

2011

新教程版



National Computer
Rank Examination

全国计算机等级考试

全真笔试+上机考题解答与训练

二级公共基础知识和
C 语言程序设计

课程详解+真题研密+分块训练
模拟演练+上机题库+学习软件



新版本 新特点

1000余题，题后有详解，90%以上高命中率
知识讲解深入全面，可当教程使用
近4年历年题按考点归类，专家权威讲解
考点、难点、重点、疑点全面剖析



专注研究等考10年



新思路教育科技研究中心
电子科技大学出版社

NEWIDEAS

2011

新教程版

National Computer
Rank Examination

全国计算机等级考试

全真笔试+上机考题解答与训练

二级公共基础知识和
C语言程序设计

课程详解+真题研密+分块训练
模拟演练+上机题库+学习软件



專業研究等考10年

新思路教育科技研究中心
电子科技大学出版社

◀ 内 容 简 介 ▶

为了适应计算机等级考试的需求,帮助考生顺利通过考试,我们总结了多年考试经验和众多图书的优点,精心策划编写了本书。

本书根据最新考试大纲要求,依据近4年考试试题的研究成果和对命题规律的把握,结合历年真题讲解各考点,知识点讲解详细,重、难点突出,真切、实用;学习不跑题、切中要点!另外,本书所有试题的答案及解析均由一线老师根据实践教学中学得经验编写,注重教授考生解题思路、方法和技巧,帮助考生系统全面的掌握各个笔试及上机考点。

本书既可以作为计算机等级考试考生的自学用书,也可以作为计算机等级考试培训班的教学参考书和辅导用书。

图书在版编目(CIP)数据

上机考题解答与训练. 二级公共基础知识和C语言程序设计 / 王丹主编. -- 成都: 电子科技大学出版社, 2010.11

(全国计算机等级考试全真笔试)

ISBN 978-7-5647-0691-3

I. ①上… II. ①王… III. ①电子计算机—水平考试—解题②C语言—程序设计—水平考试—解题 IV. ①TP3-44

中国版本图书馆CIP数据核字(2010)第212633号

全国计算机等级考试全真笔试+上机考题解答与训练

二级公共基础知识和C语言程序设计

全国计算机等级考试命题研究组 编
新思路教育科技研究中心

出 版: 电子科技大学出版社(成都市一环路东一段159号电子信息产业大厦 邮编: 610051)

策划编辑: 张 鹏

责任编辑: 张 鹏

主 页: www.uestcp.com.cn

电子邮箱: uestcp@uestcp.com.cn

发 行: 新华书店经销

印 刷: 武邑县印刷厂

成品尺寸: 210mm×285mm 印张 15.75 字数 403.2 千字

版 次: 2010年11月第一版

印 次: 2010年11月第一次印刷

书 号: ISBN 978-7-5647-0691-3

定 价: 28.80 元

版权所有 侵权必究

本书如有缺页、破损、装订错误,请寄回印刷厂调换。

邮购热线: 010-62165518 010-59796026

前言

考点即等级考试考查的知识点，抓住考点是考试成功的保证，也是考试成功的捷径。如果按照传统的方式一点一点地啃教材，势必会花很多的时间和精力。甚至很容易引起疲劳感，最终导致很多考生都不能坚持下去。

为了帮助更多的考生顺利地通过考试，我们在深入研究、全面解读考试大纲的基础上，组织有着丰富的计算机等级考试辅导教学经验的一线教师编写了本书。

♣ 知识点全面，考点讲解详细

本书把原本繁杂的知识分成一个考点一个考点地去讲解，条理清晰、目标明确。除此之外，考点分等级，各知识点讲解详细，相当于一本精简版教程，对于有可能考核的内容做到一网打尽。

♣ 分析历年真卷，科学归纳考点

考点的判断依据4年来历年试题的研究成果和对命题规律的把握而得，所有判断均有具体数据支持。如果你初次接触等级考试，对“考什么”、“如何学”都一无所知。本书就是直接地告诉你这些重要的信息，且分清哪些是重要考点、哪些是必考点……同时拿出研究数据来，让你真正信服我们的这些判断。

♣ 真题研密，分块训练

本书所有考点都配有同步历年真题及模拟试题的讲解和训练，且试题的答案及解析均由一线老师根据实践教学中所经验编写，注重教授考生解题思路、方法和技巧，让你通过做历年真题和模拟试题快速掌握各知识点及答题技巧。

♣ 笔试、上机，综合模拟

本书最后两章分别安排了数套笔试模拟试卷和上机考试试题。

笔试模拟试卷：结合最新考试大纲，筛选、组编了4套命中率高的模拟试卷，不论在形式和难度上，都与真题类似，解析详尽、透彻。

上机考试试题：通过做上机考试试题，了解上机考试的题型，且每套试题都有详细的解题步骤，手把手教你操作完成每个上机试题。

♣ 配套模拟软件

本书配有强大的考试模拟软件练习光盘，具有如下特点：

- 含多套的笔试模拟试卷和大量的上机模拟试题，让你既可进行上机模拟考试、练习，同时也可以练习笔试考试。
- 登录、抽题、答题、提交与正式上机考试一样，并提供详细的答题步骤和标准答案。同时，还可以自动生成试卷、自动计时、自动评分。

