

二级 公共基础知识

—— | 计算机等级考试命题研究组 组编 | ——

特色模块
考点分析 · 经典题解 · 标准模拟

· 突出标准性与严谨性 · 突出实用性和高效性 · 注重典型考题的分析
· 注重上机考试的辅导



机械工业出版社
CHINA MACHINE PRESS

全国计算机等级考试考点考题解析与实战

二级公共基础知识

计算机等级考试命题研究组 组编

主 编 史国川

副主编 张孟资 谢宝陵

参 编 周 生 邵文军 徐国明 等



机械工业出版社

本书严格按照教育部考试中心制定的《全国计算机等级考试考试大纲（2004 年版）》编写，章节安排与教育部考试中心主编的《全国计算机等级考试二级教程——公共基础知识（2004 年版）》同步，每节细化为 4 个板块：考点分析、经典题解、即学即练、即学即练参考答案。实践表明这种“一点一练”的结构体例更便于记忆与理解。

本书考点全面、考题典型、练习丰富，非常适合有关考生使用，也可作为高等院校或培训班的教材。

图书在版编目（CIP）数据

二级公共基础知识 / 计算机等级考试命题研究组组编. —北京：机械工业出版社，2006.1

（全国计算机等级考试考点考题解析与实战）

ISBN 7-111-17905-6

I. 二... II. 计... III. 电子计算机—水平考试—自学参考资料 IV. TP3

中国版本图书馆 CIP 数据核字（2005）第 134853 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

策 划：胡毓坚

责任编辑：孙 业

责任印制：洪汉军

北京双青印刷厂 印刷

2006 年 1 月第 1 版·第 1 次印刷

787mm×1092mm 1/16·9.25 印张·225 千字

0001—5000 册

定价：15.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

本社购书热线电话（010）68326294

封面无防伪标均为盗版

前 言

为了适应信息技术的不断发展和社会需求的新变化，经过专家充分论证，教育部考试中心于 2004 年对全国计算机等级考试的科目设置、考核内容和考试形式进行了一定的调整。配合教育部考试中心新的考试要求，我们深入研究了教育部考试中心相关资料、历年真题，按照考试中心确定的考试范围和考试重点编写了本书，目的是为了广大考生提供一本标准、严谨、实用、高效的考试用书，以便读者有针对性地复习过关。

本书具有以下特点：

☑ 突出标准性与严谨性

本书严格按照教育部考试中心制定的《全国计算机等级考试考试大纲（2004 年版）》编写，覆盖所有考核要点，内容全面，层次清晰，结构严谨。

☑ 突出实用性和高效性

本书的章节安排与教育部考试中心主编的最新指定教程同步，每节细化为 4 个板块：考点分析、经典题解、即学即练、即学即练参考答案。这种“一点一练”的结构体例更便于记忆与理解。

