才采取了特殊政策与措施。

为了满足社会对软件人才的需要，为了让更多的人可以更快地学到实用的软件理论、技术与方法，我们编著了《普通高等院校计算机专业（本科）实用教程系列》。本套丛书面向普通高等院校学生，以培养面向 21 世纪计算机专业应用人才（以软件工程师为主）为目标，以简明实用、便于自学、反映计算机技术最新发展和应用为特色，具体归纳为以下几点：

1. 进透基本理论、基本原理、方法和技术，在写法上力求叙述详细，算法具体，通俗易懂，便于自学。

2. 理论结合实际。计算机是一门实践性很强的科学，丛书贯彻从实践中来到实践中去的原则，许多技术理论结合实例讲解，以便于学习理解。

3. 本丛书形成完整的体系，每本教材既有相对独立性，又有相互衔接和呼应，为总的培养目标服务。

4. 每本教材都配以习题和实验，在各教学阶段安排课程设计或大作业，培养学生的实战能力与创新精神。习题和实验可以制作成光盘。

为了适应计算机科学技术的发展，本系列教材将本着与时俱进的精神不断修订更新，及时推出第二版、第三版……

新世纪曙光激人向上，催人奋进。江泽民同志在十五届五中全会上的讲话：“大力推进国民经济和社会信息化，是覆盖现代化建设全局的战略举措。以信息化带动工业化，发挥后发优势，实现社会生产力的跨越式发展”，指明了我国信息界前进的方向。21世纪日趋开放的国策与更加迅速发展的科技会托起祖国更加辉煌灿烂的明天。

孙家广

2004年1月

前 言

本书是与清华大学出版社出版的《数据结构实用教程（第二版）》一书相配套的辅助教材，它给出了主教材中每一章大部分习题的参考解答。

数据结构是一门实践性很强的课程。对于自己编写的每一个算法，不仅要从设计思路分析其正确性和有效性，更重要的是上机验证，并且在反复调试的过程中，通过典型的数据输入使得算法中的每条语句都被执行过，或者说不存在没有被执行过的语句或语句块。若调试过程发现语法或逻辑错误，则要及时修改。所谓逻辑错误是指算法设计上隐含的错误，虽然算法能够被正确地编辑和连接，但运行后得不到正确的结果。通过上机运行程序不仅能验证算法的正确性，而且能加深对所学知识的理解和掌握，进而获得书本上学不到的知识。

解决一个数据处理问题通常要经过以下几步：①根据题目要求分析出设计思路；②根据设计思路画出相应的流程图；③根据流程图用一种计算机语言（如 C++）编写出详细算法；④编写出能够调用该算法的完整程序；⑤上机调试和运行该程序。若发现错误则回到上述某一步开始向下修改，通过反复调试和修改，直到获得满意的结果为止。当然对于一些简单问题，上述步骤有的可以省略，有的则可以合并。

对于要解决的同一个问题，由于所采用的数据结构可能不同，所选择的运算思路和方法可能不同，则编写出的算法描述就可能不同，但只要你的算法正确并且有效（即具有较好的时间和空间复杂度）即可。例如要对 n 个数据进行排序，既可选择数组结构，也可选择二叉排序树结构；对于数组结构，可以从快速、堆、合并、直接插入、直接选择等多种排序运算方法中任选一种。当采用不同的数据结构和排序方法时编写出的排序算法也不会相同，但它们都是正确的，通过进行时间和空间复杂度的分析可以比较出哪一种或几种更为有效。因此，不要求读者编写出的算法和程序与本书所给的解答完全一致，也许你的算法更优秀，综合指标更好。同学们绝不要依赖现成的答案，而要举一反三，通过独立思考或同学之间的相互讨论完成作业。希望同学们要按照每个题目的要求，独立地分析问题和解决问题，编写出相应的算法，不要事先翻阅答案。书中所给的参考解答只有当你解题后或确实不会做时进行参考才是有益的。

书中带星号“*”的习题都是较难的，有的已超出教学要求，同学们可以选做或不做。对于不带星号的习题，也可以根据所在专业和班级的具体教学要求而选做。

进行算法分析和设计，必须有 C++ 上机运行的环境，只有上机调试通过才能证明算法是正确的，否则很难发现和解决所存在的算法设计和编程问题。

书中所有算法和程序都在 Visual C++ 6.0 版本上调试通过，但由于编写时间仓促，错误和不足之处在所难免，敬请广大读者批评指正。

欢迎读者就本书发表批评和建议,本人联系方式为:

电子邮件地址: xuxk@crtvu.edu.cn 固定电话: 010-64910302

徐孝凯

2006年8月

目 录

第 1 章 绪论	1
第 2 章 线性表	20
第 3 章 稀疏矩阵和广义表	39
第 4 章 栈和队列	49
第 5 章 树和二叉树	73
第 6 章 特殊二叉树	85
第 7 章 图	95
第 8 章 图的应用	105
第 9 章 查找	116
第 10 章 排序	133

