

● 本讲内容聚焦

● 典型例题

● 课后作业

数据结构

辅导讲案

主讲教材《数据结构》(清华·C语言版)

夏清国 编著

西北工业大学出版社

TP311.12/148C

2008

精品课·名师讲堂丛书

数据结构 辅导讲案

——主讲教材《数据结构》(清华·C语言版)

夏清国 编著

西北工业大学出版社

【内容简介】 本书主讲教材为严蔚敏、吴伟民等编著的《数据结构(C语言版)》。全书共20讲,每讲内容包括本讲内容聚焦(内容要点精讲,知识结构图解,重点、难点点击)、典型例题、课后作业。附录有三项内容:主讲教材配套习题精选详解、课程考试真题、课后作业和课程考试真题参考答案。

本书可作为大学生自学指导、考研人员系统复习、任课教师教学的参考用书,也可作为考研辅导班的教材。

图书在版编目(CIP)数据

数据结构辅导讲案/夏清国编著. —西安:西北工业大学出版社, 2008.4

(精品课程·名师讲堂丛书)

ISBN 978-7-5612-2354-3

I. 数… II. 夏… 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(2008)第 023809 号

出版发行:西北工业大学出版社

通信地址:西安市友谊西路127号 邮编:710072

电 话:(029) 88493844 88491757

网 址:www.nwpup.com

印刷者:陕西丰源印务有限公司

开 本:850 mm×1 168 mm 1/32

印 张:13

字 数:430千字

版 次:2008年4月第1版 2008年4月第1次印刷

定 价:20.00元

前 言

本书是西北工业大学出版社为落实和推动教育部实施的“精品课程建设工程”和“教育质量工程”，精心策划和组织编写的“精品课程·名师讲堂”丛书之一，主讲教材为严蔚敏、吴伟民等编著的《数据结构（C语言版）》，兼顾其他版本的《数据结构》教材。全书共 20 讲，每讲内容包括：本讲内容聚焦——内容要点精讲，知识结构图解，重点、难点点击；典型例题；课后作业。课后作业均选自严蔚敏、李为民、米宁编著的教材配套习题《数据结构题集（C语言版）》，课后作业的题后所附题号为《数据结构题集（C语言版）》中的题号。附录中有三项内容：主讲教材（《数据结构》（第 2 版））配套习题精选详解；课程考试真题；课后作业和课程考试真题参考答案。

本书倾注了编者对数据结构课程教学内容和教学方法苦心研究的成果，对数据结构课程内容深度、新度、广度、量度的理解和把握，对长期教学实践经验的凝结与升华，对国内外知名大学优秀教学成果的学习与借鉴，对研究生招生考试重点、热点、难点规律

的认识与探索。

本书从指导课程教学、学习和考试的角度，通过对大量涉及内容广、常见及经典的题型提供算法的思想，并对算法进行分析，提示了数据结构的解题方法、解题规律和解题技巧。这对于提高读者分析问题的能力，理解基本要领和理论，开拓解题思路，会起到良好的效果。本书可作为大学生自学指导、考研人员系统复习、任课教师教学的参考用书，也可作为考研辅导班的教材。

本书由夏清国编著，夏翼、姚群参与编写了部分内容，书中所有程序由夏翼整理并进行调试，夏清国负责全书的组织与定稿。在此书的编写与出版过程中，西北工业大学计算机学院和西北工业大学出版社给予了大力支持和帮助，在此谨表诚挚谢意。

编 者

2008年1月

目 录

第 1 讲 绪论	1
1.1 本讲内容聚焦	1
1.2 典型例题	5
1.3 课后作业	8
第 2 讲 线性表 (一)	11
2.1 本讲内容聚焦.....	11
2.2 典型例题.....	17
2.3 课后作业.....	20
第 3 讲 线性表 (二)	21
3.1 本讲内容聚焦.....	21
3.2 典型例题.....	29
3.3 课后作业.....	33
第 4 讲 线性表 (三)	36
4.1 本讲内容聚焦.....	36
4.2 典型例题.....	38
4.3 课后作业.....	40
第 5 讲 栈和队列 (一)	42
5.1 本讲内容聚焦.....	42
5.2 典型例题.....	52
5.3 课后作业.....	58
第 6 讲 栈和队列 (二)	60
6.1 本讲内容聚焦.....	60