由于时间仓促，书中难免有不当之处，敬请指出！

全国计算机等级考试命题研究组
新思路教育科技有限公司

编委会

主 编：王 丹

编委组：（排名不分先后）

白 杰	高强华	孟令全	付江辉
王 毅	房向阳	李 婧	刘 明
赵江静	韩雪松	安玉彦	张 龙
刘永阳	马天宝	刘艳飞	柳寒冰
唐保存	黄 和	兰天静	董 昶
黄琴华	王艳梅	金 利	刘本发
齐文斌	迟 剑	李爱菊	张铁军

责任编辑：张 鹏

编校组：（排名不分先后）

赵 波	李 润	周 雪	丁 建
丁力刚	杨宏舟	肖维萍	李 欣
梁 继	章金博	范 伟	李 瑞
胡昭昀	孟祥萍	张 宇	李金娜

新版考试大纲剖析

一、基本要求

- ① 熟悉 Visual C++ 6.0 集成开发环境。
- ② 掌握结构化程序设计的方法，具有良好的程序设计风格。
- ③ 掌握程序设计中简单的数据结构和算法并能阅读简单的程序。
- ④ 在 Visual C++ 6.0 集成环境下，能够编写简单的 C 程序，并具有基本的纠错和调试程序的能力。

二、考试内容

(1) C 语言的结构

大纲要求	考点点评及考核情况
<ol style="list-style-type: none">① 程序的构成，main 函数和其他函数。② 头文件，数据说明，函数的开始和结束标志以及程序中的注释。③ 源程序的书写格式。④ C 语言的风格。	<p>考点点评：主要识记各考点的基本内容，笔试中一般以考核概念性的知识为主。笔试考核中属于比较简单的内容，一般出现在前 10 个选择题中。</p> <p>考核情况： 笔试：所占分值在 2 分左右。 上机：改错题中经常出现，抽中几率为 7%。</p>

(2) 数据类型及其运算

大纲要求	考点点评及考核情况
<ol style="list-style-type: none">① C 的数据类型（基本类型，构造类型，指针类型，无值类型）及其定义方法。② C 运算符的种类、运算优先级和结合性。③ 不同类型数据间的转换与运算。④ C 表达式类型（赋值表达式，算术表达式，关系表达式，逻辑表达式，条件表达式，逗号表达式）和求值规则。	<p>考点点评：属于基础知识，相对来说比较简单，对于自加、自减运算需要大家了解掌握。上机、笔试中均有体现，是上机考试重点。</p> <p>考核情况： 笔试：所占分值在 6 分左右。 上机：3 种题型均有体现，经常结合及其他知识点综合考查，抽中几率为 55%。</p>

(3) 基本语句

大纲要求	考点点评及考核情况
<ol style="list-style-type: none">① 表达式语句，空语句，复合语句。② 输入输出函数的调用，正确输入数据并正确设计输出格式。	<p>考点点评：主要考核格式输入函数 printf() 和格式输出函数 scanf()。这两个知识点一般不单独考核，常与其他知识点综合起来进行考查。笔试中很少考核，重点是上机。</p> <p>考核情况： 笔试：所占分值不超过 2 分左右。 上机：3 种题型均有体现，抽中几率为 35%。</p>

(4) 选择结构程序设计

大纲要求	考点点评及考核情况
<ol style="list-style-type: none">① 用 if 语句实现选择结构。② 用 switch 语句实现多分支选择结构	<p>考点点评：属于简单程序结构，需要重点掌握分支结构的判断条件和执行顺序。另</p>

③ 选择结构的嵌套。

外,选择结构的判断条件一般由关系运算和逻辑运算构成,因此这两个知识点也是本章的学习重点。此部分知识点都是历年考题的重点,一般结合其它知识点综合考核。

考核情况:

笔试:所占分值在8分左右。

上机:3种题型均有体现,抽中几率为35%。

(5) 循环结构程序设计

大纲要求	考点点评及考核情况
① for 循环结构。 ② while 和 do-while 循环结构。 ③ continue 语句和 break 语句。 ④ 循环的嵌套。	<p>考点点评:考核的重中之重,在考试中所占分值很高,考核形式也是多样化。另外,对于 break 语句和 continue 语句比较简单,考生只需了解即可。</p> <p>考核情况:</p> <p>笔试:笔试考试中是必考点,所占分值在18分左右。</p> <p>上机:在上机考试中是必考点,应用性很强。3种题型均有体现,抽中几率为40%。</p>

