

21世纪高等院校计算机教材系列

数据结构与算法习题 解答及实验指导

●王苗 赵红 李宁 张晓莉 编著

 机械工业出版社
CHINA MACHINE PRESS



21 世纪高等院校计算机教材系列

数据结构与算法 习题解答及实验指导

王 苗 赵 红 李 宁 张晓莉 编著

机械工业出版社

本书与机械工业出版社出版的《数据结构与算法》教材配套使用,主要内容分为2篇:第1篇“学习指导与习题解答”包括9章内容,前8章与《数据结构与算法》(第2版)教材的内容相对应,每一章由重点与难点、典型题分析、课后习题解答和单元练习题组成,题型有加强概念理解的选择题、判断题、填空题,有帮助理解算法思想的简答题,也有提高算法设计能力的算法设计题,第9章给出了不同层次的8套模拟试题,并在附录中给出了试题的解答,以供读者参考;第2篇“实验指导与课程设计”包括两章内容,第10章结合教材的第2~8章,给出了实验教学中各单元的实验题目,并按照软件设计的要求对每个实验题目都给出“问题描述、基本要求、提示与分析、测试数据及选做内容”,引导学生完成实验学习部分,第11章给出了实验报告规范。

本书可作为计算机和信息类相关专业本(专)科“数据结构”课程的辅助教材,也可供相关专业的工程技术人员参考。

图书在版编目(CIP)数据

数据结构与算法习题解答及实验指导/王苗等
编著. —北京:机械工业出版社,2008.2
21世纪高等院校计算机教材系列
ISBN 978-7-111-23356-5

I. 数… II. 王… III. ①数据结构-高等学校-
教学参考资料②算法分析-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011279号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)
责任编辑:刘亚军 责任校对:吴美英
责任印制:邓 博
北京京丰印刷厂印刷
2008年6月第1版·第1次印刷
184mm×260mm·15.5印张·381千字
0 001—4 000册
标准书号:ISBN 978-7-111-23356-5
定价:25.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换
销售服务热线电话:(010) 68326294
购书热线电话:(010) 88379639 88379641 88379643
编辑热线电话:(010) 88379753 88379739
封面无防伪标均为盗版

出版说明

信息技术是当今世界发展最快、渗透性最强、应用最广的关键技术，是推动经济增长和知识传播的重要引擎。在我国，随着国家信息化发展战略的贯彻实施，信息化建设已进入了全方位、多层次推进应用的新阶段。现在，掌握计算机技术已成为 21 世纪人才应具备的基本素质之一。

为了进一步推动计算机技术的发展，满足计算机学科教育的需求，机械工业出版社聘请了全国多所高等院校的一线教师，进行了充分的调研和讨论，针对计算机相关课程的特点，总结教学中的实践经验，组织出版了这套“21 世纪高等院校计算机教材系列”。

本套教材具有以下特点：

- (1) 反映计算机技术领域的新发展和新应用。
- (2) 注重立体化教材的建设，多数教材配有电子教案、习题与上机指导或多媒体光盘等。
- (3) 针对多数学生的学习特点，采用通俗易懂的方法讲解知识，逻辑性强、层次分明、叙述准确而精炼、图文并茂，使学生可以快速掌握，学以致用。
- (4) 符合高等院校各专业人才的培养目标及课程体系的设置，注重培养学生的应用能力，强调知识、能力与素质的综合训练。
- (5) 适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班和自学用书。

机械工业出版社

前言

“数据结构”是计算机等信息类专业的一门核心课程。“数据结构”课程的内容是在长期的程序设计实践中提炼、升华而成的，主要研究在计算机中数据的表示、存储及基本运算的实现。在计算机专业的理论体系中，“数据结构”是“操作系统”、“编译原理”、“数据库原理”等课程的基础，在计算机课程的学习中起承上启下的作用。