☑ 注重典型考题的分析

本书以典型试题的分析贯穿考点，深度总结考试命题规律与解题技巧，便于考生举一反三、触类旁通。

本书考点全面、考题典型、练习丰富，非常适合有关考生使用，也可作为高等院校或培训班的教材。

参与本书编写的人员还有：李千目、彭希珩、侯君、戚湧、王立新、尹静、杨章静、王燕舞、吴婷、陈玉旺、吴晓维、李文龙、周红、俞永达。

书中不妥之处敬请广大读者批评指正。

计算机等级考试命题研究组

目 录

前言

第1章 基本数据结构与算法..... 1

1.1 算法 1

1.1.1 考点分析..... 1

1.1.2 经典题解..... 3

1.1.3 即学即练..... 3

1.1.4 即学即练答案 4

1.2 数据结构基本概念 4

1.2.1 考点分析..... 4

1.2.2 经典题解..... 6

1.2.3 即学即练..... 7

1.2.4 即学即练答案 7

1.3 线性表 7

1.3.1 考点分析..... 7

1.3.2 经典题解.....10

1.3.3 即学即练.....10

1.3.4 即学即练答案11

1.4 栈和队列11

1.4.1 考点分析.....11

1.4.2 经典题解.....15

1.4.3 即学即练.....16

1.4.4 即学即练答案17

1.5 线性链表18

1.5.1 考点分析.....18

1.5.2 经典题解.....21

1.5.3 即学即练.....23

1.5.4 即学即练答案24

1.6 树与二叉树24

1.6.1 考点分析.....24

1.6.2 经典题解.....28

1.6.3 即学即练.....30

1.6.4 即学即练答案31

1.7 基本查找技术31

1.7.1 考点分析.....31

1.7.2 经典题解.....32

1.7.3 即学即练.....34

1.7.4 即学即练答案34

1.8 基本排序技术35

1.8.1 考点分析.....35

1.8.2 经典题解.....40

1.8.3 即学即练.....41

1.8.4 即学即练答案42

第2章 程序设计基础.....43

2.1 程序设计方法与风格43

2.1.1 考点分析.....43

2.1.2 经典题解.....44

2.2 结构化程序设计45

2.2.1 考点分析.....45

2.2.2 经典题解.....46

2.2.3 即学即练.....47

2.2.4 即学即练答案47

2.3 面向对象的程序设计47

2.3.1 考点分析.....47

2.3.2 经典题解.....51

2.3.3 即学即练.....52

2.3.4 即学即练答案52

第3章 软件工程53

3.1 软件工程基本概念53

3.1.1 考点分析.....53

3.1.2 经典题解.....56

3.1.3 即学即练.....56

3.1.4 即学即练答案57

3.2 结构化分析方法57

3.2.1 考点分析.....57

3.2.2 经典题解.....60

3.2.3 即学即练.....61

3.2.4 即学即练答案62

3.3 结构化设计方法62



3.3.1 考点分析·····	62	4.4.3 即学即练·····	109
3.3.2 经典题解·····	68	4.4.4 即学即练答案·····	110
3.3.3 即学即练·····	69	第5章 标准模拟试卷及答案分析·····	111
3.3.4 即学即练答案·····	70	5.1 标准模拟试卷·····	111
3.4 软件的测试·····	70	5.1.1 标准模拟试卷一·····	111
3.4.1 考点分析·····	70	5.1.2 标准模拟试卷二·····	112
3.4.2 经典题解·····	74	5.1.3 标准模拟试卷三·····	113
3.4.3 即学即练·····	75	5.1.4 标准模拟试卷四·····	114
3.4.4 即学即练答案·····	75	5.1.5 标准模拟试卷五·····	115
3.5 程序的调试·····	75	5.1.6 标准模拟试卷六·····	117
3.5.1 考点分析·····	75	5.1.7 标准模拟试卷七·····	118
3.5.2 经典题解·····	78	5.1.8 标准模拟试卷八·····	119
3.5.3 即学即练·····	78	5.1.9 标准模拟试卷九·····	120
3.5.4 即学即练答案·····	78	5.1.10 标准模拟试卷十·····	122
第4章 数据库设计基础·····	79	5.1.11 标准模拟试卷十一·····	123
4.1 数据库系统的基本概念·····	79	5.1.12 标准模拟试卷十二·····	124
4.1.1 考点分析·····	79	5.2 标准模拟试卷答案分析·····	125
4.1.2 经典题解·····	82	5.2.1 标准模拟试卷一答案分析·····	125
4.1.3 即学即练·····	83	5.2.2 标准模拟试卷二答案分析·····	127
4.1.4 即学即练答案·····	84	5.2.3 标准模拟试卷三答案分析·····	128
4.2 数据模型·····	84	5.2.4 标准模拟试卷四答案分析·····	129
4.2.1 考点分析·····	84	5.2.5 标准模拟试卷五答案分析·····	131
4.2.2 经典题解·····	89	5.2.6 标准模拟试卷六答案分析·····	132
4.2.3 即学即练·····	90	5.2.7 标准模拟试卷七答案分析·····	133
4.2.4 即学即练答案·····	91	5.2.8 标准模拟试卷八答案分析·····	134
4.3 关系代数·····	92	5.2.9 标准模拟试卷九答案分析·····	135
4.3.1 考点分析·····	92	5.2.10 标准模拟试卷十答案分析·····	137
4.3.2 经典题解·····	102	5.2.11 标准模拟试卷十一答案 分析·····	138
4.3.3 即学即练·····	102	5.2.12 标准模拟试卷十二答案 分析·····	139
4.3.4 即学即练答案·····	103	参考文献·····	141
4.4 数据库设计与管理·····	103		
4.4.1 考点分析·····	103		
4.4.2 经典题解·····	109		

第1章 基本数据结构与算法

本章大纲要求:

- ☑ 算法的基本概念; 算法复杂度的概念和意义(时间复杂度和空间复杂度)。
- ☑ 数据结构的定义; 数据的逻辑结构和存储结构; 数据结构的图形表示; 线性结构与非线性结构的概念。
- ☑ 线性表的定义; 线性表的顺序存储结构及其插入删除运算。
- ☑ 栈和队列的定义及其顺序存储结构、基本运算。
- ☑ 线性单链表、双向链表和循环链表的结构及其基本运算。
- ☑ 树的基本概念; 二叉树的定义及其存储结构; 二叉树的前序、中序和后序遍历。
- ☑ 顺序查找与二分法查找算法。
- ☑ 基本排序算法(插入类排序、交换类排序、选择类排序)。

