

21
世纪

高等学校计算机类规划教材

数据结构

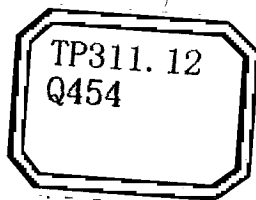
习题解答与实验指导

主编◎秦玉平 马



大连理工大学出版社
大连理工大学电子音像出版社

TP311.12
Q454



数据结构

习题解答与实验指导

主编 秦玉平 马靖善

大连理工大学出版社
大连理工大学电子音像出版社

内 容 提 要

本书是针对清华大学出版社出版的《数据结构》(秦玉平和马靖善主编)一书编写的,但也适合于做其他数据结构教材的辅导材料。

本书对教材中的所有习题都做了分析和解答,题目分为单选题、判断题、算法填空题、计算操作题和算法设计题五种类型。针对教学重点和难点,根据教材内容给出了十六组实验题目,既便于学生理解和消化教材的内容,又便于教师组织实验教学。另外,本书给出了程序员考试和研究生入学考试的样题和答案,便于学生的复习。

本书的特点是习题类型多,覆盖面广,且都是从易到难的顺序编写,既有利于调动学生学习的积极性和主动性,又有利于培养学生分析问题和解决问题的能力。

本书可作为大学本科、大学专科、夜大、函大和自学考试的教学指导书。也可作为全国计算机等级考试和程序员考试的参考书。

数据结构习题解答与实验指导

主编:秦玉平 马靖善

责任编辑:刘 剑 责任校对:达 理

封面设计:李 强

出版:大连理工大学出版社、大连理工大学电子音像出版社

地址:大连市软件园路80号

邮编:116023

电话:0411-84708842(发行), 84707464(技术支持)

传真:0411-84701466

邮购:0411-84703636

网址:<http://www.dutp.cn>

E-mail: dzcb@dutp.cn

印刷:大连理工印刷有限公司印制

幅面尺寸:185mm×230mm

印张:7

印数:3001~5000

字数:134千字

出版时间:2005年7月第1版

印刷时间:2006年12月第4次印刷

ISBN 7-900670-52-1 定价:13.50元

前 言

《数据结构习题解答与实验指导》是针对清华大学出版社出版的《数据结构》(秦玉平、马靖善主编)教材而编写的。但本书的内容也适用于采用其它数据结构教材的教学。

数据结构是一门十分抽象的课程,涉及到一系列比较实用的算法。课堂的理论教学,使学生掌握算法的基本思想,而要深刻理解,融会贯通课堂教学内容,必须经过大量的习题训练,通过习题训练培养学生独立思考的能力,培养学生分析问题和解决问题的能力。

《数据结构习题解答与实验指导》是按教材的章节顺序编写。题目分为单选题、判断题、算法填空题、计算操作题和算法设计题五种类型。针对教学重点和难点,根据教材内容给出了十六组实验题目。它引导学生认真细致地复习教材内容,深入理解和熟练掌握有关基本概念和方法。最后附有程序员考试、研究生入学考试的样题和答案。

本书第1、4、7章由马靖善编写,第2、3章由秦玉平编写,第5章由周连秋编写,第6章由王丽君编写。

本书由秦玉平、马靖善主编。

本书可作为大学本科、大学专科、夜大、函大和自学考试的教学指导书。也可作为全国计算机等级考试和程序员考试的参考书。

由于水平有限,错误和不妥之处在所难免,欢迎读者批评指教。

编 者
2005年6月

目 录

前 言		
第 1 章	概述.....	1
第 2 章	线性表.....	6
第 3 章	数组和广义表.....	20
第 4 章	树和二叉树	28
第 5 章	图.....	41
第 6 章	查找.....	49
第 7 章	内部排序.....	59
第 8 章	实验	68
附录 I	初级程序员级样题(下午试卷)	71
附录 II	程序员级样题(下午试卷)	77
附录 III	硕士研究生(课程与教学论)入学考试数据结构试题.....	83
附录 IV	硕士研究生(计算机软件与理论)入学考试数据结构试题.....	93

