

National Computer Rank Examination

全国计算机等级考试专用辅导丛书

# 全国计算机等级考试 专用辅导教程

## 二级 公共基础知识 ——2012版——

张华明 俞露 编著



**顾**——纵观历年真题，了解试题分布，总结重点内容，提炼核心考点。  
**接**——穿插最新真题，强化考试内容，了解最新动态，把握命题规律。  
**析**——挑选经典考题，熟悉解题技巧，吃透考试题目，掌握解题方法。  
**过大必备**——精选过关试题，及时自我检测，提升学习效果，瞬间巩固提高。

全面  
实用  
权威



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

TP3-44  
Z114



郑州大学 \*04010744918 \*

National Computer Rank Examination

全国计算机等级考试专用辅导丛书

# 全国计算机等级考试 专用辅导教程

## 二级 公共基础知识 ——2012版——

张华明 俞露 编著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

TP3-44

## 内容简介

本书紧扣最新版考试大纲,结合编者多年从事命题、阅卷及培训辅导的实际经验编写而成。本书章节安排与官方教程同步,主要包括:数据结构与算法、程序设计基础、软件工程基础、数据库设计基础。章节前插入了近几年考试分布,精讲考试重点与难点,讲解过程中链接最近两年的考试真题,方便读者了解最新考试动态。章节末安排了典型题讲解和适量过关习题。书末附有20套样题和解析,供考生考前实战演练。

本书以全国计算机等级考试考生为主要读者对象,适合于考生在等考前复习使用,也可作为相关考试培训班的辅导教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

## 图书在版编目(CIP)数据

全国计算机等级考试专用辅导教程:2012版. 二级公共基础知识 / 张华明, 俞露编著. —北京: 电子工业出版社, 2012.1

(全国计算机等级考试专用辅导丛书)

ISBN 978-7-121-15212-2

I. ①全... II. ①张... ②俞... III. ①电子计算机—水平考试—自学参考资料 IV. ①TP3

中国版本图书馆CIP数据核字(2011)第241419号

责任编辑:徐津平

印刷:北京天宇星印刷厂

装订:三河市皇庄路通装订厂

出版发行:电子工业出版社

北京市海淀区万寿路173信箱

邮编:100036

开本:860×1092 1/16

印张:11.25

字数:360千字

印次:2012年1月第1次印刷

印数:4000册 定价:29.80元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 [zlts@phei.com.cn](mailto:zlts@phei.com.cn), 盗版侵权举报请发邮件到 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线:(010) 88258888。

# 前言

全国计算机等级考试是目前我国规模最大、参加人数最多的全国性计算机类水平考试，因其具有权威性、公平性和广泛性而在社会上享有良好的声誉，很多单位都把获取计算机等级证书作为单位录用、职称评定的标准之一。

应广大考生的需要，我们按教育部考试中心指定教材的篇章结构，组织从事全国计算机等级考试试题研究的老师和专家精心编写了“全国计算机等级考试专用辅导丛书”，为读者打造最简单、最实用的考试教程，目的是让考生在短时间内快速过关，知己知彼，百战百胜！