1.1 算法

1.1.1 考点分析

考点 1: 算法的基本概念

算法(Algorithm)是指为解决某个特定问题而采取的确定且有限的步骤的一种描述,它是指令的有限序列,使得给定类型的问题通过有限的指令序列、在有限的时间内被求解。其中每一条指令表示一个或多个操作。

1. 算法的基本特性

- 1) 有穷性: 一个算法应包含有限个操作步骤,而且每一步都应在合理的时间内完成。
- 2) 确定性: 算法中每一条指令必须有确切的含义,不能有二义性,对于相同的输入必能得出相同的执行结果。
- 3) 可行性: 算法中指定的操作都可以通过基本运算执行有限次后实现。
- 4) 拥有足够的情报: 一个算法的执行结果总是与输入的初始数据有关,它有零个或多个输入而有一个或多个输出,不同的输入将会有不同的输出结果。通常情况下,当情报足够时,算法才是有效的,而当提供的情报不够时,算法可能无效。

2. 算法的基本要素

一个算法通常由两种基本要素构成:一是对数据对象的运算和操作;二是算法的控制结构。

1) 算法中对数据的运算和操作:对于所有的算法都是按照要求从环境能够运行的所有操作中选择合适的操作所组成的一组指令序列。

通常,在计算机系统中基本的运算和操作包括四类:

- ① 算术运算:主要包括加、减、乘、除等运算。



- ② 逻辑运算：主要包括“与”、“或”、“非”等运算。
- ③ 关系运算：主要包括“大于”、“小于”、“等于”和“不等于”等运算。
- ④ 数据传输：主要包括赋值、输入和输出等操作。

2) 算法的控制结构：算法中各操作之间的执行顺序称为算法的控制结构。算法的功能不仅取决于所选择的操作，还与各操作的执行顺序有关。

一个算法的控制结构不仅决定了算法中各操作之间的执行顺序，更直接反映了算法的设计是否符合结构化设计原则，也就是说，它给出了算法的基本框架。算法通常可以用顺序、选择和循环三种基本控制结构组合而成，而算法的描述工具通常有传统流程图、N-S 结构化流程图、伪码描述语言等。

3. 算法设计的基本方法

1) 列举法：它是计算机算法中的一个基础算法，此方法是根据实际问题，列举出所有可能出现的情况，并用问题中给定的条件检验哪些是需要的，哪些是不需要的。

2) 归纳法：通过列举少量的简单而又特殊的情况，经过分析，总结归纳出一般性的结论。

3) 递推：从给定的或已知的初始条件出发，逐步推出所要求的各中间结果和最后结果。递推关系通常是归纳的结果，此方法经常用在数值计算中。

4) 递归：在解决复杂问题或问题的规模比较大时，将问题逐层分解，最后归纳为一些简单的问题，当解决了最后那些简单的问题后，再沿原来的逆过程逐步进行综合，这就是递归基本思想。

5) 减半递推技术：“减半”，是指将问题的规模减半，而问题的性质不变；“递推”，是指重复减半的过程。这是工程上常用的分治法的一种，用来解决复杂程度与规模有密切联系的实际问题。

6) 回溯法：在有些实际问题中有时很难归纳出简单的公式或直观的求解步骤，也不能无限制的列举，一种有效的解决此类问题的方法是尝试。通过分析问题，找出一个解决的线索，沿此线索逐步试探，若成功，就得到问题的解，否则，就逐步回退，换线索再逐步试探。

4. 算法设计的要求

- 1) 正确性：算法的执行结果应当满足预先规定的功能和性能要求。
- 2) 可读性：一个算法应当思路清晰、层次分明、简单明了、易读易懂。
- 3) 健壮性：当输入不合法数据时，应能作适当处理，不至引起严重后果。
- 4) 高效性：有效使用存储空间和有较高的时间效率。

考点 2：算法复杂度

1. 时间复杂度

一个算法的时间复杂度 (Time Complexity) 是指算法运行从开始到结束所需要的计算工作量。

一个算法是由控制结构和原操作构成的，其执行时间取决于两者的综合效果。为了便于比较同一问题的不同的算法，通常的做法是：从算法中选取一种对于所研究的问题来说是基本运算的原操作，以该原操作重复执行的次数作为算法的时间度量。一般情况下，算法中原操作重复执行的次数是问题规模 n 的某个函数 $f(n)$ ，即

$$\text{算法的工作量} = f(n)$$

通常记作：



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

它表示随问题规模 n 的增大, 算法执行时间的增长率和 $f(n)$ 的增长率相同, 称为算法的时间复杂度。例如, 一个程序的实际执行时间为 $T(n)=2.7n^3+3.8n^2+5.3$, 则 $T(n)=O(n^3)$ 。通常用 $O(1)$ 表示常数计算时间, 常见的时间复杂度有:

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

在实际的问题中, 算法的时间复杂度不仅与问题规模有关, 而且还与特定的输入, 也就是初始数据状态有关。

2. 空间复杂度

一个算法的空间复杂度 (Space Complexity) 是指算法运行从开始到结束所需的存储空间量。算法执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规模无关, 主要包括程序代码、常量、简单变量、定长成分的结构变量所占的空间。

2) 可变部分: 这部分空间大小与算法在某次执行中处理的数据的大小和规模有关, 也就是算法执行过程中所需要的额外空间。

在许多实际的问题中, 为了减少算法的存储空间, 一般采取压缩存储技术, 以便尽量减少不必要的额外空间。

1.1.2 经典题解

一、选择题

【例 1】计算机算法指的是_____。

- A) 计算方法
- B) 调度方法
- C) 排序方法
- D) 解决某一问题的有限运算序列

答案: D。解析: 根据算法的定义, 本题应选 D。

【例 2】算法的时间复杂度取决于_____。

- A) 问题的规模
- B) 待处理的数据的初始状态
- C) 问题的困难度
- D) A) 和 B)

答案: D。解析: 算法的复杂度不仅与问题的规模有关, 而且与输入数据有关, 即输入数据所有的可能取值范围以及输入各种数据或数据集的概率有关, 而与问题的难度无关。

二、填空题

【例 1】描述算法的常用方法有_____。

答案: 传统流程图、N-S 结构化流程图和伪码描述语言

【例 2】一个算法的时间复杂度是_____的函数。

答案: 算法输入规模。解析: 一般情况下, 一个算法的复杂度是算法输入规模的函数。一个算法的输入规模或问题规模是指作为该算法输入的数据所含数据元素的数目, 或与此数目有关的其他参数。

1.1.3 即学即练

1. 算法具有四个特性, 以下选项中不属于算法特性的是_____。

- A) 有穷性
- B) 简洁性
- C) 可行性
- D) 确定性



2. 算法的计算量大小称为算法的_____。

- A) 概率 B) 难度 C) 复杂度 D) 效率

1.1.4 即学即练答案

1. B 2. C

1.2 数据结构基本概念

1.2.1 考点分析

考点 1: 数据结构的基本概念

1. 数据结构的定义

数据结构 (Data Structure) 是指互相之间存在着一种或多种关系的数据元素的集合。数据结构作为计算机的一门学科, 主要研究和讨论以下三个方面的问题:

- 1) 数据集合中各数据元素之间所固有的逻辑关系, 即数据的逻辑结构。
- 2) 各数据元素在计算机中的存储关系, 即数据的存储结构。
- 3) 对各种数据结构进行的运算。

研究以上问题的主要目的是为了提高数据处理的效率, 主要体现在两个方面:

- 1) 提高数据处理的速度。
- 2) 尽量节省数据处理过程中所占用的计算机存储空间。

要理解数据结构, 还需要理解以下基本概念:

数据 (Data) 是信息的载体, 它能够被计算机识别、存储和加工处理。计算机科学中, 所谓数据就是计算机加工处理的对象, 它可以是数值数据, 也可以是非数值数据。

数据元素 (Data Element) 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。

数据对象 (Data Object) 是具有相同性质的数据元素的集合。在某个具体问题中, 数据元素都具有相同的性质 (元素值不一定相等), 属于同一数据对象, 数据元素是数据元素类的一个实例。

2. 数据的逻辑结构

在任何问题中, 数据元素之间都不会是孤立的, 在它们之间都存在着这样或那样的关系, 这种数据元素之间的关系称为结构。一个数据结构应包含两个方面的信息: 一是表示数据元素的信息; 二是表示各数据元素之间的前后件关系。

其中数据元素之间的前后件关系是指它们之间的逻辑关系, 而上述的数据结构实际上是数据的逻辑结构, 它是对数据元素之间的逻辑关系的描述。根据数据元素之间关系的不同特性, 通常有下列四类基本的逻辑结构:

- 1) 集合结构: 在集合结构中, 数据元素间的关系是“属于同一个集合”, 集合是元素关系极为松散的一种结构。
- 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) 树形结构: 数据元素之间存在着一对多的关系。