第1章 概述

一、回答下列基本概念

1. 什么叫数据、数据元素、数据项和数据对象？

数据(Data)是对客观事物的符号表示。在计算机学科中，数据是指所有能输入到计算机中，并能被计算机程序所处理的符号的总称。

数据元素(Data Element)是描述数据的基本单位。

数据项(Data Item)是描述数据的最小单位。

数据对象(Data Object)是性质相同的一类数据元素的集合。

2. 什么叫数据结构？分为哪几类？

数据结构就是指相互之间有一种或多种特定关系的数据元素的集合。

数据结构依据抽象描述方式和机内存储形式可分为逻辑结构和物理结构两大类。

(1) 逻辑结构(Logical Structure)是以抽象的数学模型来描述数据结构中数据元素之间的逻辑关系。通常用二元组来描述这种关系：

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

其中，D是数据元素的有限集，R是D上的关系的有限集。

(2) 物理结构(Physical Structure)又称存储结构，是数据结构在计算机内的存储表示，也称内存映象。

3. 什么叫数据类型？什么叫抽象的数据类型？

数据类型(Data Type)就是一个值的集合和定义在这个值集上的一组操作的总称。

抽象的数据类型(Abstract Data Type,简称ADT)是指一个数学模型以及定义在该模型上的一组操作。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，即不论其内部结构如何变化，只要它的数学特性不变，都不影响其外部的使用。抽象的数据类型可以细分为如下三种类型：

(1) 原子类型(Atomic Data Type)：值是不可分的。

(2) 固定聚合类型(Fixed_aggregate Data Type)：其值由确定数目的成分按某种结构组成。

(3) 可变聚合类型(Variable_aggregate Data Type)：其值由不确定数目的成分构成。

4. 什么叫算法？算法有哪几个基本特性？

所谓“算法”(Algorithm)就是对求解问题步骤的一种描述，也称为算法设计。

(1) 有穷性：一个算法必须总是(对任何合法的输入值)在执行有穷步之后结束，且每一步都可在有穷的时间内完成。这也是算法与程序的最主要区别，程序是可以无限地

循环下去的,如操作系统的监控程序在机器启动后就一直在监测着操作者的鼠标动作和输入的命令。

(2) 确定性: 算法中的每一条指令都必须有明确的含义, 不应使读者产生二义性。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到相同的输出。

(3) 可行性: 一个算法是可以被执行的, 即算法中的每个操作都可以通过已经实现的基本运算执行有限次来完成。

(4) 有输入: 根据实际问题需要, 一个算法在执行时可能要接收外部数据, 也可能不需外部输入。所以, 一个算法应有零个或多个输入, 这取决于算法本身要实现的功能。

(5) 有输出: 一个算法在执行完成后, 一定要有一个或多个结果或结论。这就要求算法一定要有输出, 这些输出是同输入有着某些联系的量。

5. 如何评价一个算法的好与坏?

(1) 正确性: 这是算法设计的最基本要求, 算法应该严格地按照特定的规格说明去设计, 要能够解决给定的问题。

(2) 可读性: 设计算法的目的主要是为了人的阅读和交流, 其次才是机器执行。

(3) 健壮性: 当输入不合法数据时, 算法能做出相应的反应或进行适当的处理, 避免带着非法数据执行, 导致莫名其妙的结果。

(4) 高效率: 依据算法编制的程序在相同的运行环境下执行速度较快。

(5) 低存储: 依据算法编制的程序在相同的运行环境下执行时所需内存空间较少。

二、试举例说明常见的时间复杂性有哪些。

名称	时间复杂度 $T(n)$	举例
常量阶	$O(1)$	与问题规模无关的算法
线性阶	$O(n)$	与问题规模相关的单重循环
平方阶	$O(n^2)$	与问题规模相关的二重循环
立方阶	$O(n^3)$	与问题规模相关的三重循环
指数阶	$O(e^n)$	
对数阶	$O(\log_2 n)$	折半查找算法
复合阶	如 $O(n \cdot \log_2 n)$	堆排序算法
其他		过于复杂

三、写出下列算法总的语句执行次数。

1. $y=5; x=1;$

```
while(y<=10)
    if(x==5)
        { x=1;y+=x;}
    else x++;
```

x=	1	2	3	4	5	1...	1...	1...	1...	1
y=	5					6...	7...	8...	9...	10
y<=10	1	1	1	1	1	1...	1...	1...	1...	1
x==5	1	1	1	1	1	1...	1...	1...	1...	1
x=1	0	0	0	0	1	0...	0...	0...	0...	0
y+=x	0	0	0	0	1	0...	0...	0...	0...	0
x++	1	1	1	1	0	1...	1...	1...	1...	1

y 每变换一个值, 各语句执行的总次数为 16 次; 另外, 初始赋值语句 y=5 和 x=1 各执行一次, 在 y=11 时, 条件 y<=10 还要判断一次, 所以整个程序总的语句执行次数为: $16 * 6 + 3 = 99$ 次。

```
2.x=0;
   for(i=0;i<10;i++)
       for(j=0;j<=i;j++)
           x=x+1;
```

x=0;----1 次
i=0;----1 次
i<10;----11 次
i++;----10 次
j=0;----10 次
j<=i;---- $2+3+4+\dots+11=65$ 次
j++;---- $1+2+3+\dots+10=55$ 次
x++;---- $1+2+3+\dots+10=55$ 次