(6) 数组的定义和引用

大纲要求	考点点评及考核情况
① 一维数组和二维数组的定义、初始化和数组元素的引用。 ② 字符串与字符数组。	<p>考点点评:数组是一个由若干同类型变量组成的集合。按数组元素的类型不同,数组又可分为数值数组、字符数组、指针数组、结构数组等各种类别。都需要考生重点掌握。无论在笔试还是上机考试中,都占有很大的比率。</p> <p>考核情况:</p> <p>笔试:所占分值在5分左右。</p> <p>上机:上机考试中的考核频率也越来越高,3种题型均有体现,抽中几率为35%。</p>

(7) 函数

大纲要求	考点点评及考核情况
① 库函数的正确调用。 ② 函数的定义方法。 ③ 函数的类型和返回值。 ④ 形式参数与实在参数,参数值的传递。 ⑤ 函数的正确调用,嵌套调用,递归调用。 ⑥ 局部变量和全局变量。	<p>考点点评:对于主函数、局部变量、全局变量、外部函数和静态函数等知识,考生应注意掌握。重点内容是程序的模块化设计最直接的实现方式就是采用函数的方式。</p> <p>考核情况:</p> <p>笔试:所占分值不超过10分。</p>

⑦ 变量的存储类别（自动，静态，寄存器，外部），变量的作用域和生存期。

上机：对变量的定义及自加、自减运算的考核频率比较高，应熟练掌握变量的定义和自加、自减运算。为高频考题，抽中几率为40%。

(8) 编译预处理

大纲要求	考点点评及考核情况
① 宏定义和调用（不带参数的宏，带参数的宏）。 ② “文件包含”处理。	考点点评： 考查的重点是宏定义，其中，不带参数和带参数的宏定义不好理解。 考核情况： 笔试：笔试题所占分值在6分左右。 上机：一般考核不带参数和带参数的宏定义，考试频率低，抽中几率为在2%。

(9) 指针

大纲要求	考点点评及考核情况
① 地址与指针变量的概念，地址运算符与间址运算符。 ② 一维、二维数组和字符串的地址以及指向变量、数组、字符串、函数、结构体的指针变量的定义。通过指针引用以上各类型数据。 ③ 用指针作函数参数。 ④ 返回地址值的函数。 ⑤ 指针数组，指向指针的指针。	考点点评： 指针是一种数据类型。掌握指针型数据的使用，是深入理解C语言特性和掌握C语言编程技巧的重要环节。正确并灵活地运用指针，可有效地描述各种数据结构，能够动态地分配内存空间，能够方便地操作字符串，还可在函数之间传递各种类型的数据，以便提高使程序简洁性、紧凑性、执行效率。 笔试和上机考试的重点，考核频率也越来越高，考生应给以足够的重视。 考核情况： 笔试：所占分值一般在15分左右。 上机：3种题型均有体现，抽中几率为35%。

(10) 结构体（即“结构”）与共用体（即“联合”）

大纲要求	考点点评及考核情况
① 用 typedef 说明一个新类型。 ② 结构体和共用体类型数据的定义和成员的引用。 ③ 通过结构体构成链表，单向链表的建立，结点数据的输出、删除与插入。	考点点评： 本部分知识点属于C语言程序设计提高部分，要求考生重点掌握结构体、共用体和用户定义类型的概念及应用。结构体是必考知识点。 考核情况： 笔试：所占分值在8分左右。 上机：3种题型均有体现，考核几率很高，一般出现在程序设计题中，抽题几率为7%。

(11) 位运算

大纲要求	考点点评及考核情况
① 位运算符的含义和使用。 ② 简单的位运算。	考点点评： 了解位运算符的基本概念及使用，掌握简单的位运算即可。此部分知识点多出现在笔试考核，上机试题中一般不会出现。 考核情况： 笔试：所占分值为 2 分。 上机：—

(12) 文件操作

大纲要求	考点点评及考核情况
只要求缓冲文件系统（即高级磁盘 I/O 系统），对非标准缓冲文件系统（即低级磁盘 I/O 系统）不要求。 ① 文件类型指针（FILE 类型指针）。 ② 文件的打开与关闭（fopen, fclose）。 ③ 文件的读写（fputc, fgetc, fputs, fgets, fread, fwrite, fprintf, fscanf 函数的应用），文件的定位（rewind, fseek 函数的应用）。	考点点评： 此部分知识属于 C 语言程序设计提高部分，相对较难。考生应在识记文件指针概念的基础上，理解地掌握文件的读写操作。另外，对于文件的定位也不能忽视。笔试考试和上机考试中考核的重点都是文件的读写。 考核情况： 笔试：所占分值在 6 分左右。 上机：多以填空题和改错题出现，抽题几率非常低。

三、考试方式

(1) 笔试

90 分钟，满分 100 分，其中含公共基础知识部分的 30 分。

(2) 上机操作

90 分钟，满分 100 分。上机操作包括：①填空题 ②改错题 ③编程题。

目 录

第 1 章 公共基础知识

1.1 考试必备知识	2
1.1.1 数据库结构与算法	2
1.1.2 程序设计基础	7
1.1.3 软件工程基础	9
1.1.4 数据库设计基础	13
1.2 历年试题汇编	18
1.2.1 历年试题	18
1.2.2 参考答案	22
1.3 全真模拟试题汇编	27
1.3.1 模拟试题	27
1.3.2 参考答案	28