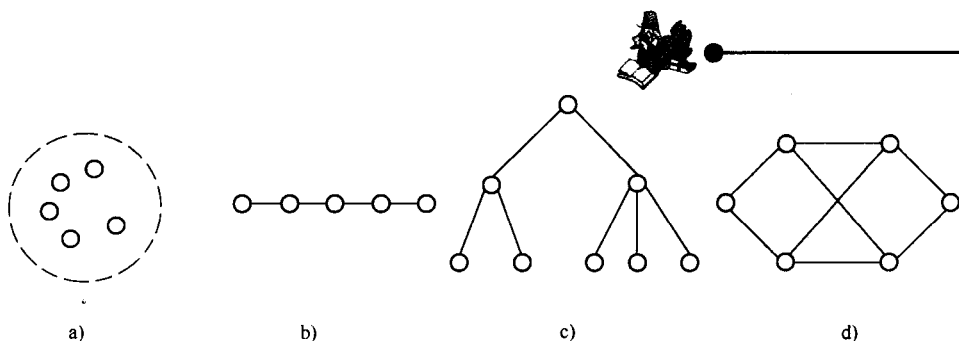


图 1-1 数据的逻辑结构

a) 集合结构 b) 线性结构 c) 树型结构 d) 图形结构

4) 图形结构: 数据元素之间存在着多对多的关系, 图形结构也称作网状结构。

3. 数据的存储结构

数据结构包括数据的逻辑结构和数据的物理结构。数据的逻辑结构可以看作是从具体问题抽象出来的数学模型, 它与数据的存储无关。我们研究数据结构的目的是为了在计算机中实现对它的操作, 为此还需要研究如何在计算机中表示一个数据结构。数据结构在计算机中的标识 (又称映像) 称为数据的物理结构, 或称存储结构。它所研究的是数据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。

1) 顺序存储方法: 是把逻辑上相邻的元素存储在物理位置相邻的存储单元中, 由此得到的存储表示称为顺序存储结构。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, 由此得到的存储表示称为链式存储结构, 链式存储结构通常借助于程序设计语言中的指针类型来实现。

3) 除了通常采用的顺序存储方法和链式存储方法外, 有时为了查找的方便还采用索引存储方法和散列存储方法。

考点 2: 数据结构的表示

1. 二元关系表示

由前面的叙述可以知道, 数据的逻辑结构有两个要素: 一是数据元素的集合, 通常记为 D ; 二是 D 上的关系, 反映数据元素之间的前后件关系, 通常记为 R 。一个数据结构可以表示成

$$B = (D, R)$$

其中 B 表示数据结构, 为了反映 D 中各数据元素之间的前后件关系, 一般用二元组表示。例如, 家庭成员数据结构可以表示成

$$B = (D, R)$$

$$D = \{\text{父亲, 儿子, 女儿}\}$$

$$R = \{(\text{父亲, 儿子}), (\text{父亲, 女儿})\}$$

2. 图形表示

在数据结构的图形表示中, 对于集合 D 中的每个数据元素用标有元素值的方框表示, 通常称为数据结点, 并简称为结点; 对于关系 R 中的每个二元组, 用一条有向线段从前件结点 (或称为前驱结点) 指向后件结点 (或称为后继结点)。例如, 上例中的家庭成员数据结

构也可以表示成图 1-2 所示。

在数据结构中，没有前件的结点称为根结点；没有后件的结点称为终端结点（或叶子结点）。一个数据结构中的结点可能是动态变化的。根据需要或在处理过程中，可以在数据结构中增加新结点（称为插入运算），或删除某个结点（称为删除运算）。这两种运算是数据结构的基本运算，除此之外，还有查找、分类、合并、复制和修改等运算。

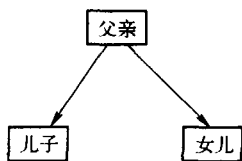


图 1-2 数据结构的图形表示

考点 3：线性结构与非线性结构

一个数据结构中如果一个数据元素都没有，则称该数据结构为空的数据结构。

根据数据结构中各数据元素之间前后件关系的复杂程度，又可以将数据结构分为两大类：线性结构和非线性结构。

如果一个非空的数据结构满足下列两个条件：

- 1) 有且只有一个根结点；
- 2) 每个结点最多有一个直接前驱，也最多有一个直接后继。

则称该数据结构为线性结构，又称线性表。

如果一个数据结构不是线性结构，则称之为非线性结构。在非线性结构中，一个结点可以有多个直接后继，或者有多个直接前驱；或者既有多个直接后继又有多个直接前驱。树型结构和图形结构都是非线性结构。