## 一、本书特色

### 1.突出标准性与严谨性

本丛书由从事全国计算机等级考试试题研究人员及在等级考试第一线从事命题研究、教学、辅导和培训的老师合作编写，层次清晰，结构严谨，导向准确。

### 2.注重典型考题的分析

紧扣新大纲要求，精讲考点、重点与难点，深入分析典型范例，抓住等级考试题眼，并提供实战训练。

### 3.突出实用性和高效性

各书的章名、节名与教育部考试中心指定教程同步，每章还精心设计以下板块。

**考情回顾：**总结本章需要掌握的重点内容，分析最近几年的考试真题，整理出试题分布、试题题型和分值，提炼出本章的考核要点。

**真题链接：**整理出最近两年的考试真题，穿插在知识点的讲解中，帮助考生理解知识点，也让考生了解最新考试动态和命题规律。

**典型考题分析：**精选出常考的典型题型，并针对每一个典型题进行详细解析，引导考生掌握重点内容，吃透重点考试题目，并且让考生了解解题方法和解题思路。

**过关必备：**针对每章知识点安排适当的过关题目，便于考生及时将知识点运用于具体题目中，方便考生检查学习效果，巩固提高所学知识。

### 4.全面模拟，实战提高

根据新大纲、新考点、新题型进行最新命题，书末提供 20 套样题及答案解析，供考生考前实战演练。

## 二、读者对象

本书可供参加全国计算机等级考试二级公共基础知识的考生复习使用，也可以作为相关等级考试培训班的辅导教材。

## 三、关于作者

本书由张华明、俞露任主编。参与本书编写工作的人员还有：李赛红、吕伟、严惠、宋永生、张艳、施俊飞、吴海涛、俞露、李胜、姚昌顺、朱贵喜、杨万扣、邓丽萍、严云洋。在此对诸位作者表示衷心的感谢。

由于作者学识有限，书中难免有错误和疏漏之处，恳请专家和广大读者指正。

编 者



# 目 录

|                        |    |
|------------------------|----|
| 第 1 章 数据结构与算法 .....    | 1  |
| 考情回顾 .....             | 1  |
| 1.1 算法 .....           | 1  |
| 1.1.1 算法的基本概念 .....    | 1  |
| 1.1.2 算法复杂度 .....      | 4  |
| 1.2 数据结构的基本概念 .....    | 5  |
| 1.2.1 什么是数据结构 .....    | 5  |
| 1.2.2 数据结构的图形表示 .....  | 7  |
| 1.2.3 线性结构与非线性结构 ..... | 8  |
| 1.3 线性表及其顺序存储结构 .....  | 8  |
| 1.3.1 线性表的基本概念 .....   | 9  |
| 1.3.2 线性表的顺序存储结构 ..... | 9  |
| 1.3.3 顺序表的插入运算 .....   | 10 |
| 1.3.4 顺序表的删除运算 .....   | 11 |
| 1.4 栈和队列 .....         | 12 |
| 1.4.1 栈及其基本运算 .....    | 12 |
| 1.4.2 队列及其基本运算 .....   | 14 |
| 1.5 线性链表 .....         | 18 |
| 1.5.1 线性链表的基本概念 .....  | 18 |
| 1.5.2 线性链表的基本运算 .....  | 19 |
| 1.5.3 循环链表及其基本运算 ..... | 21 |
| 1.6 树与二叉树 .....        | 23 |
| 1.6.1 树的基本概念 .....     | 23 |
| 1.6.2 二叉树及其基本性质 .....  | 24 |
| 1.6.3 二叉树的存储结构 .....   | 27 |
| 1.6.4 二叉树的遍历 .....     | 28 |
| 1.7 查找技术 .....         | 30 |
| 1.7.1 顺序查找 .....       | 30 |
| 1.7.2 二分法查找 .....      | 31 |
| 1.8 排序技术 .....         | 32 |

|            |                       |           |
|------------|-----------------------|-----------|
| 1.8.1      | 交换类排序法 .....          | 32        |
| 1.8.2      | 插入类排序法 .....          | 34        |
| 1.8.3      | 选择类排序法 .....          | 36        |
| 1.8.4      | 各种排序方法的比较 .....       | 37        |
|            | 典型考题分析 .....          | 38        |
|            | 过关必备 .....            | 41        |
| <b>第2章</b> | <b>程序设计基础 .....</b>   | <b>44</b> |
|            | 考情回顾 .....            | 44        |
| 2.1        | 程序设计方法与风格 .....       | 44        |
| 2.2        | 结构化程序设计 .....         | 46        |
| 2.2.1      | 结构化程序设计的原则 .....      | 46        |
| 2.2.2      | 结构化程序的基本结构与特点 .....   | 46        |
| 2.2.3      | 结构化程序设计原则和方法的应用 ..... | 47        |
| 2.3        | 面向对象的程序设计 .....       | 48        |
| 2.3.1      | 关于面向对象方法 .....        | 48        |
| 2.3.2      | 面向对象方法的基本概念 .....     | 49        |
|            | 典型考题分析 .....          | 52        |
|            | 过关必备 .....            | 54        |
| <b>第3章</b> | <b>软件工程基础 .....</b>   | <b>56</b> |
|            | 考情回顾 .....            | 56        |
| 3.1        | 软件工程基本概念 .....        | 56        |
| 3.1.1      | 软件定义与软件特点 .....       | 56        |
| 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        |
| 3.1.4      | 软件工程的目标和原则 .....      | 61        |
| 3.1.5      | 软件开发工具与软件开发环境 .....   | 62        |
| 3.2        | 结构化分析方法 .....         | 63        |
| 3.2.1      | 需求分析和需求分析方法 .....     | 63        |
| 3.2.2      | 结构化分析方法 .....         | 64        |
| 3.2.3      | 软件需求规格说明书 .....       | 68        |
| 3.3        | 结构化设计方法 .....         | 69        |
| 3.3.1      | 软件设计的基本概念 .....       | 69        |
| 3.3.2      | 概要设计 .....            | 72        |
| 3.3.3      | 详细设计 .....            | 75        |
| 3.4        | 软件测试 .....            | 77        |
| 3.4.1      | 软件测试的目的 .....         | 78        |
| 3.4.2      | 软件测试的准则 .....         | 78        |
| 3.4.3      | 软件测试技术和方法综述 .....     | 78        |
| 3.4.4      | 软件测试的实施 .....         | 81        |