第 2 章 C 语言的基本知识

2.1 C 语言基本概念	32
2.1.1 C 程序和程序设计	32
2.1.2 程序的构成、main 函数和其他函数	33
2.1.3 程序的注释、书写格式及 C 语言的风格	34
2.2 数据类型及其运算	34
2.2.1 标识符、常量和变量	34
2.2.2 整型数据	35
2.2.3 实型常量及变量	37
2.2.4 字符型常量及字符变量	37
2.2.5 算术表达式	39
2.2.6 赋值表达式	40
2.2.7 自加运算符、自减运算符和逗号运算符	40
2.2.8 关系运算符和关系表达式	41
2.2.9 逻辑运算符和逻辑表达式	42
2.3 历年试题汇编	42
2.3.1 历年试题	42
2.3.2 参考答案	45
2.4 全真模拟试题汇编	46
2.4.1 模拟试题	46
2.4.2 参考答案	49

第 3 章 C 语言程序设计

3.1 顺序结构程序设计	54
3.1.1 赋值语句	54
3.1.2 数据输出	54
3.1.3 数据输入	55
3.1.4 复合语句和空语句	56
3.2 选择结构程序设计	57
3.2.1 if 语句和用 if 语句构成的选择	57
3.2.2 switch 语句以及用 switch 语句和 break 语句构成的选择结构	59
3.2.3 条件表达式构成的选择结构	60

3.2.4 语句标号和 goto 语句	61
3.3 循环结构程序设计	62
3.3.1 while 语句和用 while 语句构成的循环结构	62
3.3.2 do-while 语句和用 do-while 语句构成的循环结构	63
3.3.3 for 语句和用 for 语句构成的循环结构	64
3.3.4 循环结构的嵌套	64
3.3.5 break 语句和 continue 语句	66
3.4 历年试题汇编	67
3.4.1 历年试题	67
3.4.2 参考答案	75
3.5 全真模拟试题汇编	78
3.5.1 模拟试题	78
3.5.2 参考答案	86

第 4 章 函数与数组

4.1 函 数	92
4.1.1 库函数的调用	92
4.1.2 函数的定义和返回值	92
4.1.3 函数的调用及要求	93
4.1.4 函数的说明	93
4.1.5 调用函数和被调用函数之间的数据传递	94
4.1.6 对函数的进一步讨论	94
4.1.7 局部变量与全局变量	95
4.1.8 变量的作用域与生存期	95
4.1.9 函数的存储分类	97
4.2 数 组	97
4.2.1 一维数组的定义和一维数组元素的引用	97
4.2.2 二维数组的定义和二维数组元素的引用	99
4.2.3 字符串与字符数组	100
4.3 历年试题汇编	102
4.3.1 历年试题	102
4.3.2 参考答案	108
4.4 全真模拟试题汇编	110
4.4.1 模拟试题	110
4.4.2 参考答案	114

第 5 章 编译预处理和指针

5.1 编译预处理	118
5.1.1 宏替换	118
5.1.2 文件包含	119
5.2 动态存储分配	119
5.3 指 针	120
5.3.1 变量的地址和指针	120
5.3.2 指针变量的定义和指针变量的基类型	120
5.3.3 给指针变量赋值	121
5.3.4 对指针变量的操作	121
5.3.5 函数之间地址值的传递	123

5.4 历年试题汇编	125
5.4.1 历年试题	125
5.4.2 参考答案	129
5.5 全真模拟试题汇编	130
5.5.1 模拟试题	130
5.5.2 参考答案	135

第6章 结构体、共用体、位运算及文件操作

6.1 结构体	138
6.1.1 用 typedef 说明一种新类型	138
6.1.2 结构体类型	138
6.2 共用体	141
6.2.1 共用体类型的说明和变量定义	141
6.2.2 共用体变量的引用	142
6.3 位运算	142
6.3.1 位运算符	142
6.3.2 位运算符的运算功能	142
6.4 文件操作	144
6.4.1 文件的基本概念	144
6.4.2 文件的操作	144
6.4.3 文件操作函数	145
6.5 历年试题汇编	151
6.5.1 历年试题	151
6.5.2 参考答案	155
6.6 全真模拟试题汇编	157
6.6.1 模拟试题	157
6.6.2 参考答案	162

第7章 笔试模拟试卷

笔试模拟试卷(1)	165
笔试模拟试卷(2)	172
笔试模拟试卷(3)	178
笔试模拟试卷(4)	185
笔试模拟试卷(1) 参考答案及解析	192
笔试模拟试卷(2) 参考答案及解析	197
笔试模拟试卷(3) 参考答案及解析	200
笔试模拟试卷(4) 参考答案及解析	204