所以, 总的语句执行次数为: $1 * 2 + 11 + 10 * 2 + 65 + 55 * 2 = 208$ 次。

四、判断如下时空性能的计算是否正确。(其中 n 为问题规模, K 为常数)

- () 1. $O(1) = O(2) = \dots = O(100)$
- () 2. $O(1) + O(2) = O(1)$
- () 3. $O(1) + O(n) = O(n)$
- () 4. $O(1) \times n = O(n)$
- () 5. $O(1) \times K = O(n)$

- () 6. $O(n) \times K = O(K \times n) = O(n)$
 () 7. $O(n) \times O(n) = O(n^2)$
 () 8. $O(n) + O(n) = O(n)$
 () 9. $O(n) + O(m) = \max(O(n), O(m))$ m 也是问题规模。
 () 10. $O(n^P) = K_P \times n^P + K_{P-1} \times n^{P-1} + \dots + K_1 \times n^1 + K_0$ $K_i (0 \leq i \leq P)$ 也是常数。
 1.对 2.对 3.对 4.对 5.错($k \neq 0$ 时正确)
 6.错($k \neq 0$ 时正确) 7.对 8.对 9.对 10.错($K_P \neq 0$ 时正确)

五、分析如下各种算法的时空性能。

1. 计算 n 个实数的平均值，并找出其中的最大数和最小数。

```
ave=0,max,min;
float calave(float a[],int n)
{ int i;
  max=min=a[0];
  for(i=0;i<n;i++)
  { ave+=a[i];
    if(max<a[i]) max=a[i];
    if(min>a[i]) min=a[i];
  }
  return ave/n;
}
```

$T(n) = O(n)$ $S(n) = O(1)$

2. 将一个字符串(有 m 个字符)在另一个字符串(有 n 个字符)中出现的字符删除。

```
int found(char *t,char *c)
{ while(*t&&*t!=*c) t++;
  return *t;
}
delchar(char *s,char *t)
{ char *p,*q;
  p=s;
  while(*p)
  if(found(t,*p))
  { q=p;
    while(*q) *q++=*(q+1);
  }
}
```

```
}
```

```
else p++;
```

```
}
```

$T_{\max}(n)=O(n^2*m)$ 最坏情况

$T_{\min}(n)=O(n*m)$ 最好情况

$S(n)=O(1)$

3.用递归法求 $n!$ 值。

```
fun(int n)
```

```
{ int s;
```

```
  if(n<=1) s=1;
```

```
  else s=n*fun(n-1);
```

```
  return s;
```

```
}
```

$T(n)=O(1*n)=O(n)$ n 为递归的深度

$S(n)=O(n)$ n 为栈空间的大小

第2章 线性表

一、单项选择题(下列每小题给出 4 个备选答案, 将其中一个正确答案的序号填在其后的括号内)

1. 下面哪一个不是线性表的特性 ()

- ① 除第一个元素外, 每个元素都有前驱。
- ② 除最后一个元素外, 每个元素都有后继。
- ③ 线性表是有限序列。
- ④ 线性表的长度为 n , $n \neq 0$ 。

答案: ④

分析: 根据线性表的定义, 线性表的长度可以为 0。

2. 下面哪一个不是顺序表的特点 ()

- ① 逻辑上相邻的元素, 存储在计算机中相邻的存储空间。
- ② 插入一个元素, 平均要移动表长一半的数据元素。
- ③ 用动态一维数组存储顺序表最合适。
- ④ 在顺序表中查找一个元素与表中元素的分布没有关系。

答案: ④

3. 在一个单链表中, 已知 q 结点是 p 的前驱结点, 若在 q 和 p 之间插入 s 结点, 则执行 ()

- ① $s \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} = s;$
- ② $p \rightarrow \text{next} = s \rightarrow \text{next}; s \rightarrow \text{next} = p;$
- ③ $q \rightarrow \text{next} = s; s \rightarrow \text{next} = p;$
- ④ $p \rightarrow \text{next} = s; s \rightarrow \text{next} = q$

