

圣才
电子书

圣才考研网

www.100xuexi.com

国内外经典教材辅导系列·计算机类

严蔚敏《数据结构》

(C语言版)

笔记和习题(含考研真题)详解

主编：圣才考研网

www.100xuexi.com



扫码领取
大礼包



中国石化出版社

HTTP://WWW.SINOPEC-PRESS.COM



免费下载2万种考研电子书（考研真题、视频、题库）
登录www.100xuexi.com（圣才考研网）

全国热线：400-900-8858（8:30-00:30）

咨询QQ：4009008858（8:30-00:30）

国内外经典教材辅导系列·计算机类

- ◆ 谢希仁《计算机网络》（第7版）笔记和课后习题（含考研真题）详解
- ◆ 严蔚敏《数据结构》（C语言版）笔记和习题（含考研真题）详解

考研辅导：kaoyan.100xuexi.com（圣才考研网）

资格考试：www.100xuexi.com（圣才学习网）

责任编辑：黄小红 谢青

封面设计：圣才考研网

ISBN 978-7-5114-5696-0



9 787511 456960 >

定价：68.00元

国内外经典教材辅导系列·计算机类

严蔚敏《数据结构》(C语言版) 笔记和习题(含考研真题)详解

主编：圣才考研网

www.100xuexi.com

中国石化出版社

内 容 提 要

国内外经典教材辅导系列是一套全面解析当前国内外各大院校权威教科书的辅导资料。严蔚敏《数据结构》是我国高校采用较多的权威教材之一。本书是该教材的学习辅导书,遵循C语言版的章目编排,共分为12章,每章由三部分组成:第一部分为复习笔记,总结本章的重难点内容;第二部分为强化习题详解,对教材相关的强化习题进行了详细的分析和解答;第三部分为名校考研真题与典型题详解,精选了近年的名校考研真题与典型题,并提供了详细的解答。由于本书第8章以及第12章并非数据结构考试内容,因此本书没有提供相应内容。

购书即可免费享受大礼包增值服务【本书电子书(手机版、电脑版)、考研专业课咨询服务】。手机扫码(本书封面右下角)免费领取本书大礼包。

图书在版编目(CIP)数据

严蔚敏《数据结构》(C语言版)笔记和习题(含考研真题)详解/圣才考研网主编. —北京:中国石化出版社,2020.2

国内外经典教材辅导系列·计算机类
ISBN 978-7-5114-5696-0

I. ①严… II. ①圣… III. ①数据结构-研究生-入学考试-自学参考资料 IV. ①TP311.12

中国版本图书馆CIP数据核字(2020)第026290号

未经本社书面授权,本书任何部分不得被复制、抄袭,或者以任何形式或任何方式传播。版权所有,侵权必究。

中国石化出版社出版发行

地址:北京市东城区安定门外大街58号

邮编:100011 电话:(010)57512500

发行部电话:(010)57512575

<http://www.sinopec-press.com>

E-mail:press@sinopec.com

武汉市盛宏源印务有限公司印刷

全国各地新华书店经销

*

787×1092毫米16开本24.5印张601千字

2020年3月第1版 2020年3月第1次印刷

定价:68.00元

国内外经典教材辅导系列·计算机类

编 委 会

主编：圣才考研网(www.100xuexi.com)

编委：程立刚 娄旭海 李 雪 段瑞权 赵芳微
涂幸运 黄 顺 赵立亭 张月华 肖 娟
夏 蜜 倪彦辉 查 慧 万军辉 李 蕊

序 言

我国各大院校一般都把国内外通用的权威教科书作为本科生和研究生学习专业课程的参考教材,这些教材甚至被很多考试(特别是硕士和博士研究生入学考试)和培训项目作为指定参考书。为了帮助读者更好地学习专业课,我们有针对性地编著了一套学习国内外教材的复习资料,并提供配套的名师讲堂、电子书和题库。