“数据结构”课程的学习过程也是进行复杂程序设计的训练过程。“数据结构”的教学知识量大，实践性强，学生常常是能听得懂设计思想，但对做题和实验感觉有很大难度。为了帮助学生深入理解数据结构的基本原理，尽快掌握各种数据结构的表示及运算实现方法，吸取经典算法的设计思想及精髓，提高运用所学知识解决实际问题的能力，并初步掌握算法的时空分析技巧，作者针对“数据结构”教材的内容并结合多年的教学经验编写了本书，对重要的知识点进行了归纳和总结，汇编了“数据结构”中常出现的典型习题的解答，给初学者提供了技能训练的具体方式。同时，为了提高学生程序设计能力，培养好的程序设计风格和习惯，本书根据具体的应用，给出了实验教学过程中的实验要求，规范了实验步骤和实验报告。

本书与机械工业出版社出版的《数据结构与算法》（第2版）教材相配套，主要内容分为2篇：第1篇是“学习指导与习题解答”；第2篇为“实验指导与课程设计”。

“学习指导与习题解答”部分共有9章内容，前8章与《数据结构与算法》（第2版）教材的内容相对应。每一章由重点与难点、典型题分析、课后习题解答和单元练习题组成，题型有加强概念理解的选择题、判断题、填空题，有帮助理解算法思想的简答题，也有提高算法设计能力的算法设计题。第9章给出了不同层次的8套模拟试题，并在附录中给出了试题的解答，以供读者参考。

“实验指导与课程设计部分”根据“数据结构”课程的重点，给出了7个单元的实验指导，读者可从每个单元中自由选择某一个实验来完成。每个实验题目采用了统一的格式，由问题描述、基本要求、提示与分析、测试数据及选做内容几部分组成。

本书可作为计算机和信息类相关专业本（专）科“数据结构”课程的辅助教材，也可供相关专业的工程技术人员参考。

本书在编写过程中力求语言表述准确，解题思路清晰，算法描述规范严谨，但限于编者水平，书中难免存在错误和不妥之处，恳请读者批评指正，编者将不胜感激。

编 者

目 录

出版说明

前言

第1篇 学习指导与习题解答

第1章 绪论	1
1.1 重点与难点	1
1.1.1 基本概念	1
1.1.2 算法及其分析	2
1.2 典型题分析	3
1.2.1 选择题	3
1.2.2 判断题	3
1.2.3 填空题	4
1.3 课后习题解答	5
1.3.1 选择题	5
1.3.2 判断题	5
1.3.3 简答题	5
1.4 单元练习题	8
第2章 线性表	9
2.1 重点与难点	9
2.1.1 基本概念	9
2.1.2 顺序表及运算	9
2.1.3 链表及运算	10
2.2 典型题分析	13
2.2.1 选择题	13
2.2.2 判断题	14
2.2.3 填空题	14
2.2.4 简答题	15
2.2.5 算法设计题	16
2.3 课后习题解答	20
2.3.1 选择题	20
2.3.2 判断题	21
2.3.3 算法设计题	21
2.4 单元练习题	27
2.4.1 选择题	27
2.4.2 判断题	27
2.4.3 填空题	27
2.4.4 简答题	27

2.4.5 算法设计题	28
第3章 栈和队列	30
3.1 重点与难点	30
3.1.1 基本概念	30
3.1.2 栈	30
3.1.3 队列	31
3.2 典型题分析	32
3.2.1 选择题	32
3.2.2 判断题	33
3.2.3 填空题	33
3.2.4 简答题	34
3.2.5 算法设计题	35
3.3 课后习题解答	37
3.3.1 选择题	37
3.3.2 判断题	38
3.3.3 简答题	38
3.3.4 算法设计题	39
3.4 单元练习题	43
3.4.1 选择题	43
3.4.2 判断题	44
3.4.3 填空题	44
3.4.4 简答题	44
3.4.5 算法设计题	45
第4章 字符串及线性结构的扩展	46
4.1 重点与难点	46
4.1.1 基本概念	46
4.1.2 字符串	47
4.1.3 数组	48
4.1.4 广义表	49
4.2 典型题分析	50
4.2.1 选择题	50
4.2.2 判断题	50
4.2.3 填空题	51