答案: ③

4. 在一个链队列中, 假定 front 和 rear 分别为队头指针和队尾指针, 则插入 s 所指向结点进行的操作是 ()

- ① $\text{front} \rightarrow \text{next} = s; \text{front} = s;$
- ② $\text{rear} \rightarrow \text{next} = s; \text{rear} = s;$
- ③ $\text{front} = \text{front} \rightarrow \text{next};$
- ④ $\text{front} = \text{rear} \rightarrow \text{next};$

答案: ②

5. 链表不具有的特点是 ()

- ① 可以随机访问任何一个元素
- ② 插入和删除元素不需要移动元素
- ③ 不必事先估计存储空间
- ④ 所需存储空间与链表长度成正比

答案: ①

6. 在具有 n 个单元的顺序存储的循环队列中, 假定 front 和 rear 分别为队首指针和

队尾指针，则判断队列为空的条件是 ()

- ① $front == rear + 1$
- ② $front + 1 == rear$
- ③ $front == rear$
- ④ $front == 0$

答案: ③

7. 串是 ()

- ① 一些符号构成的序列
- ② 一些字母构成的序列
- ③ 一个以上的字符构成的序列
- ④ 任意有限个字符构成的序列

答案: ④

8. 设输入序列为 1, 2, 3, 4, 借助一个栈可以得到的输出序列是 ()。

- ① 1, 3, 4, 2
- ② 3, 1, 4, 2
- ③ 4, 3, 1, 2
- ④ 4, 1, 2, 3

答案: ①

9. 在一个具有 n 个单元的顺序栈中, 假定以地址低端作为栈底, 以 top 作为栈顶指针, 则当作退栈处理时, top 变化为 ()

- ① top 不变
- ② $top += n$
- ③ $top --$
- ④ $top ++$

答案: ③

10. 若循环队列的队头指针为 front, 队尾指针为 rear, 则队长的计算公式为 ()。

- ① $rear - front$
- ② $front - rear$
- ③ $rear - front + 1$
- ④ 以上都不正确

答案: ④

11. 栈和队列都是 ()

- ① 顺序存储的线性表
- ② 链式存储的线性表
- ③ 限制插入、删除操作的线性表
- ④ 限制插入、删除操作的非线性表

答案: ③

12. 若某链表中最常用的操作是在最后一个元素之后插入一个元素和删除最后一个元素, 则采用 () 最节省时间。

- ① 单链表
- ② 双链表
- ③ 带头结点的双循环链表
- ④ 单循环链表

答案: ③

13. 在非空双循环链表中 q 所指向的结点前插入一个由 p 所指结点的过程依次为 ()

- ① $p \rightarrow next = q; p \rightarrow prior = q \rightarrow prior; q \rightarrow prior = p; q \rightarrow next = p;$

- ② p->next=q;p->prior=q->prior;q->prior=p;q->prior->next=p;
- ③ p->next=q;p->prior=q->prior;q->prior=p; p->prior->next=p;
- ④ p->next=q;p->prior=q->prior;q->prior=p;p->next->prior=p;

答案：③

14. 某线性表中最常用的操作是在最后一个元素之后插入一个元素和删除第一个元素，则采用以下哪种存储方式最节省操作时间（ ）。

- ① 单链表
- ② 仅有头指针的单循环链表
- ③ 双链表
- ④ 仅有头指针的双循环链表

答案：④

15. 若线性表采用链式存储，则表中各元素的存储地址（ ）

- ① 必须是连续的
- ② 部分地址必须是连续的
- ③ 一定不是连续的
- ④ 可以连续也可以不连续

答案：④

二、正误判断题(判断每小节的正误，正确打√，错误打×)

1. 单循环链表从任何结点出发可以访问链表中的任何结点(√)。
2. 双循环链表从任何结点出发可以访问该结点的直接前驱和直接后继(√)。
3. 链表中结点数据域占的存储空间越多，存储密度越大(√)。
4. 带头结点和不带头结点的单链表在查找、删除、求长度等操作上无区别(×)。
5. 栈是线性表，线性表的所有操作适用于栈(×)。
6. 队列是特殊的线性表，在队列的两端可以进行同样的操作(×)。
7. 单链表中 p 结点存在后继结点的条件是 p!=NULL.(×)。
8. 如果两个串含有相同的字符，则这两个串相等(×)。
9. 线性表中每个结点都有前驱和后继(×)。
10. 静态链表要求逻辑上相邻的元素在物理位置上也相邻(×)。