严蔚敏《数据结构》(C语言版)是我国高校采用较多的权威教材之一。作为该教材的学习辅导书,本书具有以下几个方面的特点:

1. 整理名校笔记,浓缩内容精华。本书每章的复习笔记部分对该章的重难点进行了整理,同时对重要知识点进行点拨,因此,本书的内容几乎浓缩了配套教材的知识精华。

2. 归纳经典习题,强化知识考点。为了进一步巩固和强化各章知识难点的复习,增加了强化习题及详解,并对相关知识点进行归纳和延伸,梳理知识点逻辑关系,以达到高效复习的目的。

3. 精选考研真题,巩固重难点知识。为了强化对重要知识点的理解,本书精选了部分名校近几年的数据结构考研真题以及考研中的典型题,这些高校大部分以该教材作为考研参考书目。所选考研真题基本涵盖了各个章节的考点和难点,特别注重联系实际,凸显当前热点。

与本书相配套,圣才考研网提供严蔚敏《数据结构》(C语言版)电子书、题库。

购书即可免费享受大礼包增值服务,手机扫码(本书封面右下角)免费领取本书大礼包。具体包括:①本书电子书(手机版、电脑版);②考研专业课咨询服务。

圣才考研网(www.100xuexi.com)是圣才学习网旗下的考研考博专业网站,提供考研公共课和全国500所院校考研考博专业课辅导【一对一辅导、网授精讲班等】、电子书、题库、全套资料(历年真题及答案、笔记讲义等)、国内外经典教材名师讲堂、考研教辅图书等。

考研辅导: kaoyan.100xuexi.com(圣才考研网)

官方总站: www.100xuexi.com(圣才学习网)

圣才考研网编辑部

目 录

第1章 绪论	(1)
1.1 复习笔记	(1)
1.2 强化习题详解	(4)
1.3 考研真题与典型题详解	(18)
第2章 线性表	(25)
2.1 复习笔记	(25)
2.2 强化习题详解	(28)
2.3 考研真题与典型题详解	(66)
第3章 栈和队列	(71)
3.1 复习笔记	(71)
3.2 强化习题详解	(75)
3.3 考研真题与典型题详解	(101)
第4章 串	(108)
4.1 复习笔记	(108)
4.2 强化习题详解	(112)
4.3 考研真题与典型题详解	(132)
第5章 数组和广义表	(137)
5.1 复习笔记	(137)
5.2 强化习题详解	(141)
5.3 考研真题与典型题详解	(169)
第6章 树和二叉树	(173)
6.1 复习笔记	(173)
6.2 强化习题详解	(181)
6.3 考研真题与典型题详解	(229)
第7章 图	(235)
7.1 复习笔记	(235)
7.2 强化习题详解	(241)
7.3 考研真题与典型题详解	(280)
第8章 动态存储管理	(286)
第9章 查找	(287)
9.1 复习笔记	(287)
9.2 强化习题详解	(293)
9.3 考研真题与典型题详解	(319)

第 10 章 内部排序	(328)
10.1 复习笔记	(328)
10.2 强化习题详解	(336)
10.3 考研真题与典型题详解	(367)
第 11 章 外部排序	(374)
11.1 复习笔记	(374)
11.2 强化习题详解	(375)
11.3 考研真题与典型题详解	(382)
第 12 章 文 件	(385)

第1章 绪论

1.1 复习笔记

一、什么是数据结构

数据结构是一门研究非数值计算的程序设计问题中计算机的操作对象以及它们之间的关系和操作等的学科。

数据元素之间的相互关系称为结构，数据结构是相互之间存在一种或多种特定关系的数据元素的集合。

二、基本概念和术语

1. 数据、数据元素和数据对象的概念(见表 1-1)