4.2.4 简答题	52	6.1.4 图的主要应用	101
4.2.5 算法设计题	54	6.2 典型题分析	103
4.3 课后习题解答	57	6.2.1 选择题	103
4.3.1 选择题	57	6.2.2 判断题	104
4.3.2 判断题	58	6.2.3 填空题	105
4.3.3 简答题	58	6.2.4 简答题	105
4.3.4 算法设计题	61	6.2.5 算法设计题	108
4.4 单元练习题	67	6.3 课后习题解答	112
4.4.1 选择题	67	6.3.1 选择题	112
4.4.2 判断题	67	6.3.2 判断题	113
4.4.3 填空题	67	6.3.3 简答题	114
4.4.4 简答题	68	6.3.4 算法设计题	118
4.4.5 算法设计题	69	6.4 单元练习题	125
第5章 树结构	70	6.4.1 选择题	125
5.1 重点与难点	70	6.4.2 判断题	125
5.1.1 基本概念	70	6.4.3 填空题	125
5.1.2 二叉树	71	6.4.4 简答题	125
5.1.3 哈夫曼树及哈夫曼编码	73	6.4.5 算法设计题	127
5.1.4 树	73	第7章 查找	128
5.1.5 树和森林的遍历	74	7.1 重点与难点	128
5.2 典型题分析	74	7.1.1 基本概念	128
5.2.1 选择题	74	7.1.2 线性表查找	128
5.2.2 判断题	76	7.1.3 树表查找	129
5.2.3 填空题	76	7.1.4 散列表	131
5.2.4 简答题	77	7.2 典型题分析	132
5.2.5 算法设计题	80	7.2.1 选择题	132
5.3 课后习题解答	84	7.2.2 判断题	133
5.3.1 选择题	84	7.2.3 填空题	134
5.3.2 判断题	85	7.2.4 简答题	134
5.3.3 简答题	86	7.2.5 算法设计题	136
5.3.4 算法设计题	90	7.3 课后习题解答	139
5.4 单元练习题	97	7.3.1 选择题	139
5.4.1 选择题	97	7.3.2 判断题	140
5.4.2 判断题	98	7.3.3 简答题	141
5.4.3 填空题	98	7.3.4 算法设计题	144
5.4.4 简答题	98	7.4 单元练习题	149
5.4.5 算法设计题	99	7.4.1 选择题	149
第6章 图结构	100	7.4.2 判断题	149
6.1 重点与难点	100	7.4.3 填空题	150
6.1.1 基本概念	100	7.4.4 简答题	150
6.1.2 图的存储结构	101	7.4.5 算法设计题	151
6.1.3 图的遍历	101	第8章 排序	152

8.1 重点与难点	152	8.4 单元练习题	171
8.1.1 基本概念	152	8.4.1 选择题	171
8.1.2 插入排序	152	8.4.2 判断题	171
8.1.3 交换排序	153	8.4.3 填空题	172
8.1.4 选择排序	154	8.4.4 简答题	172
8.1.5 归并排序	155	8.4.5 算法设计题	173
8.1.6 链式基数排序	155	第9章 模拟试题	174
8.2 典型题分析	155	模拟试题一(专科水平)	174
8.2.1 选择题	155	模拟试题二(专科水平)	176
8.2.2 判断题	156	模拟试题三(专科水平)	178
8.2.3 填空题	157	模拟试题四(本科水平)	180
8.2.4 简答题	157	模拟试题五(本科水平)	182
8.2.5 算法设计题	159	模拟试题六(本科水平)	184
8.3 课后习题解答	162	模拟试题七(研究生入学考试	
8.3.1 选择题	162	水平)	186
8.3.2 判断题	163	模拟试题八(研究生入学考试	
8.3.3 简答题	163	水平)	188
8.3.4 算法设计题	167		