|                            |         |
|----------------------------|---------|
| 3.5 程序的调试 .....            | 83      |
| 3.5.1 基本概念 .....           | 83      |
| 3.5.2 软件调试方法 .....         | 85      |
| 典型考题分析 .....               | 86      |
| 过关必备 .....                 | 91      |
| <br>第4章 数据库设计基础 .....      | <br>93  |
| 考情回顾 .....                 | 93      |
| 4.1 数据库系统的基本概念 .....       | 93      |
| 4.1.1 数据、数据库、数据库管理系统 ..... | 93      |
| 4.1.2 数据库系统的发展 .....       | 95      |
| 4.1.3 数据库系统的基本特点 .....     | 96      |
| 4.1.4 数据库系统的内部结构体系 .....   | 97      |
| 4.2 数据模型 .....             | 98      |
| 4.2.1 数据模型的基本概念 .....      | 98      |
| 4.2.2 E-R 模型 .....         | 99      |
| 4.2.3 层次模型 .....           | 101     |
| 4.2.4 网状模型 .....           | 101     |
| 4.2.5 关系模型 .....           | 102     |
| 4.3 关系代数 .....             | 104     |
| 4.3.1 传统的集合运算 .....        | 104     |
| 4.3.2 专门的关系运算 .....        | 108     |
| 4.3.3 关系数据库规范化理论 .....     | 113     |
| 4.4 数据库设计与管理 .....         | 115     |
| 4.4.1 数据库设计概述 .....        | 115     |
| 4.4.2 数据库设计的需求分析 .....     | 116     |
| 4.4.3 数据库概念设计 .....        | 117     |
| 4.4.4 数据库的逻辑设计 .....       | 118     |
| 4.4.5 数据库的物理设计 .....       | 120     |
| 4.4.6 数据库管理 .....          | 121     |
| 典型考题分析 .....               | 121     |
| 过关必备 .....                 | 126     |
| <br>第5章 样题及答案解析 .....      | <br>129 |
| 5.1 样题 .....               | 129     |
| 5.1.1 样题一 .....            | 129     |
| 5.1.2 样题二 .....            | 130     |
| 5.1.3 样题三 .....            | 131     |
| 5.1.4 样题四 .....            | 132     |
| 5.1.5 样题五 .....            | 133     |
| 5.1.6 样题六 .....            | 134     |