表 1-1 基本概念

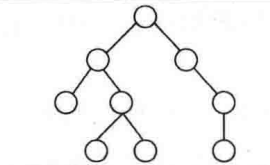
概念	主要内容
数据	对客观事物的符号表示，是计算机科学中所有能输入到计算机中并能被计算机程序处理的符号的总称
数据元素	数据的基本单位。若干个数据项构成一个数据元素，数据项是构成数据元素的最小单位
数据对象	性质相同的数据元素的集合，是数据的一个子集
数据类型	一个值的集合和定义在这个值集上的一组操作的总称
	抽象数据类型
	多形数据类型

2. 数据结构

(1) 基本概念：数据结构是相互之间存在一种或多种特定关系的数据元素的集合。

(2) 数据结构的四类基本基本结构(见表 1-2)

表 1-2 数据结构的四类基本结构

结构	数据元素之间的关系	关系图
集合	属于“同一个集合”，无其他复杂关系	
线性结构	一对一的关系	
树形结构	一对多的关系	
图状结构(网状结构)	多对多的关系	

(3) 数据结构的形式定义

数据结构的形式定义为：

$$\text{Data_Structure} = (D, S)$$

其中：D 表示数据元素的有限集，S 表示 D 上关系的有限集。

(4) 数据结构在计算机中的表示

数据结构包括数据元素的表示和关系，在计算机中称为数据的物理结构（又称存储结构）。

其中，关系有两种表示方法：顺序映象和非顺序映象。这两种表示方法对应两种存储结构：顺序存储结构和链式存储结构。

(5) 数据结构的三要素

①数据的逻辑结构：从逻辑上表示数据，与具体存储无关。常见的逻辑结构如图 1-1 所示。

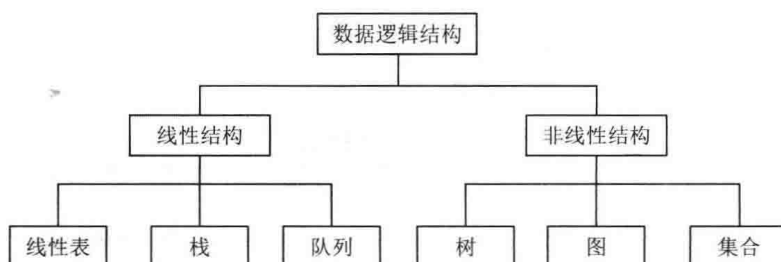


图 1-1

②数据的存储(物理)结构：指数据的逻辑结构在计算机中的具体实现，依赖于计算机语言。常见的存储结构如图 1-2 所示。



图 1-2

③数据的运算：运算包括定义和实现两部分，运算的定义指出该运算所实现的功能，与数据的逻辑结构相对应，运算的实现指出每一个具体的操作步骤，对应于数据的存储结构。

三、抽象数据类型的表示与实现

抽象数据类型(ADT)由一个值域和定义在该值域上的一组操作组成，它与计算机内部的实现无关。

抽象数据类型可用三元组(数据对象，数据关系，基本操作集)来表示

$$(D, S, P)$$

其中，D 是数据对象，S 是 D 上的数据关系集，P 是对 D 的基本操作集。

定义抽象数据类型：

ADT 抽象数据类型名

{

 数据对象：<数据对象的定义>

 数据关系：<数据关系的定义>

基本操作：<基本操作的定义>

} ADT 抽象数据类型名

四、算法和算法分析

1. 算法的描述

算法是对特定问题求解步骤的一种描述，它是指令的有限序列，其中每一条指令表示一个或多个操作。算法需要用一种语言来描述，程序框图，程序设计语言等都能对算法进行描述。

2. 算法的五个重要特性(见表 1-3)

表 1-3 算法的特性

特性	主要内容
有穷性	在执行有穷步之后结束，每一步在有穷时间内完成
确定性	每一条指令都无二义性，算法只有唯一的一条执行路径
可行性	算法中描述的操作都可以通过已实现的基本运算执行有限次来实现
输入	一个算法有零个或多个输入
输出	一个算法有一个或多个输出