三、算法阅读填空题(在_____处填写正确的内容，使算法完整)

1. 下列函数的功能是实现带头结点单链表的逆置。

```
void turn(slink *L)
{
    slink *p,*q;
    p=L->next;
    L->next=NULL;
    while(_____)
    {
        q=p;
        p=p->next;
```

```

    q->next=L->next;
    L->next=_____ ;
}
}

```

答案: p!=NULL

q

2. 下列函数的功能是实现带头结点单链表的结点值按升序排序。

```

void sort2(slink *l1)
{ slink *p,*q,*r,*s;
  p=l1;
  while(p->next!=NULL)
  { q=p->next;
    r=p;
    while(_____)
    { if(q->next->data<r->next->data)
      r=q;
      _____;
    }
    if(_____)
    {s=r->next;
      r->next=s->next;
      s->next=p->next;
      p->next=s;
    }
    _____;
  }
}

```

答案: q->next!=NULL

q=q->next
r!=p
p=p->next

3. 已知 h 是一个带头结点的双链表, 每个结点有四个成员: 指向前驱结点的指针 prior、指向后继结点的指针 next、存放数据的成员 data 和访问频度 freq。所有结点的 freq 初始

值为 0。每当在双链表上进行一次 locate(h,x)操作时，令元素值为 x 的结点的 freq 的值增 1，并使此链表中结点保持按访问频度递减的顺序排列，以便使访问频度高的结点总是靠近表头。

```
locate(dlink *h,ElemType x)
{ dlink *p=h->next,*q;
  while(p!=NULL&&_____ ) p=p->next;
  if(p==NULL) return 0;
  else
  { p->freq++;
    q=p->prior;
    while(q!=h&&_____ )
    { p->prior=q->prior;
      p->prior->next=p;
      q->next=p->next;
      if(_____ )
        q->next->prior=q;
      p->next=q;
      q->prior=p;
      _____;
    }
  }
  return 1;
}
```

答案: p->data!=x
 q->freq<p->freq
 q->next!=NULL
 q=p->prior

4. 已知长度为 len 的线性表 L 采用顺序存储结构。下列算法的功能是删除线性表 L 中所有值为 item 的数据元素。

```
void delnode(sqlist *L,ElemType item)
{ int k=0,i=0;
  while(i<L->len)
  { if(L->data[i]==item)
```

```

        _____;
    else
        L->data[i-k]=L->data[i];
        _____ ;
    }
    L->len=L->len-k;
}

```

答案: k++
i++

四、算法设计题

1. 设 A 和 B 是两个非递减的顺序表。编写算法，把 A 和 B 中都存在的元素组成新的由大到小排列的顺序表 C。

```

A_B(sqlist *L1,sqlist *L2,sqlist *L3)
{
    int i=0,j,k=0;
    while(i<L1->len)
    { j=0;
        while(j<L2->len&&L1->data[i]!=L2->data[j]) j++;
        if(L1->data[i]==L2->data[j])
            L3->data[k++]=L1->data[i];
        i++;
    }
    L3->len=k;
}

```

2. 编写算法，删除单链表 L 中 p 结点的前驱结点。

```

delp_prior(slink *L,slink *p)
{
    slink *q1,*q2;
    if(p==L||p==L->next||p==NULL)
        printf("\nNo exist!");
    else
        { q1=L;q2=L->next;
            while(q2->next!=p)

```



```

    { q1=q2;q2=q2->next;}
    q1->next=p;
    free(q2);
}
}

```

3. 编写算法, 统计字符串 s 中含有子串 t 的个数。要求: 分别用顺序串和链串实现。

```

subscount(st *s,st *t) /* 顺序串 */
{
    int i,n=0,k;
    for(i=1;i<=s->len;i++)
    {
        k=seek(s,t,i);
        if(k)
            {n++;i=k+t->len-1;}
    }
    return n;
}
subscount(lt *s,lt *t) /* 链串 */
{
    int i,n=0,k;
    for(i=1;length(s)-i+1>=length(t);i++)
    {
        k=seek(s,t,i);
        if(k)
            {n++;i=k+length(t)-1;}
    }
    return n;
}

```

4. 编写算法, 删除顺序表 A 中元素值在 x 到 $y(x \leq y)$ 之间的所有元素。

```

delxtoy(sqllist *L,ElemType x,ElemType y)
{ int k=0,i=0;
  while(i<L->len)
    { if(L->data[i]>=x&&L->data[i]<=y)

```