

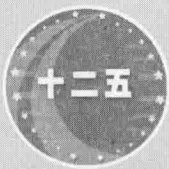


21世纪高等学校计算机公共课程“十二五”规划教材

全国计算机等级考试指导 (二级公共基础知识)

QUANGUO JISUANJI DENGJI KAOSHI ZHIDAO
(ERJI GONGGONG JICHU ZHISHI)

孙勤红 刘粉香 沈凤仙 朱颖雯 编著
顾 洪 主审



十二五

21世纪高等学校计算机公共课程“十二五”规划教材

全国计算机等级考试指导 (二级公共基础知识)

QUANGUO JISUANJI DENGJI KAOSHI ZHIDAO
(ERJI GONGGONG JICHU ZHISHI)

孙勤红 刘粉香 沈凤仙 朱颖雯 编著
顾 洪 主审

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书依据《全国计算机等级考试二级公共基础知识考试大纲》而编写,依据考纲提炼出重要常考知识点,精析历年考题,使考生能尽快掌握数据结构与算法、程序设计基础、软件工程基础以及数据库设计基础四部分的相关知识。本书研究 2005—2012 年考试真题的主要考点、要点和难点,汇集历年考题,把握考试动向,使考生能够从容应对全国计算机等级(二级)考试。

本书适用于参加全国计算机等级(二级)考试语言类考试的考生,也可作为全国计算机等级考试(二级)培训机构的教材,以及作为各高校二级考试教程的参考书。

图书在版编目(CIP)数据

全国计算机等级考试指导. 二级公共基础知识 / 孙勤红等编著. —北京:中国铁道出版社, 2013. 1

21 世纪高等学校计算机公共课程“十二五”规划教材
ISBN 978-7-113-15679-4

I. ①全… II. ①孙… III. ①电子计算机—水平考试—高等学校—教材 IV. ①TP3

中国版本图书馆 CIP 数据核字(2012)第 307755 号

书 名: 全国计算机等级考试指导(二级公共基础知识)

作 者: 孙勤红 刘粉香 沈凤仙 朱颖雯 编著

策 划: 吴宏伟 孟 欣

读者热线: 400-668-0820

责任编辑: 孟 欣 彭立辉

封面设计: 付 巍

封面制作: 白 雪

责任印制: 李 佳

出版发行: 中国铁道出版社(100054,北京市西城区右安门西街 8 号)

网 址: <http://www.51eds.com>

印 刷: 北京市燕鑫印刷有限公司

版 次: 2013 年 1 月第 1 版 2013 年 1 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 8.75 字数: 207 千

印 数: 1~3 000 册

书 号: ISBN 978-7-113-15679-4

定 价: 19.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 63549504

本书根据教育部考试中心颁布的最新版《全国计算机等级考试二级公共基础知识考试大纲》的要求编写而成。

本书对考纲中规定的知识点进行了详细透彻的讲解,并且对2005年春季至2012年春季的所有考试真题进行了分析和总结,对考生参加全国计算机等级考试(二级)语言类考试具有指导性的作用和帮助。

全书共分5章,每章主要内容如下:

第1章主要介绍了数据结构和算法的概念、线性表的定义、特殊的线性表栈和队列的概念及其基本运算和性质、二叉树的基本概念和3种遍历方法、2种典型的查找方法——顺序查找和二分查找,最后介绍了算法中常用的几种排序方法。

第2章介绍了程序设计方法、结构化程序设计的思想及3种基本结构(即顺序结构、选择结构和循环结构)。

第3章介绍了软件工程的基本概念、结构化分析方法和结构化设计方法、软件测试和程序调试的常用方法。

第4章介绍了数据库相关知识。首先介绍了数据库的相关概念,其次介绍了概念模型、逻辑模型等内容,以及关系代数中常用的传统运算和关系运算,最后介绍了数据库设计的步骤。

第5章为各章节知识结构图及考试情况说明。

本书由孙勤红、刘粉香、沈凤仙、朱颖雯编著,其中第1、2章和附录由孙勤红编写,第3章由沈凤仙编写,第4章由刘粉香编写,第5章由朱颖雯编写。全书由孙勤红统稿,由顾洪老师主审。苏兆中老师为本书提出了很多宝贵意见和建议,单启成老师、陈林老师和张凤莲老师等为编写本书提供了很多帮助,在此表示感谢。同时感谢中国铁道出版社的各位编辑为本书的出版所提供的大力支持和帮助。