3. 算法设计的要求(见表 1-4)

表 1-4 算法设计的要求

要求	主要内容
正确性	程序无语法错误，对于一般和典型输入数据可得到满足要求的结果，对于一切合法输入都能得到满足要求的结果
可读性	算法可以被人们很好地阅读和理解
健壮性	输入非法数据时，算法能适当的做出反应或进行处理，不会产生异常输出或中止程序执行
效率与低存储量需求	与问题规模有关，执行时间短的算法效率高，算法执行中需要的最大存储空间为存储量需求

4. 算法效率的度量

(1)算法执行时间通过该算法编制的程序在计算机上运行时所消耗的时间来度量，度量一个程序的执行时间通常有两种方法：事后统计的方法和事前分析估算的方法。

(2)算法的时间复杂度是对算法效率的度量。一般情况下，算法中基本操作重复执行的次数是问题规模 n 的某个函数，记作 $f(n)$ ，时间复杂度是关于问题规模的函数，通常用 O 表示，记作

$$T(n) = O(f(n))$$

其中 n 表示问题的规模，上式表示算法执行时间的增长率和问题规模 n 的某个函数 $f(n)$ 的增长率相同。

(3)常见时间复杂度按照数量级递增排列为：

$$O(1) < O(\log_2 n) < O(n) < O(n \log_2 n) < O(n^2) < O(n^3) < O(n^k) < O(2^n)$$

5. 算法的存储空间需求

算法的空间复杂度是对算法运行所需要的存储空间的度量。算法的存储空间需求记作

$$S(n) = O(f(n))$$

其中 n 表示问题的规模。

一个程序执行除了需要存储空间存放本身数据外，也需要存储一些实现计算所需信息的辅助空间，若输入数据所占空间与算法无关，只取决于问题本身，则只需要分析除了输入和程序之外的额外空间。

原地工作：算法运行所需的辅助空间相对输入数据量来说是常量。

1.2 强化习题详解

1. 简述下列术语：数据、数据元素、数据对象、数据结构、存储结构、数据类型和抽象数据类型。

答：(1) 数据是对客观事物的符号表示，在计算机科学中是指所有能输入到计算机中并能被计算机程序处理的符号的总称。

(2) 数据元素是数据的基本单位。

(3) 数据对象是性质相同的数据元素的集合，是数据的一个子集。

(4) 数据结构是相互之间存在一种或多种特定关系的数据元素的集合。

(5) 存储结构是数据结构在计算机中的表示(又称映象或数据的物理结构)。

(6) 数据类型是一个值的集合和定义在这个值集上的一组操作的总称。

(7) 抽象数据类型是指一个数学模型以及定义在该模型上的一组操作。

2. 试描述数据结构和抽象数据类型的概念与程序设计语言中数据类型概念的区别。

答：(1) 数据结构：相互之间存在一种或多种特定关系的数据元素的集合；

(2) 抽象数据类型：一个数学模型以及在该模型上的一组操作。

(3) 区别：

① 抽象数据类型的定义仅取决于它的一组逻辑特性，而与其在计算机内部如何表示和实现无关。

② 抽象数据类型包含一般数据类型的概念，但含义比一般数据类型的范畴更广。

③ 抽象数据类型可通过程序设计语言中的数据类型来表示和实现。

3. 设有数据结构 (D, R) ，其中 $D = \{d_1, d_2, d_3, d_4\}$ ， $R = \{r\}$ ， $r = \{(d_1, d_2), (d_2, d_3), (d_3, d_4)\}$ ，试按图论中图的画法惯例画出其逻辑结构图。