6.2	典型例题	66
6.3	课后作业	70
第7讲	串 (一)	71
7.1	本讲内容聚焦	71
7.2	典型例题	84
7.3	课后作业	90
第8讲	串 (二)	92
8.1	本讲内容聚焦	92
8.2	典型例题	98
8.3	课后作业	100
第9讲	数组和广义表 (一)	101
9.1	本讲内容聚焦	101
9.2	典型例题	111
9.3	课后作业	118
第10讲	数组和广义表 (二)	119
10.1	本讲内容聚焦	119
10.2	典型例题	129
10.3	课后作业	130
第11讲	树和二叉树 (一)	132
11.1	本讲内容聚焦	132
11.2	典型例题	142
11.3	课后作业	143
第12讲	树和二叉树 (二)	144
12.1	本讲内容聚焦	144
12.2	典型例题	147
12.3	课后作业	152

第 13 讲	树和二叉树 (三)	154
13.1	本讲内容聚焦	154
13.2	典型例题	162
13.3	课后作业	162
第 14 讲	图 (一)	165
14.1	本讲内容聚焦	165
14.2	典型例题	174
14.3	课后作业	176
第 15 讲	图 (二)	178
15.1	本讲内容聚焦	178
15.2	典型例题	194
15.3	课后作业	204
第 16 讲	动态存储管理	206
16.1	本讲内容聚焦	206
16.2	典型例题	217
16.3	课后作业	220
第 17 讲	查找	222
17.1	本讲内容聚焦	222
17.2	典型例题	235
17.3	课后作业	243
第 18 讲	内部排序	245
18.1	本讲内容聚焦	245
18.2	典型例题	259
18.3	课后作业	270
第 19 讲	外部排序	272
19.1	本讲内容聚焦	272

19.2 典型例题.....	277
19.3 课后作业.....	278
第 20 讲 文件	279
20.1 本讲内容聚焦.....	279
20.2 典型例题.....	284
附录	288
一、主讲教材配套习题精选详解.....	288
二、课程考试真题.....	342
三、课后作业和课程考试真题参考答案.....	352

第1讲

绪论

本讲涵盖了主讲教材第1章的内容(2学时)。

1.1 本讲内容聚焦



一、内容要点精讲

(一)什么是数据结构

1. 用计算机解决一个具体问题需经过的几个步骤

(1) 从具体问题抽象出一个适当的数学模型;

(2) 设计一个解此数学模型的算法;

(3) 编出程序,进行调试,调整直至得到最终解。

2. 问题的类型

例1:图书馆的书目检索系统自动化问题。

小结:在此类文档管理的数学模型中,计算机处理的对象之间通常存在着的一种最简单的线性关系,这类数学模型——线性数据结构。

例2:计算机和人对奕问题。

小结:在此类从一种格局能够派生出多种新格局的某些非数值计算问题的数学模型中,计算机处理的对象之间存在着——一对多的关系,通过图示出来,可得一棵倒置的“树”,这类数学模型——树形数据结构,它是非线性的。

例3:多叉路口交通灯的管理问题。

小结:在类似交通、道路等问题的数学模型中,计算机处理的对象之间存在着多对多的关系,通过图示出来,是一种类似“网”的形状,这类数学模型——网状数据结构,它是非线性的。

3. 数据结构的定义

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关系和操作等的学科。

(二)基本概念和术语

1. 数据

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被计算机程序处理的符号的总称。

2. 数据元素

数据元素是数据的基本单位,在计算机程序中通常作为一个整体进行考虑和处理。一个数据元素可由若干个数据项组成。

3. 数据项

数据项是数据的不可分割的最小单位。

4. 数据对象

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。

5. 数据结构

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。数据结构包括三方面的内容:数据的逻辑结构、数据的存储结构和数据的运算。

6. 数据的逻辑结构

数据的逻辑结构:指数据元素之间的逻辑关系,即从逻辑关系上描述数据。它与数据的存储无关,是独立于计算机的。通常有下列4种基本结构:

(1) 集合:结构中的数据元素之间除了“同属于一个集合”的关系外,别无其他关系。

(2) 线性结构:结构中的数据元素之间存在一对一的关系。若结构是非空集,则有且仅有一个开始节点和一个终端节点,并且除开始节点无直接前驱和终端节点无直接后继外,其他所有节点都只有一个直接前驱和一个直接后继。