第8章 上机考试试题

第1套 上机考试试题	209
第2套 上机考试试题	211
第3套 上机考试试题	213
第4套 上机考试试题	215
第5套 上机考试试题	217
第6套 上机考试试题	220
第7套 上机考试试题	222
第8套 上机考试试题	225
第9套 上机考试试题	227
第10套 上机考试试题	230

第1套 上机考试试题答案及详解	233
第2套 上机考试试题答案及详解	234
第3套 上机考试试题答案及详解	235
第4套 上机考试试题答案及详解	235
第5套 上机考试试题答案及详解	236
第6套 上机考试试题答案及详解	237
第7套 上机考试试题答案及详解	238
第8套 上机考试试题答案及详解	239
第9套 上机考试试题答案及详解	240
第10套 上机考试试题答案及详解	241

第 1 章 公共基础知识

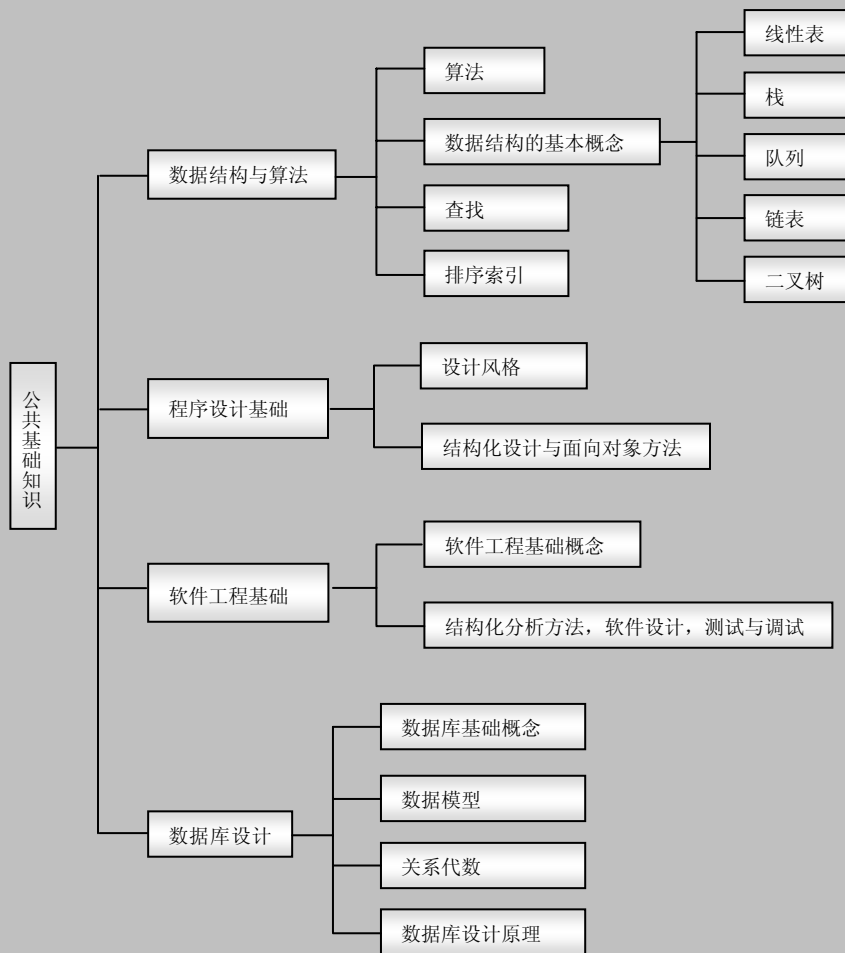
本章导读

本章介绍了公共基础知识的相关知识。本章内容在笔试题目中所占比例基本是固定的，共占 30 分。作为后面章节的学习基础，建议考生理解学习，以便更好地掌握本章的相关概念和知识点。

本章考试大纲

1. 数据结构与算法
2. 程序设计基础
3. 软件工程基础
4. 数据库基础

本章知识的逻辑结构图



1.1 考试必备知识

1.1.1 数据库结构与算法

1. 算法（基础，识记）

(1) 算法的基本概念

计算机算法为计算机解题的过程实际上是在实施某种算法。所谓算法是对解题方案准确而完善的描述。

- ①算法的基本特征：可行性、确定性、有穷性、拥有足够的情报。
- ②基本运算和操作包括：算术运算、逻辑运算、关系运算、数据传输。
- ③算法的 3 种基本控制结构：顺序结构、选择结构、循环结构。
- ④算法基本设计方法：列举法、归纳法、递推、递归、减半递推技术、回溯法。
 - 列举法：根据提出的问题，列举所有可能的情况，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；
 - 归纳法：通过列举少量特殊情况，经过分析，最后找出一般的关系；
 - 递推：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；
 - 递归：解决一些复杂问题时，为了降低问题的复杂程度等，一般总是将问题逐层分解，最后归结为一些简单的问题，再沿着原来分解的逆过程逐步进行综合。递归分为直接递归和间接递归两种；
 - 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。所谓“递推”是指重复“减半”过程；
 - 回溯法：通过对问题的分析，找出一个解决问题的线索，然后沿着这个线索逐步试探，对于每一步的试探，若试探成功，就得到问题的解；若试探失败，就逐步回退，回到别的路线再进行试探。