本书适用于参加全国计算机等级考试(二级)的各类计算机语言(如VFP、C语言、VC++、C++、VB、Access、Java等科目)考试的读者,也适用于对计算机感兴趣的读者。

由于编者水平有限,书中疏漏或不妥之处在所难免,恳请广大读者批评指正。

编者

2012年12月

第 1 章 数据结构与算法	1
1.1 数据结构的基本概念	1
1.2 算法	4
1.2.1 算法的基本概念	4
1.2.2 算法分析	6
1.3 线性表及其存储结构	9
1.4 栈和队列	15
1.4.1 栈的基本概述及基本运算	15
1.4.2 队列的基本概述及基本运算	20
1.5 树和二叉树	25
1.5.1 关于树和二叉树的基本术语和性质	25
1.5.2 满二叉树与完全二叉树	31
1.5.3 二叉树的遍历	32
1.6 查找技术	38
1.7 排序技术	40
第 2 章 程序设计基础	47
2.1 程序设计方法与风格	47
2.2 结构化程序设计	49
2.3 面向对象的程序设计	52
第 3 章 软件工程基础	55
3.1 软件工程的基本概念	55
3.1.1 软件的定义与特点	55
3.1.2 软件危机与软件工程	57
3.1.3 软件工程过程与软件生命周期	59
3.2 结构化分析方法	61
3.2.1 需求分析与需求分析方法	61
3.2.2 结构化分析方法	63
3.3 结构化设计方法	66
3.3.1 软件设计的基本概念	66
3.3.2 概要设计	68
3.3.3 详细设计	71
3.4 软件测试	74
3.4.1 软件测试的目的和原则	74
3.4.2 软件测试分类	76
3.5 程序的调试	78

第4章 数据库设计基础	81
4.1 数据库系统的基本概念	81
4.1.1 数据库技术的发展及组成	81
4.1.2 数据库系统的特点	84
4.1.3 数据库系统的内部结构体系	86
4.2 数据模型	87
4.2.1 数据模型的基本概念	87
4.2.2 概念模型和 E-R 模型	88
4.2.3 逻辑模型的分类	91
4.3 关系数据库	92
4.3.1 关系数据模型的数据结构	92
4.3.2 关系操作	94
4.3.3 关系的完整性	101
4.4 数据库设计	101
第5章 各章知识结构图及考试情况说明	105
5.1 各章知识结构图	105
5.2 考试情况	113
附录 A 全国计算机等级考试二级公共基础知识考试大纲	115
附录 B 最近两年考试真题	117
附录 C 各章节重要知识点	125
参考文献	132

第 1 章 数据结构与算法

本章主要介绍了以下内容：算法的概念及其评价、线性表的表示、栈和队列的概念、二叉树的概念和性质、查找的方法和排序的种类。

重点掌握：

- 掌握算法的概念和评价算法的方法（时间复杂度和空间复杂度）；
- 熟悉数据结构的逻辑结构和物理结构的概念；
- 掌握线性表的两种存储结构：顺序存储和链式存储；
- 掌握栈和队列的概念和区别，并会应用其性质；
- 掌握二叉树的概念，应用二叉树的性质；
- 掌握查找方法的顺序查找和二分查找的适用范围和区别；
- 熟悉常见的几种排序方法：插入排序、选择排序、快速排序、堆排序，以及这几种算法的时间复杂度。

一位瑞士计算机科学家尼·沃思（N.Wirth）在 20 世纪 70 年代曾经提出过一个著名的公式：数据结构+算法=程序。人们常说：软件的主体是程序，程序的核心是算法。这是因为要使用计算机解决某个问题，首先必须针对该问题设计一个解题方法和步骤，然后再据此编写程序交给计算机执行。这里所说的解题方法和步骤就是“算法”，描述问题的对象称为“数据结构”，这是编写程序时首先考虑的两个重要方面。