1.2.2 经典题解

【例 1】数据在计算机内存中的表示是指_____。

- | | |
|------------|--------------|
| A) 数据的存储结构 | B) 数据结构 |
| C) 数据的逻辑结构 | D) 数据元素之间的关系 |

答案：A。解析：数据结构是指相互之间存在一种或多种特定关系的数据元素的集合，它包括数据的逻辑结构和存储结构，因此 B) 错。数据的逻辑结构描述的是数据之间的逻辑关系，因此 C) 错。数据元素之间的关系是指数据的逻辑结构，因此 D) 错。

【例 2】数据的_____包括集合、线性结构、树型结构和图形结构四种基本类型。

- | | |
|---------|---------|
| A) 算法描述 | B) 基本运算 |
| C) 逻辑结构 | D) 存储结构 |

答案：C。解析：数据结构是数据元素之间逻辑关系的整体，根据数据元素之间关系的不同特性，数据的逻辑结构通常包括集合、线性结构、树型结构和图形结构四种基本类型。

【例 3】在数据结构中，从逻辑上可以把数据结构分成_____。

- | | |
|---------------|---------------|
| A) 动态结构和静态结构 | B) 紧凑结构和非紧凑结构 |
| C) 线性结构和非线性结构 | D) 内部结构和外部结构 |

答案：C。解析：逻辑结构反映数据元素之间的逻辑关系，线性结构表示数据元素之间为一对一的关系，非线性结构表示数据元素之间为一对多或者多对一的关系，所以答案为 C)。



1.2.3 即学即练

一、选择题

1. _____中任何两个结点之间都没有逻辑关系。
A) 集合 B) 图形结构
C) 树型结构 D) 线性结构
2. 数据的存储结构包括顺序、_____、索引和散列四种基本类型。
A) 向量 B) 数组
C) 集合 D) 链式

二、填空题

1. 数据结构研究的主要内容包括_____、_____和数据元素之间的联系三方面。
2. 从逻辑结构上讲, 数据结构主要分为两大类, 它们是_____和_____。
3. 一个数据结构在计算机中的表示(映象)称为_____。
4. 数据的存储结构通常分为_____和_____两种。
5. 在线性结构和树型结构中, 前驱结点和后继结点之间分别存在着_____和_____的联系。

1.2.4 即学即练答案

一、选择题

1. A
2. D

二、填空题

1. 数据存储结构; 数据逻辑结构
2. 线性数据结构; 非线性数据结构
3. 数据的存储结构
4. 顺序存储结构; 链式存储结构
5. 一对一; 一对多

1.3 线性表

1.3.1 考点分析

考点 1: 线性表的基本概念

线性表 (Linear List) 是最简单、最常用的一种数据结构, 其定义如下:

线性表是具有相同数据类型的 $n(n \geq 0)$ 个数据元素组成的有限序列, 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顺序关系。将 a_{i-1} 称为 a_i 的直接前趋, a_{i+1} 称为 a_i 的直接后继。对于非空的线性表, 有如下一些结构特征:

- 1) 有且只有一个根结点 a_1 , 无前趋。
- 2) 有且只有一个终端结点 a_n , 无后继。
- 3) 除根结点和终端结点外, 其他所有结点有且仅有一个直接前趋, 也有且仅有一个直接后继。



考点 2: 线性表的顺序存储结构——顺序表

在计算机中存放线性表, 一种最简单的方法是顺序存储, 也称为顺序分配。

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序存放线性表的各元素, 用这种存储形式存储的线性表称其为顺序表。线性表的顺序存储结构具有以下两个基本特点:

- 1) 线性表中所有元素所占的存储空间是连续的。
- 2) 线性表中各元素在存储空间中是按逻辑顺序依次存放的。

在程序设计语言中, 一维数组在内存中占用的存储空间就是一组连续的存储区域, 因此, 用一维数组来表示顺序表的数据存储区域是再合适不过的。考虑到线性表的运算有插入、删除等运算, 即表长是可变的, 因此, 数组的容量需设计得足够大, 设用: `data[MAXSIZE]` 来表示, 其中 `MAXSIZE` 是一个根据实际问题定义的足够大的整数, 线性表中的数据从 `data[0]` 开始依次顺序存放, 但当前线性表中的实际元素个数可能未达到 `MAXSIZE` 多个, 因此需用一个变量 `last` 记录当前线性表中最后一个元素在数组中的位置, 即 `last` 起一个指针的作用, 始终指向线性表中最后一个元素, 因此, 表空时 `last=-1`。这种存储思想的具体描述可以是多样的。