(2) 算法复杂度

算法复杂度包括时间复杂度和空间复杂度。详见表 1-1。

表 1-1 算法复杂度

名称	描述
时间复杂度	指执行算法所需要的计算工作量。算法的工作量可以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
空间复杂度	指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用的空间、输入的初始数据所占用的存储空间以及算法执行过程中所需要的额外空间

2. 数据结构的基本概念（基础，识记）

(1) 数据结构研究的 3 个方面

- ①数据集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。

(2) 数据结构

数据结构是指相互有关联的数据元素的集合。

①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，它可以用一个数据元素的集合和定义在此集合中的若干关系来表示。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，它反映了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，一般用二元组来表示。

②存储结构。

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存储结构（也称数据的物理结构）。

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，常用的有顺序、链式等存储结构。

- 顺序存储方式主要用于线性的数据结构，它把逻辑上相邻的数据元素存储在物理上相邻的存储单元中，结点之间的关系由存储单元的邻接关系来体现。
- 链式存储结构就是在每个结点中至少包含一个指针域，用指针来体现数据元素之间逻辑上的联系。

(3) 线性结构和非线性结构

根据数据结构中各数据元素之间前后件关系的复杂程度，一般将数据结构分为两大类型：线性结构与非线性结构。

①如果一个非空的数据结构满足下列两个条件：

- 有且只有一个根结点；
- 每一个结点最多有一个前件，也最多有一个后件。

则称该数据结构为线性结构。线性结构又称线性表。在一个线性结构中插入或删除任何一个结点后还应是线性结构。栈、队列、串等都是线性结构。

②如果一个数据结构不是线性结构，则称之为非线性结构。数组、广义表、树和图等数据结构都是非线性结构。

3. 线性表（基础，识记）

(1) 线性表的基本概念

线性表是由 $n(n \geq 0)$ 个数据元素 a_1, a_2, \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。

(2) 线性表的顺序存储结构的基本特点

- 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。

元素 a_i 的存储地址为： $ADR(a_i) = ADR(a_1) + (i-1)k$ ， $ADR(a_1)$ 为第一个元素的地址， k 代表每个元素占的字节数。

(3) 线性表的运算

线性表的运算有查找、插入、删除 3 种。

①线性表的插入。

- 插入的实现方法：要在第 $i(1 \leq i \leq n)$ 个元素之前插入一个新元素时，首先要从最后一个元素开始，直到第 i 个元素之间的 $n-i+1$ 个元素依次向后移动一个位置，移动结束后，第 i 个元素位置被空出，然后将新元素插入到第 i 项。插入结束后，线性表的长度就增加 1。
- 线性表插入操作算法的复杂度分析：由线性表插入过程可知，如果插入位置在第 $i(1 \leq i \leq n)$ 个元素之前，则原来第 i 个元素之后的所有元素都必须向后移动。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，最坏情况下要移动表中的所有元素。

②线性表的删除。

- 删除的实现方法：要删除第 $i(1 \leq i \leq n)$ 个元素时，首先要从第 $i+1$ 个元素开始，直到最后一个元素之间的 $n-i$ 个元素依次向前移动一个位置。删除结束后，线性表的长度就减少 1。
- 线性表删除操作算法的复杂度分析：如果删除的是第 $i(1 \leq i \leq n)$ 个元素，则原来第 i 个元素之后的所有元素都必须向前移动一个位置。在平均情况下，要在线性表中删除一个已有的元素，需要移动表中一半的元素，最坏情况下要移动表中的所有元素。

(4) 线性表顺序存储结构的适用场合

线性表的顺序存储结构对于小线性表或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。但是这种顺序存储结构对于元素需要变动的大线性表就不合适了，因为插入和删除的效率比较低。

4. 栈（基础，识记）

(1) 栈的基本概念

栈（stack）是一种特殊的线性表，是限定只在一端进行插入与删除的线性表。



在栈中，一端是封闭的，既不允许进行插入元素，也不允许删除元素；另一端是开口的，允许插入和删除元素。通常称插入、删除的这一端为栈顶，另一端为栈底。当表中没有元素时称为空栈。栈顶元素总是最后被插入的元素，从而也是最先被删除的元素；栈底元素总是最先被插入的元素，从而也是最后才能被删除的元素。

栈是按照“先进后出”或“后进先出”的原则组织数据的。例如，枪械的子弹匣就可以用来形象地表示栈结构。子弹匣的一端是完全封闭的，最后被压入弹匣的子弹总是最先被弹出，而最先被压入的子弹最后被弹出。