1.1 数据结构的基本概念

数据结构作为计算机的一门学科，主要研究和讨论以下 3 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 在对数据进行处理时，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。

1. 几个相关概念

数据元素：数据的基本单位，在计算机中通常作为整体进行考虑和处理。例如，一年 12 个月就可以作为一年月份的数据元素；赵、钱、孙、李等可以作为百家姓的数据元素。

(1) 数据对象

数据对象指性质相同的数据元素的集合。例如，字母字符数据对象是集合 $C=\{'A','B'\dots$

'Z','a','b'...'z'}, 学生基本信息表也是一个数据对象。

(2) 前后件

前后件指数据元素之间的关系和结构。春、夏、秋、冬按照季节以此更替, 春是夏的前件, 夏是春的后件。英文字母 A 是 B 的前件, C 是 B 的后件。

(3) 数据结构

数据结构是指相互之间存在一种或多种特性关系的数据元素的集合。即数据结构是指带有结构的数据元素的集合。因此, 一个数据结构应包含以下两方面的信息: 表示数据元素的信息, 表示各数据元素之间的前后件关系(结构)。例如, 学生信息的统计表的数据。

数据结构包括逻辑结构和物理结构两个层次, 下面分别介绍这两个概念。

2. 数据的逻辑结构

数据的逻辑结构是对数据元素之间的逻辑关系的描述, 它与数据的存储无关, 是独立于计算机的。因此, 数据的逻辑结构可以看成是从具体问题抽象出来的数据模型。

数据的逻辑结构有两个要素: 一是数据元素, 二是关系。

例如, 一年有 24 个节气, 构成了一个简单的农历节气关系。又如, 一个年级的学生按成绩总分由高到低排列, 则学生信息表中各信息项之间构成了成绩高低的关系。

可以使用二元组来表示一种数据结构:

$$B=(D,R)$$

其中 B 是一种数据结构, D 是数据元素的集合, R 是 D 上的二元关系的集合。

数据的逻辑结构包括集合结构、线性结构、树状结构和图或网状结构 4 种。

(1) 集合结构

数据元素之间除了“属于同一集合”的关系外, 别无其他关系。

(2) 线性结构

数据元素之间存在一对一的关系。例如, 将学生信息数据按照其入学报到的时间先后顺序进行排列, 将组成一个线性结构。图 1-1 和图 1-2 都是线性结构。

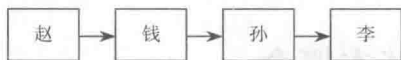


图 1-1



图 1-2

(3) 树状结构

数据元素之间存在一对多的关系。例如, 家庭的族谱, 如图 1-3 所示。

(4) 图或网状结构

数据元素之间存在多对多的关系。例如, 图 1-4 即为一个图。

其中集合结构、树状结构、图或网状结构都属于非线性结构。即图 1-1 和图 1-2 是线性结构, 图 1-3 和图 1-4 是非线性结构。

线性结构包括线性表、栈和队列, 非线性结构主要包括树、二叉树和图等。

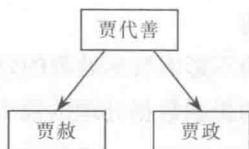


图 1-3

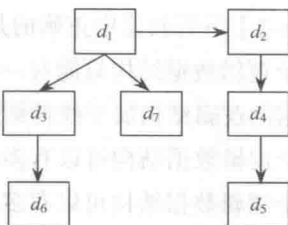


图 1-4

3. 数据的存储结构

数据的存储结构也称为数据的物理结构，是数据的逻辑结构在计算机存储空间中的存放形式。把数据对象存放在计算机时，通常要求既要存储各数据元素的数据，又要存储数据元素之间的逻辑关系。