(3) 树形结构:结构中的数据元素之间存在一对多的关系。若结构是非空集,则除第一个节点外,其他所有节点都只有一个直接前驱,除叶子节点外,其他所有节点可能有多个直接后继。

(4) 图状结构或网状结构:结构中的结构元素之间存在多对多关系。若结构是非空集,所有节点都可能有多直接前驱和多个直接后继。

7. 数据的存储结构

数据的存储结构:指数据元素及其关系在计算机存储器内的表示(也称为映像)。数据的存储结构是逻辑结构用计算机语言的实现,它依赖于计算机语言。通常有下列4种存储映像方法:

(1) 顺序存储方法:该方法是把逻辑上相邻的节点存储在物理位置上相邻

的存储单元里,节点间的逻辑关系由存储单元的邻接关系来体现,由此得到的存储结构称为顺序存储结构,通常顺序存储结构是借助于程序语言的数组来描述的。

(2) 链接存储方法:该方法不要求逻辑上相邻的节点在物理位置上也相邻,节点间的逻辑关系是由附加的指针字段表示的,由此得到的存储表示称为链式存储结构,通常要借助于程序语言的指针类型来描述它。

(3) 索引存储方法:该方法通常是在存储节点信息的同时,还建立附加的索引表。索引表中的每一项称为索引项,索引项的一般形式是:(关键字,地址)。其中关键字唯一标识节点,地址作为指向节点的指针。

(4) 散列存储方法:该方法的基本思想是根据节点的关键字直接计算出该节点的存储地址。

8. 数据的运算

数据的运算是在数据的逻辑结构上定义的操作算法,如检索、插入、删除、更新和排序等。

9. 数据类型

(1) 原子类型:其值不可再分的数据类型。

(2) 结构类型:其值可以再分解为若干成分(分量)的数据类型。

(3) 抽象数据类型:抽象数据组织和与之相关的操作。

10. 数据的操作

数据的基本操作主要有以下几种:

(1) 插入:在数据结构中的指定位置上增添新的数据元素。

(2) 删除:删去数据结构中某个指定的数据元素。

(3) 更新:改变数据结构中某个数据元素的值,在概念上等价于删除和插入操作的组合。

(4) 查找:在数据结构中寻找满足某个特定要求的数据元素的位置或值。

(5) 排序:(在线性结构中)重新安排数据元素之间的逻辑顺序关系,使之按值由小到大或由大到小的次序排列。

(三) 算法和算法分析

1. 算法

• 算法: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,其中每条指令表示一个或多个操作。

• 特性:算法有以下5个主要特征:

(1) 有穷性:一个算法必须总是(对任何合法的输入)在执行有穷步之后结束,且每一步都可在有穷时间内完成。

(2) 确定性:算法中每一条指令必须有确切的含义,确保不会产生二义性。并且,在任何条件下,算法只有唯一的一条执行路径,即对于相同的输入只能得出相同的输出。

(3) 可行性:一个算法是能执行的,即算法中描述的操作都是可以通过已实现的基本运算执行有限次来实现的。

(4) 输入性:一个算法有0个或多个的输入。

(5) 输出性:一个算法有一个或多个的输出。

2. 算法设计的要求

通常设计一个“好”的算法应考虑达到以下目标:

(1) 正确性:算法应当满足具体问题的要求。

(2) 可读性:可读性好有助于人对算法的理解,并易于调试和修改。

(3) 健壮性:当输入数据非法时,算法也能适当地作出反应或进行处理,而不会产生莫名其妙的输出结果。

(4) 效率与低存储量需求:效率是指算法的执行时间,而存储量是指算法执行过程中所需要的最大存储空间,它们都与问题的规模有关。

3. 算法效率的度量

(1) 时间复杂度。一个语句的频度,是指该语句在算法中被重复执行的次数。算法中所有语句的频度之和记作 $T(n)$,它是该算法所求解问题规模 n 的函数。当问题的规模趋向无穷大时, $T(n)$ 的数量级称为渐近时间复杂度,简称为时间复杂度,记作 $T(n) = O(f(n))$ 。