从结构性上考虑, 通常将 `data` 和 `last` 封装成一个结构作为顺序表的类型:

```
typedef struct
{
    datatype data[MAXSIZE];
    int last;
} SeqList;
```

定义一个顺序表: `SeqList L;`

这样表示的线性表如图 1-3a 所示。表长= $L.last+1$, 线性表中的数据元素 $a_1 \sim a_n$ 分别存放在 `L.data[0]` 至 `L.data[L.last]` 中。

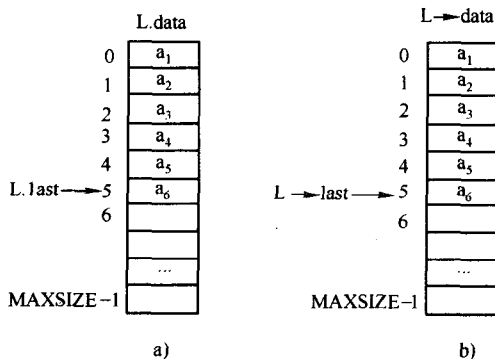


图 1-3 线性表的顺序存储示意图

a) 结构体系表示 b) 指针表示

由于算法通常用 C 语言描述, 根据 C 语言中的一些规则, 有时定义一个指向 `SeqList` 类型的指针更为方便:

```
SeqList *L;
```



L 是一个指针变量, 线性表的存储空间通过 $L = \text{malloc}(\text{sizeof}(\text{SeqList}))$ 操作来获得。

L 中存放的是顺序表的地址, 这样表示的线性表如图 1-3b 所示。表长表示为 $(*L).last$ 或 $L \rightarrow last + 1$, 线性表的存储区域为 $L \rightarrow data$, 线性表中数据元素的存储空间为 $L \rightarrow data[0] \sim L \rightarrow data[L \rightarrow last]$ 。

考点 3: 顺序表的基本运算

1. 插入运算

线性表的插入是指在表的第 i ($1 \leq i \leq n+1$) 个位置上插入一个值为 x 的新元素, 插入后使原表长为 n 的表:

$$(a_1, a_2, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n)$$

成为表长为 $n+1$ 的表:

$$(a_1, a_2, \dots, a_{i-1}, x, a_i, a_{i+1}, \dots, a_n)$$

顺序表上完成这一运算通过以下步骤进行:

- 1) 将 $a_i \sim a_n$ 顺序向后移动, 为新元素让出位置。
- 2) 将 x 置入空出的第 i 个位置。
- 3) 修改 $last$ 指针 (相当于修改表长), 使之仍指向最后一个元素。

插入算法的时间性能分析:

顺序表的插入运算, 时间主要消耗在了数据的移动上, 在第 i 个位置上插入 x , 从 $a_i \sim a_n$ 都要向后移动一个位置, 共需要移动 $n-i+1$ 个元素, 而 i 的取值范围: $1 \leq i \leq n+1$, 即有 $n+1$ 个位置可以插入。设在第 i 个位置上作插入的概率为 P_i , 则平均移动数据元素的次数:

$$E_{in} = \sum_{i=1}^{n+1} p_i (n-i+1)$$

设: $P_i = 1/(n+1)$, 即为等概率情况, 则

$$E_{in} = \sum_{i=1}^{n+1} p_i (n-i+1) = \frac{1}{n+1} \sum_{i=1}^{n+1} (n-i+1) = \frac{n}{2}$$

这说明: 在顺序表上做插入操作需移动表中一半的数据元素, 显然时间复杂度为 $O(n)$ 。

2. 删除运算 DeleteList(L, i)

线性表的删除运算是指将表中第 i ($1 \leq i \leq n$) 个元素从线性表中去掉, 删除后使原表长为 n 的线性表:

$$(a_1, a_2, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n)$$

成为表长为 $n-1$ 的线性表:

$$(a_1, a_2, \dots, a_{i-1}, a_{i+1}, \dots, a_n)$$

顺序表上完成这一运算的步骤如下:

- 1) 将 $a_{i+1} \sim a_n$ 顺序向前移动。
- 2) 修改 $last$ 指针 (相当于修改表长) 使之仍指向最后一个元素。

删除算法的时间性能分析:

与插入运算相同, 其时间主要消耗在了移动表中元素上, 删除第 i 个元素时, 其后面的元素 $a_{i+1} \sim a_n$ 都要向前移动一个位置, 共移动了 $n-i$ 个元素, 所以平均移动数据元素的次数:



$$E_{de} = \sum_{i=1}^n p_i (n-i)$$

在等概率情况下, $p_i = 1/n$, 则:

$$E_{de} = \sum_{i=1}^n p_i (n-i) = \frac{1}{n} \sum_{i=1}^{n+1} (n-i) = \frac{n-1}{2}$$