第1章 绪 论

【习题 1-1】 根据二元组表示分析其数据结构。

有下列几种用二元组表示的数据结构，试画出它们分别对应的图形表示（当出现多个关系时，对每个关系画出相应的结构图），并指出它们分别属于何种结构。

1. $A=(K,R)$ ，其中，

$$K=\{a_1, a_2, a_3, \dots, a_n\}$$

$$R=\{\}$$

2. $B=(K,R)$ ，其中，

$$K=\{a, b, c, d, e, f, g, h\}$$

$$R=\{r\}$$

$$r=\{\langle a, b \rangle, \langle b, c \rangle, \langle c, d \rangle, \langle d, e \rangle, \langle e, f \rangle, \langle f, g \rangle, \langle g, h \rangle\}$$

3. $C=(K,R)$ ，其中，

$$K=\{a, b, c, d, e, f, g, h\}$$

$$R=\{\langle d, b \rangle, \langle d, g \rangle, \langle b, a \rangle, \langle b, c \rangle, \langle g, e \rangle, \langle g, h \rangle, \langle e, f \rangle\}$$

4. $D=(K,R)$ ，其中，

$$K=\{1, 2, 3, 4, 5, 6\}$$

$$R=\{(1, 2), (2, 3), (2, 4), (3, 4), (3, 5), (3, 6), (4, 5), (4, 6)\}$$

5. $E=(K,R)$ ，其中，

$$K=\{48, 25, 64, 57, 82, 36, 75, 43\}$$

$$R=\{r_1, r_2, r_3\}$$

$$r_1=\{\langle 48, 25 \rangle, \langle 25, 64 \rangle, \langle 64, 57 \rangle, \langle 57, 82 \rangle, \langle 82, 36 \rangle, \langle 36, 75 \rangle, \langle 75, 43 \rangle\}$$

$$r_2=\{\langle 48, 25 \rangle, \langle 48, 64 \rangle, \langle 64, 57 \rangle, \langle 64, 82 \rangle, \langle 25, 36 \rangle, \langle 82, 75 \rangle, \langle 36, 43 \rangle\}$$

$$r_3=\{\langle 25, 36 \rangle, \langle 36, 43 \rangle, \langle 43, 48 \rangle, \langle 48, 57 \rangle, \langle 57, 64 \rangle, \langle 64, 75 \rangle, \langle 75, 82 \rangle\}$$

参考解答：略。

【习题 1-2】 按要求设计抽象数据类型。

设计二次多项式 ax^2+bx+c 的一种抽象数据类型，假定该抽象数据类型命名为 QUAdratic，该类型的数据部分为 3 个系数项 a 、 b 和 c ，操作部分为：

1. 初始化数据成员 a 、 b 和 c （假定用记录类型 Quadratic 定义数据成员），每个数据成员的默认值为 0。

```
void InitQuadratic(Quadratic& q, float aa=0, float bb=0, float cc=0);
```

2. 做两个多项式加法，即使对应的系数相加，返回相加结果。

```
Quadratic Add(Quadratic& q1, Quadratic& q2);
```

3. 根据给定 x 的值, 计算多项式的值并返回。

```
float Eval(Quadratic& q, float x);
```

4. 计算方程 $ax^2+bx+c=0$ 的两个实数根并引用返回, 对于有实根、无实根和不是二次方程 (即 $a=0$) 这 3 种情况都要返回不同的整数值, 以便调用函数做不同的处理。

```
int Root(Quadratic& q, float& r1, float& r2);
```

5. 按照 $ax^{**2}+bx+c$ 的格式 (x^2 用 x^{**2} 表示) 输出二次多项式, 在输出时要注意去掉系数为 0 的项, 并且当 b 和 c 的值为负时, 其前不能出现加号。

```
void Print(Quadratic& q);
```

请写出上面每一个操作的具体实现。作为选择, 有兴趣的学生还可以给出该抽象数据类型所对应的 C++ 类的描述。

参考答案包含在下面的程序之中。

```
#include<iostream.h>
#include<math.h>
struct Quadratic {
    float a; float b; float c;
};

void InitQuadratic(Quadratic& q, float aa, float bb, float cc)
{
    q.a=aa;
    q.b=bb;
    q.c=cc;
}

Quadratic Add(Quadratic& q1, Quadratic& q2)
{
    Quadratic q;
    q.a=q1.a+q2.a;
    q.b=q1.b+q2.b;
    q.c=q1.c+q2.c;
    return q;
}

float Eval(Quadratic& q, float x)
{
    return (q.a*x*x+q.b*x+q.c);
}

int Root(Quadratic& q, float& r1, float& r2)
{
```

```
    if(q.a==0) return -1;
    float x=q.b*q.b-4*q.a*q.c;
    if(x>=0){
        r1=(float)(-q.b+sqrt(x))/(2*q.a);
        r2=(float)(-q.b-sqrt(x))/(2*q.a);
        return 1;
    }
    else return 0;
}

void Print(Quadratic& q)
{
    if(q.a) {
        if(q.a==1.0) cout<<'+';
        else if(q.a==-1.0) cout<<'-' ;
        else cout<<q.a;
        cout<<"x**2";
    }
    if(q.b)
        if(q.b>0) {
            cout<<"+";
            if(q.b!=1.0) cout<<q.b;
            cout<<"x";
        }
        else {
            if(q.b==-1.0) cout<<'-' ;
            else cout<<q.b;
            cout<<"x";
        }
    if(q.c)
        if(q.c>0) cout<<"+"<<q.c;
        else cout<<q.c;
    cout<<endl;
}

void main()
{
    Quadratic q1,q2;
    float a,b,c,d1,d2;
    cout<<"输入一个二次多项式三个系数项的值: ";
    cin>>a>>b>>c;
    InitQuadratic(q1,a,b,c);
    cout<<"输入另一个二次多项式三个系数项的值: ";
    cin>>a>>b>>c;
    InitQuadratic(q2,a,b,c);
```

```

    q2=Add(q1,q2);
    cout<<Eval(q1,2)<<' '<<Eval(q2,3)<<endl;
    int f=Root(q1,d1,d2);
    if(f==1) cout<<d1<<' '<<d2<<endl;
    Print(q1);
    Print(q2);
}