在数据结构中，数据的逻辑结构与数据的存储结构可以是不同的，如图 1-5 所示。

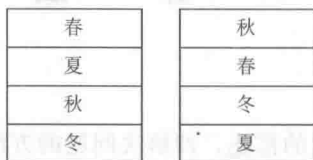


图 1-5 不同的存储结构

例如，由{春、夏、秋、冬}构成的数据结构，在逻辑结构上，春在夏的前面，夏在春的后面。但在计算机的存储结构中，春不一定在夏的前面，夏不一定在春的后面。

因此，在数据的存储结构中不仅要存放各元素的信息，还需要存放各数据元素之间的前后件关系的信息，以表示存放在存储空间中的各元素之间的逻辑关系。

一种数据的逻辑结构根据需要可以表示成多种存储结构，常用的存储结构有顺序、链表、索引等（参见 1.3 节）。不同的存储结构其数据处理的效率是不同的。

注意：数据的存储结构与数据的逻辑结构没有必然的联系。例如，线性结构可以采用顺序存储结构，也可以采用链式存储结构，同样对于非线性结构如树和二叉树也可以采用多种存储结构。

4. 真题解析

【试题 1-1】数据的存储结构是指（ ）。

- A. 存储在外存中的数据
- B. 数据所占的存储空间量
- C. 数据在计算机中的顺序存储方式
- D. 数据的逻辑结构在计算机中的表示

答案：D

解析：数据的存储结构是指数据的逻辑结构在计算机存储空间中的存放形式，“在计算机中的表示”可理解为“在计算机存储空间中的存放形式”。

【试题 1-2】下列叙述中正确的是()。

- A. 一个逻辑数据结构只能有一种存储结构
- B. 数据的逻辑结构属于线性结构, 存储结构属于非线性结构
- C. 一个逻辑数据结构可以有多种存储结构, 且各种存储结构不影响数据处理的效率
- D. 一个逻辑数据结构可以有多种存储结构, 且各种存储结构影响数据处理的效率

答案: D

解析: 逻辑结构是对数据元素之间逻辑关系的描述; 存储结构是指逻辑结构在计算机存储空间中的存放形式, 同一个数据的逻辑结构可以有多种存储结构, 不同的存储结构其数据处理的效率是不同的; 线性结构和非线性结构表述的是数据结构中各数据元素之间前后件关系, 此关系也必须保存在其对应的存储结构中。

1.2 算 法

1.2.1 算法的基本概念

算法: 指解题方案准确而完整的描述, 即解决问题的方法与步骤。

例如: 如何求一元二次方程 $ax^2 + bx + c = 0$ 的解?

求解步骤如下:

第一步: 计算 $\Delta = b^2 - 4ac$;

第二步: 判断 Δ 的情况, 如果 $\Delta \geq 0$, 则 $x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}$ 。如果 $\Delta < 0$, 则方程无解;

第三步: 输出方程的根或无解的信息。

1. 算法的基本特征

(1) 可行性

可行性也叫能行性, 算法中的每一步应该能够准确执行。例如, 一元二次方程, 正常情况下输入不同的 a 、 b 、 c 这 3 个数, 可得到不同的输出结果。当输入的 $a=0$ 时, 算法就无法执行 ($\Delta \geq 0$ 时, 分母为 0!)。

(2) 确定性

算法中的每一步操作必须有确切的含义, 即每一步运算应该执行何种操作必须具有明确的定义, 不允许有模棱两可的解释, 也不允许有多义性。

例如, “输入一元二次方程的一个系数”, 既可理解为二次项系数也可理解为一次项系数甚至是常数项, 这就是多义性。

(3) 有穷性

有穷性是指算法必须在有限的时间内做完, 即算法必须在执行有限个步骤之后终止, 而不是无终止地循环。

(4) 有输入信息

算法要求有零个或多个输入。

(5) 有输出信息

算法要求至少产生一个输出，以看到运行处理的结果。

注意：有的教材中把(4)和(5)归为一类，称为拥有足够的情报。

2. 算法的基本要素

一个算法通常由两种基本要素组成：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

计算机可以执行的基本操作都是使用指令来描述和完成，所有指令的集合构成计算机指令系统。计算机程序解题就是按算法要求从指令系统中选择合适的指令，组成指令序列，执行指令完成解题。

一般的计算机系统中包含4类基本运算和操作：

- ① 算术运算：主要包括加、减、乘、除等基本运算；
- ② 逻辑运算：主要包括与、或、非、异或等运算；
- ③ 关系运算：主要包括大于、小于、等于、不等于等运算；
- ④ 数据传输：包括赋值、输入、输出等操作。