(2) 栈的顺序存储及其运算

栈的基本运算有三种：入栈、退栈与读栈顶元素。

①入栈运算。在栈顶位置插入一个新元素。首先将栈顶指针加 1，然后将新元素插入到栈顶指针指向的位置。

②退栈运算。取出栈顶元素并赋给一个指定变量。首先将栈顶元素赋给一个指定的变量，然后将栈顶指针减 1。当栈顶指针为 0 时，说明栈空，不可能进行退栈操作。此时，发生栈“下溢”错误。

③读栈顶元素。将栈顶元素赋给一个指定的变量，栈指针不变。当栈顶指针为 0 时，读栈顶元素操作失败。

5. 队列（基础，识记）

(1) 队列的基本概念

队列是只允许在一端进行删除，在另一端进行插入的顺序表，通常将允许删除的这一端称为队头，允许插入的这一端称为队尾。当表中没有元素时称为空队列。

队列的修改是依照先进先出的原则进行的，因此队列也称为先进先出的线性表，或者后进后出的线性表。若有队列：

$$Q = (q_1, q_2, \dots, q_n)$$

那么， q_1 为队头元素（排头元素）， q_n 为队尾元素。队列中的元素是按照 q_1, q_2, \dots, q_n 顺序进入的，退出队列也只能按照这个次序依次退出，即只有在 q_1, q_2, \dots, q_{n-1} 都退队之后， q_n 才能退出队列。因最先进入队列的元素将最先出队，所以队列具有先进先出特性，体现“先来先服务”的原则。

队头元素 q_1 是最先被插入的元素，也是最先被删除的元素。队尾元素 q_n 是最后被插入的元素，也是最后被删除的元素。因此，与栈相反，队列又称为“先进先出”（First In First Out, FIFO）或“后进后出”（Last In Last Out, LILO）的线性表。

(2) 队列运算

①入队运算是往队列队尾插入一个数据元素。首先将队尾指针加 1 ($rear=rear+1$)，并当 $rear=m+1$ (m 表示队列的长度) 时置 $rear=1$ ；然后将新元素插入到指针指向的位置。

②退队运算是从队列的队头删除一个数据元素。首先将排头指针加 1 ($front=front+1$)，并当 $front=m+1$ (m 表示队列的长度) 时置 $front=1$ ；然后将排头指针指向的元素赋给指定变量。

(3) 循环队列

在实际应用中，队列的顺序存储结构一般采用循环队列的形式。所谓循环队列，就是将队列存储空间最后一个位置绕到第一个位置，形成逻辑上的环状空间。循环队列 $s=0$ 表示队列空； $s=1$ 且 $front=rear$ 表示队列满。计算循环队列的元素个数：“尾指针减头指针”，若为负数，再加其容量即可。

6. 链表（基础，识记）

在链式存储方式中，要求每个结点由两部分组成：一部分用于存放数据元素值，称为数据域；另一部分用于存放指针，称为指针域。其中指针用于指向该结点的前一个或后一个结点（即前件或后件）。

链式存储方式既可用于表示线性结构，也可用于表示非线性结构。

(1) 线性链表

①线性表的链式存储结构称为线性链表。

在线性链表中，各数据元素结点的存储空间可以是不连续的，且各数据元素的存储顺序与逻辑顺序可以不一致。在线性链表中进行插入与删除，不需要移动链表中的元素。

②在某些应用中，对线性链表中的每个结点设置两个指针，一个称为左指针，用以指向其前件结点；另一个称为右指针，用以指向其后件结点。这样的表称为双向链表。

线性单链表中，HEAD 称为头指针，HEAD=NULL（或 0）称为空表。如果是双向链表的两指针：左指针（Llink）指向前件结点，右指针（Rlink）指向后件结点。

(2) 带链的栈

栈也是线性表，也可以采用链式存储结构。带链的栈可以用来收集计算机存储空间中所有空闲的存储结点，

这种带链的栈称为可利用栈。

(3) 带链的队列

队列的链式结构和线性链表的存储结构基本相同。只不过队列的链式结构保持有两个指针，一个是指向队列头的头指针，另一个是指向队列尾的尾指针。

(4) 线性链表的基本运算

线性链表的基本运算：查找、插入、删除。

①线性链表的查找。从头指针指向的结点开始往后沿指针进行扫描，直到后面已没有结点或下一个结点的数据域为查找值为止。

②线性链表的插入。先从栈中为新元素分配一个新的结点，并赋值。然后利用线性链表的查找算法找到待插入位置的前一个结点指针，将新结点指向该结点的后件，然后将新结点挂在该结点的后面。