算法的时间复杂度不仅仅依赖于问题的规模,也取决于输入实例的初始状态。一个问题的输入实例是满足问题陈述中所给出的限制和为计算该问题的解所需要的所有输入构成的。

最坏时间复杂度是指在最坏情况下算法的时间复杂度。

平均时间复杂度是指所有可能的输入实例均以等概率出现的情况下,算法的期望运行时间。

上述表达式中“ O ”的含义是 $T(n)$ 的数量级,其严格的数学定义是:若 $T(n)$ 和 $f(n)$ 是定义在正整数集合上的两个函数,则存在正的常数 C 和 n_0 ,使得当 $n \geq n_0$ 时,都满足 $0 \leq T(n) \leq Cf(n)$ 。

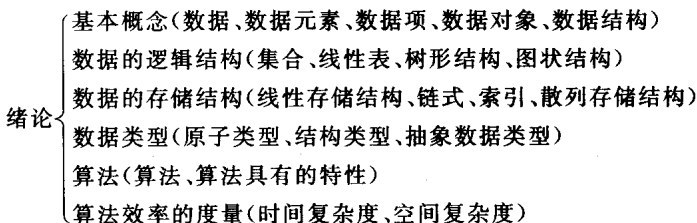
一般总是考虑在最坏的情况下的时间复杂度,以保证算法的运行时间不会

比它更长。

(2) 空间复杂度。算法的空间复杂度 $S(n)$, 定义为该算法所耗费的存储空间, 它是问题规模 n 的函数。渐进空间复杂度也常常简称为空间复杂度, 记作 $S(n) = O(f(n))$ 。



二、知识结构图解



三、重点、难点点击

重点: 数据结构的逻辑结构、存储结构、数据运算三方面的概念及相互关系; 算法的时间复杂度分析。

难点: 分析算法的时间复杂度。

1.2 典型例题

【例 1.1】 逻辑结构和存储结构之间的关系?

【解】 对于已经建立的逻辑结构是设计人员根据解题需要选定的数据组织形式, 因此建立的机内表示应遵循选定的逻辑结构, 所建立数据的机内表示称为数据存储结构。

【例 1.2】 设有数据逻辑结构为 $line = (D, R)$, 其中

$$D = \{01, 02, 03, 04, 05, 06, 07, 08, 09, 10\}$$

$$R = \{r\}$$

$$r = \{ \langle 05, 01 \rangle, \langle 01, 03 \rangle, \langle 03, 08 \rangle, \langle 08, 02 \rangle, \langle 02, 07 \rangle, \langle 07, 04 \rangle, \langle 04, 06 \rangle, \langle 06, 09 \rangle, \langle 09, 10 \rangle \}$$

试分析该数据结构属于哪种逻辑结构。

【分析】 根据题中所给的数据逻辑结构, 得到对应的图形如图 1.1 所示。

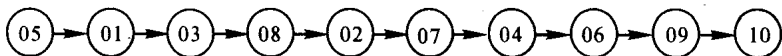


图 1.1 数据的线性结构示意图

在 line 中,每个数据元素有且仅有一个直接前驱元素(除结构中第一个元素 05 外),有且仅有一个直接后继元素(除结构中最后一个元素 10 外)。这种数据结构的特点是数据元素之间的 1 对 1(1:1)关系,即线性关系。

【解】 本题所给定的数据结构为线性结构。

【评注】 注意线性结构的特点。

【例 1.3】 设有数据逻辑结构为 $\text{graph} = (D, R)$, 其中

$$D = \{01, 02, 03, 04, 05, 06, 07\}$$

$$R = \{r\}$$

$$r = \{ \langle 01, 02 \rangle, \langle 02, 01 \rangle, \langle 01, 04 \rangle, \langle 04, 01 \rangle, \langle 02, 03 \rangle, \langle 03, 02 \rangle, \langle 02, 06 \rangle, \langle 06, 02 \rangle, \langle 02, 07 \rangle, \langle 07, 02 \rangle, \langle 03, 07 \rangle, \langle 07, 03 \rangle, \langle 04, 06 \rangle, \langle 06, 04 \rangle, \langle 05, 07 \rangle, \langle 07, 05 \rangle \}$$