第2篇 实验指导与课程设计

第10章 实验题目	190	10.7 排序的应用	207
10.1 线性表的应用	190	第11章 实验报告规范	208
10.1.1 约瑟夫环问题	190	11.1 数据结构实验概述	208
10.1.2 一元多项式运算	191	11.2 实验步骤	209
10.2 栈和队列的应用	193	11.3 实验报告规范及样例	210
10.2.1 模拟停车场管理	193	11.3.1 实验报告规范	210
10.2.2 算术表达式求值	195	11.3.2 实验报告样例	210
10.3 串和数组的应用	196	附录 模拟试题参考答案	217
10.3.1 格式化文本	196	模拟试题一	217
10.3.2 稀疏矩阵的运算	198	模拟试题二	219
10.4 树及其应用	199	模拟试题三	221
10.4.1 哈夫曼编码/译码器	199	模拟试题四	223
10.4.2 二叉树遍历	201	模拟试题五	226
10.5 图及其应用	202	模拟试题六	229
10.5.1 校园导游程序	202	模拟试题七	232
10.5.2 教学计划编排	203	模拟试题八	235
10.6 查找的应用	205	参考文献	239
10.6.1 职工信息检索系统	205		
10.6.2 个人图书管理系统	206		

第1篇 学习指导与习题解答

第1章 绪 论

1.1 重点与难点

1.1.1 基本概念

1. 数据

①数据。数据是信息的载体，是所有能够被计算机识别、存储和加工处理的符号的总称。它是计算机程序加工的原料，应用程序可以处理各种各样的数据。

②数据项。数据项是具有独立含义的标识单位，是数据不可分割的最小单位。

③数据元素。数据元素是数据的基本单位。在不同的条件下，数据元素又可称为元素、结点、顶点、记录等。

2. 数据结构

数据结构是指互相之间存在着一种或多种关系的数据元素的集合。

数据结构涉及数据元素之间的逻辑关系、数据在计算机中的存储方式和这些数据上定义的一组运算，一般称这3个方面为数据的逻辑结构、数据的存储结构和数据的运算。

3. 逻辑结构

数据的逻辑结构可以被看做从具体问题抽象出来的数学模型，它与数据的存储无关。数据的逻辑结构分为以下4类：

①集合。集合中元素间的关系极为松散，数据元素间的关系是“属于同一个集合”。

②线性结构。该结构的数据元素之间存在着一对一的关系。

③树结构。该结构的数据元素之间存在着一对多的关系。

④图结构。该结构的数据元素之间存在多对多的关系。

4. 存储结构

数据结构在计算机中的映像称为数据的物理结构，又称存储结构。它所研究的是数据的逻辑结构在计算机中的实现方法，包括数据结构中元素的表示及元素间关系的表示。数据的存储结构分为以下四类：

①顺序存储方法。把逻辑上相邻的元素存储在物理位置相邻的存储单元中，结点间的逻辑关系由存储单元的邻接关系来体现。

②链式存储方法。对逻辑上相邻的元素不要求其物理位置相邻，元素间的逻辑关系通过

附设的指针字段来表示。

③索引存储方法。在存储结点信息的同时，还建立附加的索引表。索引存储主要是针对数据内容的存储，而不强调关系的存储，索引存储方法主要面向查找操作。

④散列存储方法。以数据元素的关键字的值为自变量，通过某个函数（散列函数）计算出该元素的存储位置。

5. 运算

运算是对于数据的处理。运算与逻辑结构紧密相连，每种逻辑结构都有一个运算的集合。数据的运算是定义在数据的逻辑结构上的，但运算的具体实现是在数据的存储结构上进行的。运算分为两种类型：

①引用型运算。这类运算不改变数据原有的逻辑结构的值，只根据需要读取某些信息。

②加工型运算。这类运算的结果会改变数据原有的逻辑结构的值。

6. 数据类型

数据类型是一个值的集合和定义在这个值集上一组操作的总称。数据类型可分为两类：

①原子类型。其值是不可分解的。

②结构类型。其值是由若干成分按某种结构组成的，是可分解的，并且它的成分可以是非结构的或结构的。

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。抽象数据类型的定义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。抽象数据类型的特征是使用与实现相分离，实行封装和信息隐蔽。