|        |               |            |
|--------|---------------|------------|
| 5.1.7  | 样题七           | 135        |
| 5.1.8  | 样题八           | 135        |
| 5.1.9  | 样题九           | 136        |
| 5.1.10 | 样题十           | 137        |
| 5.1.11 | 样题十一          | 138        |
| 5.1.12 | 样题十二          | 139        |
| 5.1.13 | 样题十三          | 140        |
| 5.1.14 | 样题十四          | 141        |
| 5.1.15 | 样题十五          | 142        |
| 5.1.16 | 样题十六          | 143        |
| 5.1.17 | 样题十七          | 144        |
| 5.1.18 | 样题十八          | 146        |
| 5.1.19 | 样题十九          | 147        |
| 5.1.20 | 样题二十          | 148        |
| 5.2    | 样题答案与解析       | 149        |
| 5.2.1  | 样题一答案与解析      | 149        |
| 5.2.2  | 样题二答案与解析      | 150        |
| 5.2.3  | 样题三答案与解析      | 151        |
| 5.2.4  | 样题四答案与解析      | 152        |
| 5.2.5  | 样题五答案与解析      | 153        |
| 5.2.6  | 样题六答案与解析      | 154        |
| 5.2.7  | 样题七答案与解析      | 155        |
| 5.2.8  | 样题八答案与解析      | 156        |
| 5.2.9  | 样题九答案与解析      | 157        |
| 5.2.10 | 样题十答案与解析      | 158        |
| 5.2.11 | 样题十一答案与解析     | 159        |
| 5.2.12 | 样题十二答案与解析     | 160        |
| 5.2.13 | 样题十三答案与解析     | 161        |
| 5.2.14 | 样题十四答案与解析     | 162        |
| 5.2.15 | 样题十五答案与解析     | 163        |
| 5.2.16 | 样题十六答案与解析     | 164        |
| 5.2.17 | 样题十七答案与解析     | 165        |
| 5.2.18 | 样题十八答案与解析     | 166        |
| 5.2.19 | 样题十九答案与解析     | 167        |
| 5.2.20 | 样题二十答案与解析     | 168        |
|        | <b>本书习题答案</b> | <b>170</b> |
|        | 第1章           | 170        |
|        | 第2章           | 170        |
|        | 第3章           | 170        |
|        | 第4章           | 171        |

# 第 1 章 数据结构与算法

## 考情回顾

表 1-0 统计了最近 5 次与本章考点相关的试卷分析。

表 1-0 历年考题知识点分布统计表

| 年份      | 试题分布 | 题型  | 分值 | 考核要点                                                  |
|---------|------|-----|----|-------------------------------------------------------|
| 2011.09 | 1-3  | 选择题 | 6  | 栈和队列的概念、数据结构、二叉树的性质、各类排序                              |
|         | 1-2  | 填空题 | 4  |                                                       |
| 2011.03 | 1-3  | 选择题 | 6  | 栈的基本运算、非线性数据结构、二叉树的性质、二分法查找、二叉树的前序遍历、二叉树的后序遍历         |
|         | 1-2  | 填空题 | 4  |                                                       |
| 2010.09 | 1-2  | 选择题 | 4  | 顺序、链式存储结构、栈的基本运算、线性表                                  |
|         | 1-3  | 填空题 | 6  |                                                       |
| 2010.03 | 1-3  | 选择题 | 6  | 栈的基本运算、非线性数据结构、线性数据结构、二叉树的深度计算、二叉树的前序遍历、二叉树的中序遍历、二分查找 |
|         | 1-2  | 填空题 | 4  |                                                       |
| 2009.09 | 1-4  | 选择题 | 8  | 非线性数据结构、循环队列、栈的概念、算法的空间复杂度、二叉树的节点计算                   |
|         | 1    | 填空题 | 2  |                                                       |

## 1.1 算 法

考核说明：本章节内容比较简单，只要对算法的概念及其复杂度做一般性了解即可。

### 1.1.1 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，它不等于程序，也不等于计算机方法，程序的编制不可能优于算法的设计。算法是指令的有限序列，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

算法的基本特性是一组严谨地定义运算顺序的规则，每一个规则都是有效的、明确的，此顺序将在有限的次数内终止。

##### （1）可行性

算法的可行性是指算法中指定的操作都可以通过基本运算执行有限的次数后实现。在设计一个

算法时必须要考虑它的可行性，否则是不会得到满意结果的。

### （2）确定性

算法的确定性是指算法中的每一步骤都必须是有明确定义的，不允许有模棱两可的解释，也不允许有多义性。这一性质也反映了算法与数学公式的明显差别。在解决实际问题时，可能会出现这样的情况：针对某种特殊问题，数学公式是正确的，但按此数学公式设计的计算过程可能会使计算机无所适从。这是因为根据数学公式设计的计算过程只考虑了正常使用的情况，而当出现异常情况时，此计算过程就不能适应了。