```

若采用 C++ 语言中的“类”描述,则参考程序如下。

```

#include<iostream.h>
#include<math.h>
class Quadratic {
    float a; float b; float c;
public:
    Quadratic(float aa=0, float bb=0, float cc=0);
    Quadratic Add(Quadratic& q2);
    float Eval(float x);
    int Root(float& r1, float& r2);
    void Print();
};

Quadratic::Quadratic(float aa, float bb, float cc)
{
    a=aa; b=bb; c=cc;
}

Quadratic Quadratic::Add(Quadratic& q2)
{
    Quadratic q;
    q.a=a+q2.a;
    q.b=b+q2.b;
    q.c=c+q2.c;
    return q;
}

float Quadratic::Eval(float x)
{
    return (a*x*x+b*x+c);
}

int Quadratic::Root(float& r1, float& r2)
{
    if(a==0) return -1;
    float x=b*b-4*a*c;
}

```

```
    if(x>=0){
        r1=(float)(-b+sqrt(x))/(2*a);
        r2=(float)(-b-sqrt(x))/(2*a);
        return 1;
    }
    else return 0;
}

void Quadratic::Print()
{
    if(a) {
        if(a==1.0) cout<<'+';
        else if(a==-1.0) cout<<'-' ;
        else cout<<a;
        cout<<"x**2";
    }
    if(b)
        if(b>0) {
            cout<<"+";
            if(b!=1.0) cout<<b;
            cout<<"x";
        }
        else {
            if(b==-1.0) cout<<'-' ;
            else cout<<b;
            cout<<"x";
        }
    if(c)
        if(c>0) cout<<"+"<<c;
        else cout<<c;
    cout<<endl;
}

void main()
{
    float a,b,c,d1,d2;
    cout<<"输入一个二次多项式三个系数项的值: ";
    cin>>a>>b>>c;
    Quadratic q1(a,b,c);
    cout<<"输入另一个二次多项式三个系数项的值: ";
    cin>>a>>b>>c;
    Quadratic q2(a,b,c);
    q2=q1.Add(q2);
    cout<<q1.Eval(2)<<' '<<q2.Eval(3)<<endl;
    int f=q1.Root(d1,d2);
}
```



```

    if(f==1) cout<<d1<<' '<<d2<<endl;
    q1.Print();
    q2.Print();
}

```

【习题 1-3】 用 C++ 函数描述算法并求出其时间复杂度。

1. 比较同一简单类型的两个数据 x_1 和 x_2 的大小, 对于 $x_1 > x_2$, $x_1 == x_2$ 和 $x_1 < x_2$ 这 3 种不同情况应分别返回 “>”, “=” 和 “<” 字符。假定简单类型用 SimpleType 表示, 它可通过 typedef 语句定义为任一简单类型。

2. 将一个字符串中的所有字符按相反的次序重新放置。

3. 求一维 double 型数组 $a[n]$ 中的所有元素之乘积。

4. 计算 $\sum_{i=0}^n \frac{x^i}{i+1}$ 的值。

5. 假定一维整型数组 $a[n]$ 中的每个元素值均在 $[0, 200]$ 区间内, 分别统计出落在 $[0, 20]$, $[20, 50]$, $[50, 80]$, $[80, 130]$, $[130, 200]$ 等各区间内的元素个数。

6. 从二维整型数组 $a[m][n]$ 中查找出最大元素所在的行、列下标。

参考解答如下。

```

1. char Compare(SimpleType x1, SimpleType x2)
{
    if(x1>x2) return '>';
    else if(x1==x2) return '=';
    else return '<';
}

```

时间复杂度为 $O(1)$ 。

```

2. void Reverse(char* p)
{
    int n=strlen(p);
    for(int i=0;i<n/2;i++){
        char ch;
        ch=p[i];
        p[i]=p[n-i-1];
        p[n-i-1]=ch;
    }
}

```

时间复杂度为 $O(n)$ 。

```

3. double Product(double a[], int n)
{
    double p=1;
    for(int i=0; i<n; i++)
        p*=a[i];
    return p;
}

```