1.1.2 算法及其分析

1. 算法

算法是对特定问题求解步骤的一种描述，是指令的有限序列。其中每一条指令表示一个或多个操作。

一个算法应该具有下列特性：有穷性、确定性、可行性、输入、输出。

设计一个好的算法要考虑以下的要求：正确、可读、健壮、高效。

2. 算法的分析

评价一个算法的优劣通常从算法的计算时间与所需存储空间来讨论。

1) 算法的时间复杂度就是指算法的时间耗费，用 $T(n)$ 表示。

一个算法执行所耗费的时间，是算法中所有语句执行时间之和，而每条语句的执行时间是该语句执行一次所用时间与该语句重复执行次数（语句的频度）的乘积。

一个算法的时间复杂度 $T(n)$ 表示为

$$T(n) = \sum_{\text{语句}} (t_i \times c_i)$$

其中， t_i 表示语句 i 执行一次的时间； c_i 表示语句 i 的频度。

在实际应用中，常采用一些简单的函数来近似表示时间性能，即渐进时间复杂度。

定义（大 O 记号）：设 $T(n)$ 是问题规模 n 的函数 $f(n)$ ，若存在两个正常数 c 和 n_0 ，使得对所有的 n ， $n \geq n_0$ ，有 $T(n) \leq cf(n)$ ，则记为 $T(n) = O(f(n))$ 。

使用大 O 记号表示的算法的时间复杂度，称为算法的渐进时间复杂度，简称时间复杂度。

2) 算法的空间复杂度是指程序运行从开始到结束所需的存储量。

程序运行所需的存储空间包括以下两部分:

①固定部分。这部分空间与所处理数据的大小和个数无关,或者称与问题的实例的特征无关。

②可变部分。这部分空间大小与算法在某次执行中处理的特定数据的大小和规模有关。

当问题规模较大时,可变部分可能会远大于固定部分,一般讨论算法的渐进空间复杂度。

1.2 典型题分析

1.2.1 选择题

1. 数据结构是一门研究非数值计算程序设计问题中计算机的____①____以及它们之间的____②____等的学科。

- ①A. 操作对象 B. 计算方法 C. 逻辑存储 D. 数据映像
②A. 结构和操作 B. 关系和运算 C. 运算和操作 D. 算法和操作

【分析】数据结构是一门研究非数值计算程序设计中计算机的操作对象以及它们之间的关系和运算等的学科。

【答案】①A ②B

2. 算法是指____①____,它必须具备的3个特性是____②____。

- ①A. 计算方法 B. 解决问题的步骤序列 C. 排序方法 D. 调度方法
②A. 可执行性、可移植性、可扩充性 B. 易读性、稳定性、安全性
C. 确定性、有穷性、稳定性 D. 有穷性、确定性、可行性

【分析】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;算法具有五个重要特性:有穷性、确定性、可行性、输入、输出。

【答案】①B ②D

```
3. 程序段 for (i=1;i<n;i++)  
        for (j=1;j<i;j++)  
            if (a[j]>a[j+1])  
                {t=a[j];  
                  a[j]=a[j+1];  
                  a[j+1]=t;  
                }
```