试分析该数据结构属于哪种逻辑结构。

【分析】 根据题中所给的数据逻辑结构,得到对应的图形如图 1.2 所示。

从图 1.2 可以看出, r 是 D 上的对称关系,为了简化起见,我们把 $\langle x, y \rangle$ 和 $\langle y, x \rangle$ 这两个对称序偶用一个无序对 (x, y) 或 (y, x) 来代替;在图形表示中,我们把 x 节点和 y 节点之间两条相反的有向边用一条无向边来代替。这样 r 关系可改写为

$$r = \{(01, 02), (01, 04), (02, 03), (02, 06), (02, 07), (03, 07), (04, 06), (05, 07)\}$$

对应的图形如图 1.3 所示。

从图 1.2 或图 1.3 可以看出,节点之间的联系是 M 对 $N(M:N)$ 联系($M \geq 0, N \geq 0$),即网状关系。也就是说,每个节点可以有多个前驱节点和多个后继节点。

【解】 本题所给定的数据结构为图状结构。

【评注】 注意图状结构的特点。

【例 1.4】 设 n 为整数,指出下列各算法的时间复杂度。

```
(1) void prime(int n)           // n 为一个正整数
    { int i = 2;
```

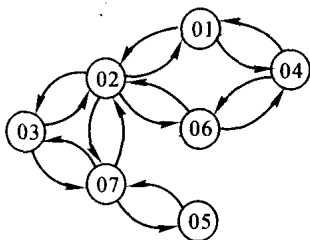


图 1.2 数据的图状结构示意图

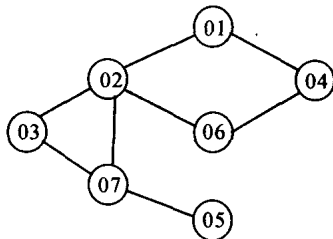


图 1.3 图 1.2 的等价表示

```
while((n%i) != 0 && i * 1.0 < sqrt(n))
    i++;
if(i * 1.0 > sqrt(n))
    printf("%d 是一个素数\n", n);
else
    printf("%d 不是一个素数\n", n);
}
```

(2) sum1(int n) // n 为一个正整数

```
{ int p = 1, sum = 0, i;
  for(i=1; i<=n; i++)
  { p *= i;
    sum += p;
  }
  return sum;
}
```

(3) sum2(ihtn) // n 为一个正整数

```
{ int sum = 0, i, j;
  for(i=1; i<=n; i++)
  { p = 1;
    for(j=1; j<=i; j++)
      p *= j;
    sum += p;
  }
}
```

```
return sum;
```

```
}
```

【分析】

(1) 算法的时间复杂度是由嵌套最深层语句的执行次数决定的。prime 算法的嵌套最深层语句为

```
i++;
```

它的执行次数由条件 $((n \% i) != 0 \ \&\& \ i * 1.0 < \sqrt{n})$ 决定, 显然执行次数小于 $\sqrt{n}$, 所以 prime 算法的时间复杂度是 $O(n^{1/2})$ 。

(2) 算法的时间复杂度是由嵌套最深层语句的执行次数决定的。sum1 算法的嵌套最深层语句为

```
p *= i;
```

```
sum += p;
```

它的执行次数为 n 次, 所以 sum1 算法的时间复杂度是 $O(n)$ 。

(3) 算法的时间复杂度是由嵌套最深层语句的执行次数决定的。sum2 算法的嵌套最深层语句为

```
p *= i;
```

它的执行次数为 $1+2+3+\dots+n=n(n+1)/2$ 次, 所以 sum2 算法的时间复杂度是 $O(n^2)$ 。

【解】 各算法的时间复杂度分别为 $O(n^{1/2}), O(n), O(n^2)$ 。

【评注】 决定时间复杂度的主要是循环, 要分析时间复杂度, 应从循环入手。

【例 1.5】 将数量级 $O(1), O(n), O(n^2), O(n^3), O(n \log n), O(\log n), O(2^n)$ 按增长率从小到大排列。

【分析】 在题目给出的 7 种类型的数量级中: $O(1)$ 为常量型, $O(n)$ 为线性型, $O(n^2)$ 为平方型, $O(n^3)$ 为立方型, $O(n \log n)$ 为线性对数型, $O(\log n)$ 为对数型, $O(2^n)$ 为指数型。

【解】 以上 7 种类型按增长率从小到大排列如下:

$$O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(n^3) < O(2^n)$$

【评注】 可用洛必达法则判断。

1.3 课后作业

1.1 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