答：逻辑结构图如图 1-3 所示。

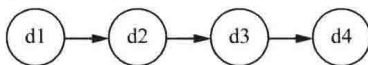


图 1-3 程序逻辑结构图

4. 试仿照三元组的抽象数据类型分别写出抽象数据类型复数和有理数的定义(有理数是其分子、分母均为自然数且分母不为零的分数)。

答：

ADT Complex {

数据对象： $D = \{r, i | r, i \text{ 为实数}\}$

数据关系： $R = \{ \langle r, i \rangle \}$

基本操作：

InitComplex(&C, re, im)

操作结果：构造一个复数 C，其实部和虚部分别为 re 和 im

DestroyComplex(&C)

操作结果：销毁复数 C

Get(C, k, &e)

操作结果：用 e 返回复数 C 的第 k 元的值

Put(&C, k, e)

操作结果：改变复数 C 的第 k 元的值为 e

IsAscending(C)

操作结果：如果复数 C 的两个元素按升序排列，则返回 1，否则返回 0

IsDescending(C)

操作结果：如果复数 C 的两个元素按降序排列，则返回 1，否则返回 0

Max(C, &e)

操作结果：用 e 返回复数 C 的两个元素中值较大的一个

Min(C, &e)

操作结果：用 e 返回复数 C 的两个元素中值较小的一个

} ADT Complex

ADT RationalNumber{

数据对象：D = { s, m | s, m 为自然数, 且 m 不为 0 }

数据关系：R = { < s, m > }

基本操作：

InitRationalNumber(&R, s, m)

操作结果：构造一个有理数 R，其分子和分母分别为 s 和 m

DestroyRationalNumber(&R)

操作结果：销毁有理数 R

Get(R, k, &e)

操作结果：用 e 返回有理数 R 的第 k 元的值

Put(&R, k, e)

操作结果：改变有理数 R 的第 k 元的值为 e

IsAscending(R)

操作结果：若有理数 R 的两个元素按升序排列，则返回 1，否则返回 0

IsDescending(R)

操作结果：若有理数 R 的两个元素按降序排列，则返回 1，否则返回 0

Max(R, &e)

操作结果：用 e 返回有理数 R 的两个元素中值较大的一个

Min(R, &e)

操作结果：用 e 返回有理数 R 的两个元素中值较小的一个

} ADT RationalNumber

5. 试画出与下列程序段等价的框图。

(1)

```
product = 1;  
i = 1;  
while( i <= n)  
{  
    product *= i;  
    i ++;  
}
```

(2)

```
i = 0;  
do  
{  
    i ++;  
} while( ( i != n) && ( a[ i ] != x ) );
```

(3)

```
switch  
{  
    case x < y:  
        z = y - x;  
        break;  
    case x == y:  
        z = abs( x * y );  
        break;  
    default:  
        z = ( x - y) / abs( x) * abs( y); //abs( ) 为取绝对值函数  
}
```

答：以上程序段等价框图分别如图 1-4，图 1-5 和图 1-6 所示。

(1)

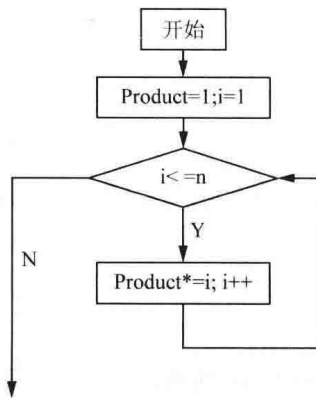


图 1-4 等价框图(1)

(2)

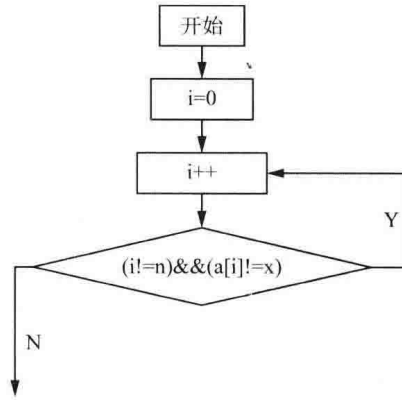


图 1-5 等价框图(2)