(2) 算法的控制结构

一个算法的功能不仅取决于所选用的操作，而且还与各操作之间的执行顺序有关。算法中各操作之间的执行顺序称为算法的控制结构。算法的描述工具有3种：传统流程图、N-S结构化流程图、算法描述语言。

算法的基本控制结构有以下3种：顺序、选择（也称为分支）和循环（也称为重复）。

3. 常用的算法设计方法

计算机解题的过程实际上是在实施某种算法，这种算法就称为计算机算法。常用的有如下几种：

(1) 列举法

列举法也称为穷尽法，是计算机算法中的几个基础算法之一，其基本思想是根据提供的问题列举所有可能的情况。例如，由A、B、C、D这4个字母组成的排列方式有哪些？

列举法的特点是算法比较简单，但是当列举的可能情况较多时，执行列举算法的工作量将会增大。例如，银行储蓄使用的密码由6位数字组成，总共的可能性共有 10^6 种情况。

(2) 回溯法

回溯法是穷举思想的改进，其目的是尽可能减小解的搜索范围。找出一条试探路线，并沿该路线向前试探，如试探成功，则继续；否则返回上一步继续试探其他的路线。典型的问题如走迷宫问题、八皇后问题等是回溯法的典型。

(3) 递推法

递推是指从已知的初始条件出发，逐次推出所要求的各中间结果和最后结果。其中初始条

件或是问题本身已经给定,或是通过对问题的分析与化简而确定。

(4) 递归法

递归的基本思想是在解决复杂问题时,将问题逐层分解,降低问题的复杂程度,直到最后那些最简单的问题被解决后,再沿着原路分解的逆过程进行综合。

例如,比较著名的汉诺塔问题就是使用递归法完成。

(5) 减半递推技术(也称为分治法)

所谓分治法就是把一个复杂的问题分成两个或更多的相同或相似的子问题,再把子问题分成更小的子问题,直到最后子问题可以简单地直接求解。比如快速排序、归并排序等都是使用这种方法。

4. 真题解析

【试题 1-3】算法的有穷性是指()。

- A. 算法程序运行的时间是有限的
- B. 算法程序所处理的数据是有限的
- C. 算法程序的长度是有限的
- D. 算法只能被有限的用户使用

答案: A

解析: 算法具有 5 个特点或要求: 确定性、有穷性、能行性、输入、输出。确定性是指每一步运算应该执行何种操作必须是明确的; 有穷性是指一个算法必须在有限的时间内做完,即执行有限步的操作后终止; 能行性是指算法中有待实现的操作都是可以实现的; 输入是指有 0 个或多个输入; 输出是指至少产生一个输出。

【试题 1-4】下列叙述中正确的是()。

- A. 算法就是程序
- B. 设计算法时只需要考虑数据结构的设计
- C. 设计算法时只需要考虑结果的可靠性
- D. 以上三种说法都不对

答案: D

解析: 算法是解决问题的方法和步骤,而程序一般是指能够被计算机执行的指令代码。设计算法时要依靠逻辑结构和物理结构以及算法的特性。

【试题 1-5】问题处理方案正确而完整的描述称为_____。

答案: 算法

1.2.2 算法分析

1. 算法评价的依据

一个问题的计算机求解,可以有许多不同的算法,不同的算法运行时,执行过程也许会不一样,运行的效率可能差别很大。为此,必须对算法进行分析和评价。

通常从以下几个方面来评价一个算法的性能:

- ① 正确性:也称为有效性,就是要求算法在合理的输入条件下,能够在有限的运行时间内得出正确的结果。
- ② 可读性:要求算法是易读的、易理解的。
- ③ 健壮性:要求算法具有对输入数据的合法性进行检查的功能,能够检测出非法的数据输入,并且进行适当的处理。
- ④ 效率:算法的效率是指一个算法运行的时间和空间代价,即时间复杂度和空间复杂度。