这说明顺序表上作删除运算时大约需要移动表中一半的元素，显然该算法的时间复杂度为 $O(n)$ 。

1.3.2 经典题解

【例 1】在一个长度为 n 的顺序表中, 向第 i 个元素 ($1 \leq i \leq n+1$) 位置插入一个新元素时, 需要从后向前依次后移_____个元素。

- A) $n-i$
B) i
C) $n-i-1$
D) $n-i+1$

答案: D。解析: 根据顺序表的插入运算的定义知道, 在第 i 个位置上插入 x , 从 $a_1 \sim a_n$ 都要向后移动一个位置, 共需要移动 $n-i+1$ 个元素。

【例 2】从一个长度为 n 的顺序表中，删除第 i 个元素 ($1 \leq i \leq n+1$) 时，需要从前向后依次向前移_____个元素。

- [illegible]

答案: B。解析: 与插入运算相同, 删除第 i 个元素时, 其后面的元素 $a_{i+1} \sim a_n$ 都要向前移动一个位置, 共移动了 $n-i$ 个元素。

【例 3】对顺序存储的线性表，设其长度为 n ，在任何位置上插入或删除操作都是等概率的，插入一个元素时平均移动表中的_____个元素。

- A) $n/2$
B) $(n-1)/2$
C) $(n+1)/2$
D) n

答案：A。解析：在顺序表中，插入操作可在第 1, 2, ..., n, n+1 个位置上进行，对应的移动表中元素的个数分别是 n, n-1, ..., 1, 0，它们的和为 $s=n(n+1)/2$ 。在任何位置上插入或删除操作都是等概率时，插入一个元素平均要移动元素个数为 $s/n+1$ 个，即 $n/2$ 个。

1.3.3 即学即练

一、选择题

1. 在一个长度为 n 的顺序表中, 删除值为 x 的元素需要比较和移动元素的平均次数是_____。

- A) $n/2$
B) $(n+1)/2$
C) n
D) $n+1$

2. 在一个顺序表的表尾插入一个元素的时间复杂度的量级为_____。

- A) $O(n)$
B) $O(n \log_2 n)$
C) $O(1)$
D) $O(\log_2 n)$

3. 向一个有 127 个元素的顺序表中插入一个新元素并保存, 原来顺序不变, 平均要移动的元素个数是_____。

A) 8

B) 63.5

C) 63

D) 7

二、填空题

1. 线性表中_____称为表的长度。

2. 如果向一个长度为 n 的顺序表中的第 i 个元素 ($0 \leq i \leq n-1$) 之前插入一个元素, 需要向后移动_____元素。

3. 在一个长度为 n 的顺序表中的删除第 i 个元素 ($0 \leq i \leq n-1$), 需要向前移动_____元素。

1.3.4 即学即练答案

一、选择题

1. C 2. C 3. B

二、填空题

1. 数据元素的个数 2. $n-i+1$ 3. $n-i$

1.4 栈和队列

1.4.1 考点分析

考点 1: 栈的定义

栈 (Stack) 是一种只允许在一端进行插入和删除的线性表, 它是一种操作受限的线性表。表中只允许进行插入和删除的一端称为栈顶 (top), 另一端称为栈底 (bottom)。栈的插入操作通常称为入栈或进栈, 其删除操作则称为出栈或退栈。当栈中无数据元素时, 称为空栈。

根据栈的定义可知, 栈顶元素总是最后入栈的, 因而是最先出栈; 栈底元素总是最先入栈的, 因而也是最后出栈。这种表是按照后进先出 (Last In First Out, LIFO) 的原则组织数据的, 因此, 栈也被称为“后进先出”的线性表。

图 1-4 是一个栈的示意图, 通常用指针 top 指示栈顶的位置, 用指针 bottom 指向栈底。栈顶指针 top 动态反映栈的当前位置。

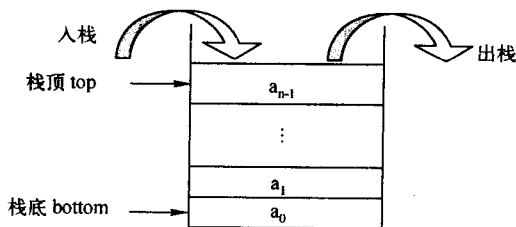


图 1-4 栈的示意图