### （3）有穷性

算法的有穷性是指算法必须能在有限时间内做完，即算法必须能在执行有限的步骤之后终止。数学中的无穷级数在实际计算时只能取有限项，即计算无穷级数值的过程只能是有穷的。因此，一个数的无穷级数表示只是一个计算公式，而根据精度要求确定的计算过程才是有穷的算法。

另外，算法的有穷性还应包括合理的执行时间的含义。因为，如果一个算法需要执行千万年，则显然失去了实用价值。

### （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，它可以有多个输入，也可以不要输入，但必须有一个或多个输出，不同的输入将会有不同的输出结果。一个算法是否有效，还取决于为算法所提供的情报是否足够。通常情况下，当情报足够时，算法才是有效的，当情报不够时，算法可能无效。

综上所述，所谓算法，就是一组严谨地定义运算顺序的规则，并且每一个规则都是有效的、明确的，且此顺序将在有限的次数内终止。

## 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）算法中对数据对象的运算和操作

每个算法实际上是按照解题要求从环境能进行的所有操作中选择合适的操作所组成的一组指令序列。因此，计算机算法就是计算机能处理的操作所组成的指令序列。

通常，计算机可以执行的基本操作是以指令形式描述的。一个计算机系统能执行的所有指令的集合，称为该计算机系统的指令系统。计算机程序就是按解题要求从计算机指令系统中选择合适的指令所组成的指令序列。在一般的计算机系统中，基本的运算和操作包括以下4类：

- 算术运算：主要包括加、减、乘、除等运算。
- 逻辑运算：主要包括“与”、“或”、“非”等运算。
- 关系运算：主要包括“大于”、“小于”、“等于”和“不等于”等运算。
- 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用的操作，而且还与各操作之间的执行顺序有关。

算法的控制结构给出了算法的基本框架，它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。算法通常可以用顺序、选择和循环3种基本控制结构组合而成，而它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

列举法是计算机算法中的一个基础算法，此方法是根据提出的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。因此，列举法常用于解决“是否存在”或“有多少种可能”等类型的问题。

列举法的特点是算法比较简单。但当列举的可能情况较多时，执行列举算法的工作量将会很大。因此，需重点注意的是，在用列举法设计算法时，应尽量使方案优化，减少运算工作量。

列举法虽然是一种比较笨拙且原始的方法，其运算量比较大，但在有些实际问题中（如查找、搜索、寻找捷径等问题），局部使用列举法却是很有有效的，因此，列举法是计算机算法中的一种基础算法。

#### (2) 归纳法

归纳法的基本思想是，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。从本质上讲，归纳就是通过观察一些简单而特殊的情况，最后总结出一般性的结论。

显然，归纳法要比列举法更能反映问题的本质，并且可以解决列举量为无限的问题。但是，从一个实际问题中总结归纳出一般的关系并不是一件容易的事情，尤其是要归纳出一个数学模型更为困难。

#### (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。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。递推关系通常是归纳的结果，递推法就其本质来说也属于归纳法，此方法还经常在数值计算中使用。

#### (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，最后归结为一些简单的问题，逐个解决这些简单的问题，然后再沿原来的逆过程逐步进行综合，这就是递归的基本思想。由此可以看出，递归的基础也是归纳。在工程实际中，有许多问题就是用递归来定义的，数学中的许多函数也是用递归来定义的。递归在可计算性理论和算法设计中占有很重要的地位。

#### (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，而“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，其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，也不能无限制地列举，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，若试探成功，就得到问题的解，否则，就逐步回退，换线索再逐步试探。回溯法在处理复杂数据结构方面有着广泛的应用。

### 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) 健壮性：当输入不合法数据时，应能作适当处理，不致引起严重后果。

- (3) 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。
- (4) 高效性：有效使用存储空间和有较高的时间效率。