其中, n 为正整数,则最后一行的语句频度在最坏情况下是_____。

- A. $O(n)$ B. $O(\log n)$ C. $O(n^2)$ D. $O(n^3)$

【分析】在最坏情况下,每次循环最后一行语句都会执行一次,一共执行了

$(n-1)+(n-2)+(n-3)+\cdots+1=n(n-1)/2$ 次,故选 C。

【答案】C

1.2.2 判断题

1. 顺序存储方式只能用于线性结构,不能用于非线性结构。

【分析】顺序存储方式也能用于存储非线性结构。例如，对于树结构，其中完全二叉树和堆就是用数组存储实现的。

【答案】错误

2. 数据元素是数据的最小单位。

【分析】数据元素是数据的基本单位，而数据元素又可以由数据项组成，所以数据项才是数据的最小单位。

【答案】错误

3. 在相同的规模 n 下，复杂度为 $O(n)$ 的算法在时间上总是优于复杂度为 $O(2^n)$ 的算法。

【分析】在相同的规模 n 下，函数 $f(n) = n$ 的值总小于函数 $f(n) = 2^n$ 的值。

【答案】正确

4. 程序一定是算法。

【分析】算法和程序的一个重要区别在于，程序不一定是有穷的，而算法必须是有穷的。也就是说，一个算法必须具有有穷性，而程序则不一定。

【答案】正确

5. 数据的存储结构是指数据在计算机内的实际存储形式。

【分析】数据的存储结构是数据元素及其关系在计算机内的存储表示。

【答案】正确

1.2.3 填空题

1. 抽象数据类型的定义仅取决于它的一组 ① 而与 ② 无关。

【答案】①逻辑特性 ②其在计算机内部如何表示和实现

2. 数据结构中评价算法的两个性能指标是 ① 和 ② 。

【答案】①时间复杂度 ②空间复杂度

3. 线性结构中元素的关系是 ①，树结构中元素的关系是 ②，图结构中元素的关系是 ③。

【答案】①一对一 ②一对多 ③多对多

4. 下面程序段的时间复杂度为_____。

```
for (i=0; i<m; i++)
    for (j=0; j<n; j++)
        a[i][j] = i * j;
```

【分析】第1个for循环执行 m 次，第2个for循环执行 n 次，两个for循环嵌套起来共执行 $m \times n$ 次，故时间复杂度为 $O(m \times n)$ 。

【答案】 $O(m \times n)$

5. 计算机执行下面的语句时，语句 s 的执行次数为_____。

```
for (i=1; i<n-1; i++)
    for (j=n; j>=i; j--)
        s;
```

【分析】外层for循环执行 $n-2$ 次，内层for循环分别执行 $n, n-1, n-2, \dots, 3$ 次，两个for循环嵌套起来共执行 $n + (n-1) + (n-2) + \dots + 3 = (n+3)(n-2)/2$ 次。

【答案】 $(n+3)(n-2)/2$

1.3 课后习题解答

1.3.1 选择题

1. 从逻辑上可以把数据结构分为 (C) 两大类。
A. 动态结构、静态结构 B. 顺序结构、链式结构
C. 线性结构、非线性结构 D. 初等结构、构造型结构
2. 在下面的程序段中, 对 x 的赋值语句的频度为 (C)。

```
for(k=1;k<=n;k++)  
    for(j=1;j<=n;j++)  
        x=x+1;
```


A. $O(2n)$ B. $O(n)$ C. $O(n^2)$ D. $O(\log_2 n)$
3. 采用链式存储结构表示数据时, 相邻的数据元素的存储地址 (C)。
A. 一定连续 B. 一定不连续
C. 不一定连续 D. 部分连续, 部分不连续
4. 下面关于算法说法正确的是 (D)。
A. 算法的时间复杂度一般与算法的空间复杂度成正比
B. 解决某问题的算法可能有多种, 但肯定采用相同的数据结构
C. 算法的可行性是指算法的指令不能有二义性
D. 同一个算法, 实现语言的级别越高, 执行效率就越低
5. 在发生非法操作时, 算法能够作出适当处理的特性称为 (B)。
A. 正确性 B. 健壮性 C. 可读性 D. 可移植性

1.3.2 判断题

1. 数据的逻辑结构是指数据的各数据项之间的逻辑关系。(✓)
2. 顺序存储方式的优点是存储密度大, 且插入、删除运算效率高。(×)
3. 数据的逻辑结构说明数据元素之间的次序关系, 它依赖于数据的存储结构。(×)
4. 算法的优劣与描述算法的语言无关, 但与所用计算机的性能有关。(×)
5. 算法必须有输出, 但可以没有输入。(✓)

1.3.3 简答题

1. 常见的逻辑结构有哪几种, 各自的特点是什么? 常用的存储结构有哪几种, 各自的特点是什么?