2. 算法的时间复杂度

所谓算法的时间复杂度,就是指算法运行所需要的时间,即算法的时间代价。

一个算法在计算机系统上的实际运行时间除了与算法本身的性能有关外,还与计算机系统的硬件、软件的性能有关,与问题的规模也有关。

算法时间复杂度一般以执行算法所需要的计算工作量计算,算法工作量用算法所执行的基本运算次数来度量。

算法的时间复杂度通常用函数来描述,即时间复杂度函数,是问题规模的函数。通常把算法运行的时间特性记为 $T(n)$, n 表示问题规模的大小。

例如,对 n 个整数元素进行排序, n 就是问题规模的度量值。算法的时间特性 $T(n)$ 是 n 的函数,通常表示为多项式。

算法的时间复杂度指的是问题的规模 n 充分大时完成该算法所需时间的数量级表示。例如,某程序运行的时间特性为 $T(n)=2n^3+3n^2+2n+1$,则表明该程序运行时所需的时间与 n^3 成正比,引入符号 O ,当 n 充分大时,则有 $T(n)=O(n^3)$,则 $O(n^3)$ 称作该算法的时间复杂度。

例如,求 1 到 n 的和,程序如下:

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

程序中的 $s=0$ 执行 1 次,for 循环执行 n 次,所以总共的执行运算次数为 $n+1$,则此算法时间复杂度为 $O(n)$ 。

在有些情况下,算法中关键操作执行的次数与问题的特定输入有关。例如,在学生表中找一个学生,如果第一个就是要找的学生,则只要经过一次比较即可完成查找。如果要找的学生在最后一个或者根本不存在,则查找过程需要经过与全部记录的关键字进行比较才能结束。

在这种情况下,时间复杂度计量方法有两种:

第一种是最坏情况时间复杂度,即以在所有输入下的计算量的最大值为算法的计算量;

第二种是算法的平均时间复杂度,以算法在所有输入下的计算量的加权平均值作为算法的计算量。

3. 算法的空间复杂度

算法的空间复杂度是对一个算法在运行过程中所需要的辅助存储空间大小的度量。一个算法在计算机上运行所需要的存储空间包括以下 3 个方面:

- ① 算法本身 (如程序指令、变量、常量) 所需要的存储空间;
- ② 输入数据占用的存储空间;
- ③ 算法运行过程中所需要的辅助空间。

算法的空间复杂度也是问题规模的函数, 也用大 O 符号表示。

注意: 同一个问题使用不同的方法解决, 则算法的时间和空间复杂度可能是不同的。

4. 真题解析

【试题 1-6】算法的时间复杂度是指 ()。

- A. 算法的执行时间
- B. 算法所处理的数据量
- C. 算法程序中的语句或指令条数
- D. 算法在执行过程中所需要的基本运算次数

答案: D

解析: 算法的时间复杂度, 是指执行算法所需要的计算工作量, 一般采用算法所执行的基本运算次数来度量, 是问题规模的函数。

【试题 1-7】算法的空间复杂度是指 ()。

- A. 算法在执行过程中所需要的计算机存储空间
- B. 算法所处理的数据量
- C. 算法程序中的语句或指令条数
- D. 算法在执行过程中所需要的临时工作单元数

答案: A

解析: 算法的空间复杂度是指执行算法所需要的内存空间, 包括算法程序所占空间、输入的初始数据所占空间和执行过程中所需要的额外空间。

【试题 1-8】下列叙述中正确的是 ()。

- A. 一个算法的空间复杂度大, 则其时间复杂度也必定大
- B. 一个算法的空间复杂度大, 则其时间复杂度必定小
- C. 一个算法的时间复杂度大, 则其空间复杂度必定小
- D. 上述 3 种说法都不对

答案: D

解析: 算法的时间和空间复杂度没有直接的关系, 它们仅与算法本身相关。

【试题 1-9】下列叙述中正确的是 ()。

- A. 程序执行的效率与数据的存储结构密切相关
- B. 程序执行的效率只取决于程序的控制结构
- C. 程序执行的效率只取决于所处理的数据量
- D. 以上 3 种说法都不对

答案: A

【试题 1-10】下列叙述中正确的是 ()。

- A. 算法的效率只与问题的规模有关, 而与数据的存储结构无关
- B. 算法的时间复杂度是指执行算法所需要的计算工作量
- C. 数据的逻辑结构与存储结构是一一对应的
- D. 算法的时间复杂度与空间复杂度一定相关