### 1.1.2 算法复杂度

算法的复杂度主要包括时间复杂度和空间复杂度。

#### 1. 算法的时间复杂度

所谓算法的时间复杂度（Time Complexity）是指执行算法所需要的计算工作量。

为了能够比较客观地反映出一个算法的效率，在度量一个算法的工作量时，不仅与所使用的计算机、程序设计语言以及程序编制者无关，而且还应与算法实现过程中的许多细节无关。为此，可以用算法在执行过程中所需基本运算的执行次数来度量算法的工作量。基本运算反映了算法运算的主要特征，因此，用基本运算次数来度量算法工作量是客观的也是实际可行的，有利于比较同一问题的几种算法的优劣。例如，在考虑两个矩阵相乘时，可以将两个实数之间的乘法运算作为基本运算，而对于所用的加法（或减法）运算忽略不计。又如，当需要在一个表中进行查找时，可以将两个元素之间的比较作为基本运算。

另外，算法所执行的基本运算次数还与问题的规模有关。例如，两个 20 阶矩阵相乘与两个 10 阶矩阵相乘，所需要的基本运算次数显然是不同的，前者需要更多的运算次数。因此，在分析算法的工作量时，还必须对问题的规模进行度量。

综上所述，算法的工作量用算法所执行的基本运算次数来度量，而算法所执行的基本运算次数是问题规模的函数，即：

$$\text{算法的工作量} = f(n)$$

通常记作：

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

它表示随着问题规模  $n$  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：

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



#### 真题链接

【例 1-1】算法的时间复杂度是指\_\_\_\_\_。(2011.03)

- 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

解析：本题考查的知识点是时间复杂度的概念。算法的时间复杂度是指算法在执行过程中所需要的基本运算次数。

答案：D

#### 2. 算法的空间复杂度

一个算法的空间复杂度就是执行这个算法所需要的内存空间。

一个算法所占用的存储空间包括算法程序所占的空间、输入的初始数据所占的存储空间以及算

法执行过程中所需要的额外空间。其中额外空间包括算法程序执行过程中的工作单元以及某种数据结构所需要的附加存储空间（例如，在链式结构中，除了需要存储数据本身之外，还需要存储链接信息）。如果额外空间量相对于问题规模来说是常数，则称该算法是原地（in place）工作的。

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



### 真题链接

【例 1-2】下列叙述中正确的是\_\_\_\_\_。（2011.09）

- 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

解析：所谓算法是指解题方案的准确而完整的描述。但算法不等于程序，也不等于计算方法。故 A 项错误。设计算法时要考虑可行性、确定性、有穷性和拥有足够的情报。B、C 皆错误，故选 D。

答案：D

【例 1-3】下列叙述中正确的是\_\_\_\_\_。（2010.09）

- 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

解析：算法在运行过程中所需要的辅助存储空间的大小称为算法空间复杂度；算法的时间复杂度是执行该算法所需要的计算工作量，即算法执行过程中所需要的基本运算次数。为了能够比较客观地反映出一个算法的效率，在度量一个算法的工作量时，与所使用的计算机、程序设计语言以及程序编制者无关，而且还与算法实现过程中的许多细节无关，但可以用算法在执行过程所需基本运算的执行次数来度量算法的工作量。

答案：D

## 1.2 数据结构的基本概念



**考核说明：**本节介绍数据结构的基本概念，不是重点考核内容，对线性结构和非线性结构稍作了解即可。

### 1.2.1 什么是数据结构

#### 1. 数据结构的定义

计算机已经被广泛用于数据处理。数据结构（Data Structure）是指反映数据元素之间关系的数据元素集合的表示，其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）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提高数据处理的效率，主要包括以下两个方面：

- （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。在计算机科学中，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、节点、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

数据结构是指反映数据元素之间的关系的的数据元素集合的表示，即带有结构的数据元素的集合。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以下两个方面的信息：

- （1）表示数据元素的信息；
- （2）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。所谓数据的逻辑结构，是指反映数据元素之间逻辑关系的数据结构。根据数据元素之间关系的不同特性，通常有 4 类基本的逻辑结构，如图 1-1 所示。