【解答】常见的四种逻辑结构有:

- ①集合结构: 数据元素间的关系是“属于同一个集合”。
- ②线性结构: 数据元素之间存在着一对一的关系。
- ③树结构: 数据元素之间存在着一对多的关系。

④图结构：数据元素之间存在着多对多的关系。

常见的四种存储结构有：

①顺序存储：把逻辑上相邻的元素存储在物理位置相邻的存储单元中。顺序存储结构是一种最基本的存储表示方法，通常借助于程序设计语言中的数组来实现。

②链接存储：对逻辑上相邻的元素不要求物理位置相邻的存储单元，元素间的逻辑关系通过附设的指针域来表示。

③索引存储：通过建立索引表存储结点信息的方法，其中索引表一般存储结点关键字和一个地点信息，可通过该地址找到结点的其他信息。

④散列存储：根据结点的关键字直接计算出该结点的存储地址的方法。

2. 简述算法和程序的区别。

【解答】一个算法若用程序设计语言来描述，则它就是一个程序。算法的含义与程序十分相似，但又有区别。一个程序不一定满足有穷性。例如，操作系统，只要整个系统不遭破坏，它将永远不会停止，即使没有作业需要处理，它仍处于动态等待中。因此，操作系统不是一个算法。另一方面，程序中的指令必须是机器可执行的，而算法中的指令则无此限制。算法代表了对问题的解，而程序则是算法在计算机上的特定的实现。

3. 试举一个数据结构的例子，叙述其逻辑结构、存储结构、运算这3方面的内容。

【解答】略。

4. 运算是数据结构的一个重要方面。试举例说明两个数据结构的逻辑结构和存储方式完全相同，只是对于运算的定义不同，使得两个结构具有显著不同的特性。

【解答】比如顺序栈和循环队列，二者的逻辑结构都是线性结构，都采用顺序存储方式存储，但它们的运算不同，栈限定元素的插入和删除在栈顶进行，队列限定元素在队尾插入、在队首删除，因此它们是截然不同的数据结构。

5. 分析下列程序段中带标号“#”语句的执行频度（ n 为正整数）。

```
(1) j = 1; k = 0;
    while(j <= n - 1)
    { j++;
      k += j; /* # */
    }
```

【解答】 $n - 1$

```
(2) i = 0; s = 0; n = 100;
    do { i++;
        s += 10 * i; /* # */
      } while(! (i < n && s < n));
```

【解答】1

```
(3) k = 0;
    for(i = 0; i < n; i++)
      for(j = i; j < n; j++)
        k++; /* # */
```

【解答】 $n(n + 1)/2$

```
(4) a = 1; b = 0;
```

```

while(a + b <= n)
{ if(a < b)  a++;  /* **/
  else  b++;
}

```

【解答】n

```

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

```

【解答】1000

6. 写出下列各程序段关于 n 的时间复杂度。

```

(1) a=1; m=1;
while (a < n)
{ m += a;
  a *= 3;
}

```

【解答】 $O(n)$

(2) 设 n 是偶数。

```

for(i=1, s=0; i <= n; i++)
  for(j=2*i; j <= n; j++)
    s++;

```

【解答】 $O(n^2)$

```

(3) for(i=1; i <= n-1; i++)
{ k=i;
  for(j=i+1; j <= n; j++)
    if(R[j] > R[j+1]) k=j;
  t=R[k]; R[k]=R[i]; R[i]=t;
}

```

【解答】 $O(n^2)$

7. 计算一元 n 次多项式 $P(x, n) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$ 的值，输入 x, n, $a_0, a_1, \cdots, a_n$ ，输出多项式 $P(x, n)$ 的值。设计算法求解，请选择合适的输入、输出格式，要求算法具有较好的时间性能。

【解答】

①将一元 n 次多项式做如下改写：

$$\begin{aligned}
 P(x, n) &= a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \\
 &= a_0 + x(a_1 + a_2x + \cdots + a_nx^{n-1}) \\
 &= \cdots \\
 &= a_0 + x(a_1 + x(a_2 + x(a_3 + \cdots + x(a_{n-1} + a_nx^{n-1}) \cdots)))
 \end{aligned}$$