答案: B

解析: 算法的效率不仅与问题的规模相关, 采用不同的存储结构(顺序、链式)会影响算法的效率, 所以 A 错; 数据的逻辑结构有线性、树状、图或网状等, 它们可以采用不同的存储结构, 不存在一一对应关系, C 错; 衡量算法的两大指标时间和空间复杂度仅与算法本身相关, 包括算法的思想、数据的存储结构、数据的输入等, D 错。

【试题 1-11】算法复杂度主要包括时间复杂度和 _____ 复杂度。

答案: 空间

1.3 线性表及其存储结构

1. 线性表的基本概念及结构特征

线性表是由 n ($n \geq 0$) 个数据元素 $a_1, a_2, a_3, \dots, a_n$ 组成的一个有限序列, 表中的每一个数据元素, 除了第一个外, 有且只有一个前件, 除了最后一个外, 有且只有一个后件。即线性表或者是一个空表, 或者可以表示为:

$(a_1, a_2, \dots, a_i, \dots, a_n)$

其中, a_i ($i=1, 2, \dots, n$) 是数据对象的元素, 通常也称为线性表中的一个结点。

非空线性表有如下一些结构特征:

- ① 有且仅有一个根结点 a_1 , 它无前件;
- ② 有且只有一个终端结点 a_n , 它无后件;

③ 除根结点与终端结点外, 其他所有结点有且只有一个前件, 也有且只有一个后件。线性表中结点的个数 n 称为线性表的长度。(当 $n=0$ 时, 称为空表)。

无序线性表: 指线性表的元素没有一定的规律, 是随机排列的。

有序线性表: 指线性表的元素按照一定规律如数值大小排列的线性表, 是一种特殊的线性表。

线性表的存储结构有两种: 顺序存储结构和链式存储结构。

下面分别介绍顺序存储结构和链式存储结构。

2. 顺序表的插入运算和删除运算

(1) 顺序存储结构的特点

在计算机中, 利用一组地址连续的存储单元依次存储线性表的数据元素, 称为线性表的顺序存储结构。

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 线性表中各数据元素在存储空间中是按逻辑顺序依次存放的,即逻辑上相邻的数据元素,其物理上也是相邻的。

③ 顺序存储结构可以实现随机存取。对于一个顺序存储的线性表,只要确定了初始位置,线性表中的任一数据元素都可随机存取,所以线性表的顺序存储结构是一种随机存储的存储结构。

线性顺序表的数据的逻辑结构与数据的存储结构是相同的。例如,数据结构一年四季的元素春、夏、秋、冬中,秋处于第三个位置且位于夏后面,则不管是逻辑结构还是存储结构,都满足这个条件。

由于高级程序设计语言如 VB、C、VFP 等程序设计语言中的数组也有随机存取的特性,因此,通常都用数组来描述数据结构中的顺序存储结构。

(2) 插入运算

例如,在一个长度为 7 的线性表顺序存储 6 个数据,现在要求在第 2 个元素之前插入一个新元素 25。其插入的过程图 1-6 所示。

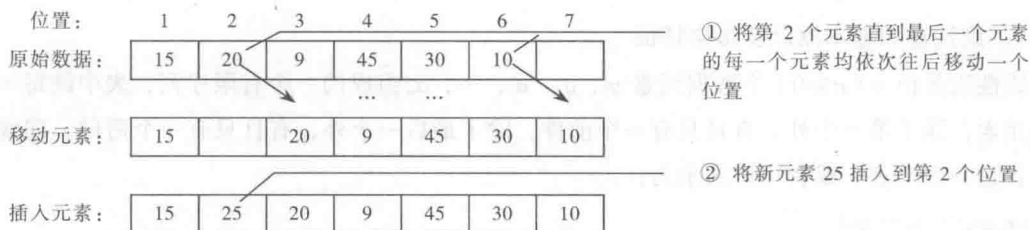


图 1-6 顺序表的插入过程

首先从最后一个元素开始直到第 2 个元素,将其中的每一个元素均依次往后移动一个位置,然后将新元素插入到第 2 个位置。

一般情况下,要在第 i 个位置处插入一个新元素涉及两个基本操作:

① 从第 i 个元素(包括第 i 个元素)开始直到最后一个元素 n (共计 $n-i+1$ 个元素),每个元素均依次后移一个位置。

② 在第 i 个位置插入新元素。

由此看出,所需移动结点的次数不仅与表的长度有关,还与插入的位置有关:

在线性表的末尾插入新元素(即 $i=n+1$): 没有结点移动,其时间复杂度为 $O(1)$;

当在线性表的第 1 个元素之前插入新元素(即 $i=1$): n 个结点依次后移一次,需要移动 n 次,其时间复杂度为 $O(n)$ 。

在平均情况下,要在线性表中插入一个新元素,需要移动表中一半的元素。其时间复杂度为 $n/2=O(n)[(0+1+2+3+\dots+n)/(n+1)]$ 。

(3) 删除运算

例如,在一个长度为 7 的线性表顺序存储 6 个数据,现在要求将第 2 个元素删除。其删除

过程如图 1-7 所示。

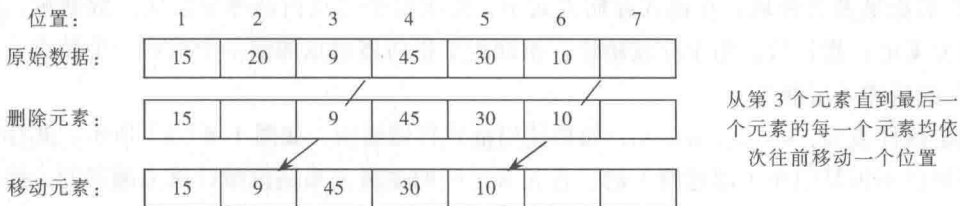


图 1-7 顺序表的删除过程

一般情况下, 要删除第 i 个位置的元素:

① 要从第 $i+1$ 个元素开始直到最后一个元素 n (共计 $n-i$ 个元素), 每一个元素均依次前移一个位置, 需要移动 $n-1$ 次。

② 删除成功后, 线性表的长度减少了 1。

与插入操作类似, 在平均情况下, 要删除第 i ($1 \leq i \leq n$) 个元素时, 需要移动表中一半的元素。即时间复杂度为 $(n-1)/2 = O(n)[(0+1+2+3+\dots+n-1)/n]$; 在最坏情况下 (即删除第 1 个元素), 时间复杂度为 $n-1 = O(n)$ 。

由线性表在顺序存储结构下的插入与删除运算可以看出: 线性表的顺序存储结构对于小线性表或者其中元素不常变动的线性表来说是合适的, 因为顺序存储的结构比较简单。但对于元素经常需要变动的大线性表就不太合适了, 因为当 n 较大时, 平均时间复杂度为 $O(n)$, 插入和删除算法效率较低。但是, 由于线性表的顺序存储结构可以实现随机存取, 所以查找元素的操作是非常容易实现的。

(4) 线性表顺序存储结构的优缺点

优点: 简单、运算方便, 对于小线性表或长度固定的线性表优越性尤其突出。可以实现元素的随机存取。

缺点:

① 插入、删除一个元素时, 在平均情况下需要移动一半的元素, 对于大的线性表或插入、删除运算很频繁的情况下, 效率都很低。

② 当为某线性表分配了一定的存储空间后, 总会出现存储空间满了以后还要插入新元素的情况, 此时若要在保持顺序存储结构的前提下, 扩充存储空间就很不方便了。

③ 在实际应用中, 往往同时会有多个线性表共享计算机的存储空间的情况, 此时为每个线性表按顺序存储结构分配多少存储空间就成为难题。如果平均分配, 就会出现有的线性表存储空间很快就存满了而出现“上溢”, 而有的线性表则有大量空闲存储空间的情况; 若要保持其顺序存储结构不变“动态”地分配存储空间就要增加不必要的开销 (比如进行大批量数据的移动)。

特点②和③可以总结为线性表的顺序存储结构需要分配连续的存储空间。

3. 线性链表的基本概念和操作

(1) 线性链表的相关概念

① 存储结点: 假设数据结构中的每一个数据结点对应于一个存储单元, 这种存储单元称