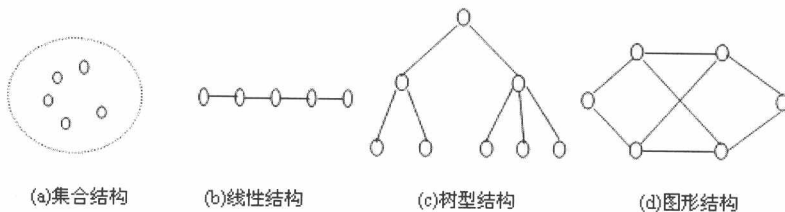


图 1-1 数据的逻辑结构

（1）集合结构：在集合结构中，数据元素之间的关系是“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）树型结构：数据元素之间存在着一对多的关系。

（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成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。数据的逻辑结构在计算机存储空间中的存放形式称为数据的物理结构，或称数据的存储结构。

由于数据元素在计算机存储空间中的位置关系可能与逻辑关系不同，因此，为了表示存放在计算机存储空间中的各数据元素之间的逻辑关系（即前后件关系），在数据的存储结构中，不仅要存放

各数据元素的信息，还需要存放各数据元素之间的前后件关系的信息。

一般来说，一种数据的逻辑结构根据需要可以表示成多种存储结构。

(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，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，该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。

## 1.2.2 数据结构的图形表示

### 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  $D$  中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，通常称为数据节点，并简称为节点；对于关系  $R$  中的每个二元组，用一条有向线段从前件节点（或称为前驱节点）指向后件节点（或称为后继节点）。例如，上例中的家庭成员数据结构也可以用图 1-2 表示。

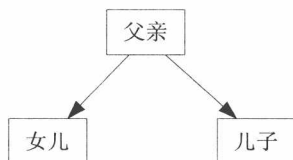


图 1-2 数据结构的图形表示

在数据结构中，没有前件的节点称为根节点，没有后件的节点称为终端节点（或叶子节点）。一个数据结构中的节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节点（称为插入运算），或删除某个节点（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。

### 1.2.3 线性结构与非线性结构

如果在一个数据结构中一个数据元素都没有，则称该数据结构为空的数据结构。在一个空的数据结构中插入一个新的元素后就变为非空的数据结构；在只有一个数据元素的数据结构中，将该元素删除后就变为空的数据结构。

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，则称该数据结构为线性结构，又称线性表。

- (1) 有且只有一个根节点；
- (2) 每个节点最多有一个直接前件，也最多有一个直接后件。

特别需要说明的是，在一个线性结构中插入或删除任何一个节点后还应是线性结构，否则，不能称为线性结构。

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，一个节点可以有多个直接后件，或者有多个直接前件；或者既有多个直接后件又有多个直接前件。树型结构和图形结构都是非线性结构。

线性结构和非线性结构都可以是空的数据结构。一个空的数据结构究竟是属于线性结构还是属于非线性结构，这要根据具体情况来确定。如果对该数据结构运算是按线性结构的规则来处理的，则属于线性结构，否则属于非线性结构。



#### 真题链接

【例 1-4】下列叙述中正确的是\_\_\_\_\_。(2011.03)

- A. 有一个以上根节点的数据结构不一定是非线性结构
- B. 只有一个根节点的数据结构不一定是线性结构
- C. 循环链表是非线性结构
- D. 双向链表是非线性结构

解析：本题考查的知识点是数据结构。线性表的链式存储结构称为线性链表，是线性结构，而循环链表和双向链表都是在线性链表的基础上演化而来的，因此，选项 C、D 错误。线性结构必须有且只有一个根节点，因此选 B。

答案：B

【例 1-5】数据结构分为线性结构和非线性结构，带链的栈属于\_\_\_\_\_。(2011.09)

解析：对于数据结构课程而言，简单地说，线性结构是  $n$  个数据元素的有序（次序）集合。数据结构中线性结构指的是数据元素之间存在着“一对一”的线性关系的数据结构。相对应于线性结构，非线性结构的逻辑特征是一个结点元素可能对应多个直接前驱和多个后驱。常用的线性结构有：线性表、栈、队列、双队列、数组、串。

答案：线性结构

## 1.3 线性表及其顺序存储结构



**考核说明：**本节考查的是线性表和顺序存储结构的概念，这些概念只在 2009 年考过一次，考到的概率较小。