②按指数递减次序输入各系数，即输入次序为 $a_n, a_{n-1}, \cdots, a_2, a_1, a_0$

③算法如下：

```

main()
{
    s = 0;
    scanf("%f", &x);
    for(k = n; k >= 1; k--)
        {
            scanf("%f", &a);
            s = s * x + a;
        }
    printf("P(x,n) = %f", s);
}

```

④分析：该算法仅使用了 n 次乘法和 n 次加法，来完成一元 n 次多项式的求值问题，具有较好的时间和空间性能。

1.4 单元练习题

1. 若有 100 个学生，每个学生要求具有学号、姓名、平均成绩 3 个数据项。请问，采用什么样的数据结构最方便？写出这些结构。

2. 试举一例，说明对相同的逻辑结构，同一种运算在不同的存储方式下实现，其运算效率不同。

3. 设计一个数据结构，用来表示某一银行储户的基本信息：账号、姓名、开户年月日、储蓄类型、存入累加数、利息、账面总数。

4. 分析下面程序段中循环语句的执行次数。

```

i = 0;
s = 0;
n = 100;
while((i >= n) || (sum >= n))
{
    i = i + 1;
    sum = sum + 10 * i;
}

```

5. 调用函数 $f(n)$ ，回答下列问题：

(1) 试指出 $f(n)$ 值的大小，并写出 $f(n)$ 值的推导过程；

(2) 假定 $n = 5$ ，试指出 $f(5)$ 值的大小和执行 $f(5)$ 时的输出结果。

```

int f(int n)
{
    int i, j, k, sum = 0;
    for(i = 1; i <= n; i++)
        {
            for(j = n; j > i - 1; j--)
                for(k = 1; k <= j; k++)
                    sum++;
            printf("sum = %d\n", sum);
        }
    return (sum);
}

```

第2章 线性表

2.1 重点与难点

2.1.1 基本概念

1. 线性表

线性表是具有相同数据类型的 $n(n \geq 0)$ 个数据元素的有限序列，通常记为 $(a_1, a_2, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n)$ ，其中 n 为表长， $n=0$ 时称为空表。表中相邻元素之间存在着次序关系。将 a_{i-1} 称为 a_i 的直接前驱， a_{i+1} 称为 a_i 的直接后继。

线性表的基本运算有线性表初始化、求线性表的长度、取表元、按值查找、插入操作、删除操作。

2. 顺序表

顺序表指采用顺序存储结构存放的线性表。存储的方式为在内存中用地址连续的一块存储空间顺次存放线性表的各元素。

3. 单链表

单链表指采用链式存储结构存放的线性表。单链表通过一组任意的存储单元来存储线性表中的数据元素的。为建立起数据元素之间的线性关系，对于每个数据元素，除了存放数据元素自身的信息外，还需要存放其后继元素的地址。

4. 头指针

头指针是一个指针变量，里面存放的是链表首结点的地址，并以此来标识一个链表。

如链表 H 、链表 L 等，表示链表中第 1 个结点的地址存放在 H 、 L 中。通常用头指针来唯一标识一个链表。

5. 头结点

头结点指附加在第 1 个元素结点之前的一个结点，头指针指向头结点。当该链表表示一个非空的线性表时，头结点的指针域指向第一个元素结点；为空表时，该指针域为空。

头结点具有以下作用：

- ①使对空表和非空表的处理得到统一。
- ②在链表的第 1 个位置上的操作和在其他位置上的操作一致，无须特殊处理。

2.1.2 顺序表及运算

1. 顺序表

线性表的顺序存储是指在内存中用地址连续的一块存储空间顺次存放线性表的各元素，用这种存储形式存储的线性表称为顺序表。内存中的地址空间是线性的，顺序表就是用物理位置的相邻来表现数据元素之间的逻辑关系的相邻。