(3)

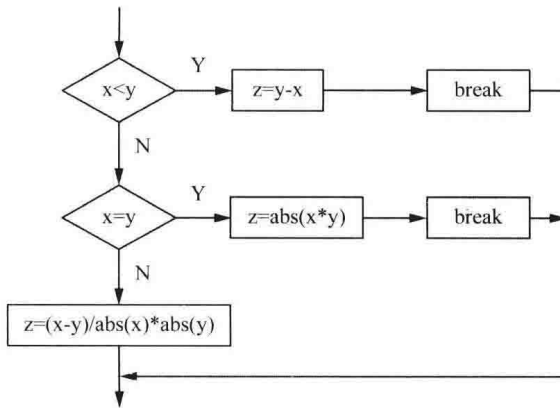


图 1-6 等价框图(3)

6. 在程序设计中，常用下列三种不同的出错处理方式：

- (1) 用 `exit` 语句终止执行并报告错误；
- (2) 以函数的返回值区别正确返回或错误返回；
- (3) 设置一个整型变量的函数参数以区别正确返回或某种错误返回。

试讨论这三种方法各自的优缺点。

答：(1) 优点：`exit` 用于异常错误处理，可以强行中断程序的执行，并返回至操作系统，操作系统会自动回收资源。

缺点：退出地点太多不利于调试。

(2) 优点：以函数的返回值区别正确返回或错误返回，常用于子程序的测试，便于实现程序的局部控制，不会直接终止程序的运行。

缺点：判断太多，必须人工维护一份错误值列表。

(3) 优点：用整型函数进行错误处理可以给出错误类型，便于迅速确定错误。

缺点：需要完整的整型变量的释义，才能方便理解。

7. 在程序设计中，可采用下列三种方法实现输出和输入：

- (1) 通过 `scanf` 和 `printf` 语句；
- (2) 通过函数的参数显式传递；
- (3) 通过全局变量隐式传递。

试讨论这三种方法的优缺点。

答：(1)用 scanf 和 printf 直接进行输入输出

优点：形象、直观。

缺点：需要格式控制，较为烦琐，如果出现错误，则会引起整个系统的崩溃，参数类型受限制，内存开销大。

(2)通过函数的参数显式传递进行输入输出

优点：便于实现信息的隐蔽，减少出错的可能。

缺点：参数说明比较烦琐，内存开销大。

(3)通过全局变量的隐式传递进行输入输出

优点：方便，只需修改变量的值即可，内存开销小；

缺点：过多的全局变量使程序的维护较为困难，内容极易丢失。

8. 设 n 为正整数。试确定下列各程序段中前置以记号@的语句的频度：

(1)

```
i = 1;
k = 0;
while (i <= n - 1)
{
@   k += 10 * i;
    i ++;
}
```

(2)

```
i = 1;
k = 0;
do
{
@   k += 10 * i;
    i ++;
} while (i <= n - 1);
```

(3)

```
i = 1;
k = 0;
while (i <= n - 1)
{
    i ++;
@   k += 10 * i;
}
```

(4)

```
k = 0;
for (i = 1; i <= n; i ++ )
{
```

```

    for(j = i; j <= n; j++)
@   k++;
}

```

(5)

```

for(i = 1; i <= n; i++)
{
    for(j = 1; j <= i; j++)
        for(k = 1; k <= j; k++)
@   x += delta;
}

```

(6)

```

i = 1;
j = 0;
while(i + j <= n)
{
@   if(i > j)
        j++;
    else
        i++;
}

```

(7)

```

x = n; //n 是不小于 1 的常数
y = 0;
while(x >= (y + 1) * (y + 1))
{
@   y++;
}

```

(8)

```

x = 91;
y = 100;
while(y > 0)
{
@   if(x > 100)
        {
            x -= 10;
            y--;
        } else
            x++;
}

```

答: (1) $n - 1$