③线性链表的删除。利用线性链表的查找算法找到要删除元素的前一个结点，将该结点的指针指向删除元素的结点的指针指向的结点，将包含删除元素的结点送回可利用栈。

7. 二叉树（基础，识记）

(1) 二叉树概念及其基本性质

①二叉树及其基本概念。

二叉树是一种很有用的非线性结构，具有以下两个特点：

- 非空二叉树只有一个根结点；
- 每一个结点最多有两棵子树，且分别称为该结点的左子树和右子树。

在二叉树中，每一个结点的度最大为 2，即所有子树（左子树或右子树）也均为二叉树。另外，二叉树中每个结点的子树被明显地分为左子树和右子树。

在二叉树中，一个结点可以只有左子树而没有右子树，也可以只有右子树而没有左子树。当一个结点既没有左子树也没有右子树时，该结点即为叶子结点。

例如，一个家族中的族谱关系如图 1-1 所示：

A 有后代 B, C;

B 有后代 D, E;

C 有后代 F;

典型的二叉树如图 1-1 所示：

下面就表 1-2 详细讲解二叉树的一些基本概念。

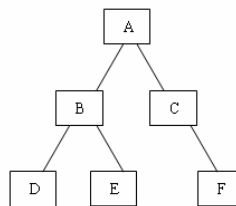


图 1-1 族谱二叉树

表 1-2 二叉树概念

父结点（根）	在树结构中，每一个结点只有一个前件，称为父结点，没有前件的结点只有一个，称为树的根结点，简称树的根。例如，在图 1-1 中，结点 A 是树的根结点
子结点和叶子结点	在树结构中，每一个结点可以有多个后件，称为该结点的子结点。没有后件的结点称为叶子结点。例如，在图 1-1 中，结点 D, E, F 均为叶子结点
度	在树结构中，一个结点所拥有的后件个数称为该结点的度，所有结点中最大的度称为树的度。例如，在图 1-1 中，根结点 A 和结点 B 的度为 2，结点 C 的度为 1，叶子结点 D, E, F 的度为 0。所以，该树的度为 2
深度	定义一棵树的根结点所在的层次为 1，其他结点所在的层次等于它的父结点所在的层次加 1。树的最大层次称为树的深度。例如，在图 1-1 中，根结点 A 在第 1 层，结点 B, C 在第 2 层，结点 D, E, F 在第 3 层。该树的深度为 3
子树	在树中，以某结点的一个子结点为根构成的树称为该结点的一棵子树

②二叉树基本性质。

二叉树具有以下几个性质：

性质 1：在二叉树的第 k 层上，最多有 2^{k-1} ($k \geq 1$) 个结点；

性质 2：深度为 m 的二叉树最多有 $2^m - 1$ 个结点；

性质 3：在任意一棵二叉树中，度为 0 的结点（即叶子结点）总是比度为 2 的结点多一个。

性质 4：具有 n 个结点的二叉树，其深度至少为 $\lceil \log_2 n \rceil + 1$ ，其中 $\lceil \log_2 n \rceil$ 表示取 $\log_2 n$ 的整数部分。

(2) 满二叉树与完全二叉树

满二叉树是指除最后一层外，每一层上的所有结点都有两个子结点。在满二叉树中，每一层上的结点数都

达到最大值，即在满二叉树的第 k 层上有 2^{k-1} 个结点，且深度为 m 的满二叉树有 2^m-1 个结点。

完全二叉树是指除最后一层外，每一层上的结点数均达到最大值；在最后一层上只缺少右边的若干结点。

对于完全二叉树来说，叶子结点只可能在层次最大的两层上出现：对于任何一个结点，若其右分支下子孙结点的最大层次为 p ，则其左分支下子孙结点的最大层次为 p ，或为 $p+1$ 。

完全二叉树具有以下两个性质：

性质 5：具有 n 个结点的完全二叉树的深度为 $\lceil \log_2 n \rceil + 1$ 。

性质 6：设完全二叉树共有 n 个结点。如果从根结点开始，按层次（每一层从左到右）用自然数 $1, 2, \dots, n$ 给结点进行编号，则对于编号为 k ($k=1, 2, \dots, n$) 的结点有以下结论：

①若 $k=1$ ，则该结点为根结点，它没有父结点；若 $k>1$ ，则该结点的父结点编号为 $\text{INT}(k/2)$ 。

②若 $2k \leq n$ ，则编号为 k 的结点的左子结点编号为 $2k$ ；否则该结点无左子结点（显然也没有右子结点）。

③若 $2k+1 \leq n$ ，则编号为 k 的结点的右子结点编号为 $2k+1$ ；否则该结点无右子结点。

(3) 二叉树的遍历

在遍历二叉树的过程中，一般先遍历左子树，再遍历右子树。在先左后右的原则下，根据访问根结点的次序，二叉树的遍历分为 3 类：前序遍历、中序遍历和后序遍历。

①前序遍历。

先访问根结点，然后遍历左子树，最后遍历右子树，并且在遍历左、右子树时，仍然先访问根结点，然后遍历左子树，最后遍历右子树。例如，对图 1-1 中二叉树进行前序遍历的结果（或称为该二叉树的前序序列）为：A, B, D, E, C, F。

②中序遍历。

先遍历左子树，然后访问根结点，最后遍历右子树，并且在遍历左、右子树时，仍然先遍历左子树，然后访问根结点，最后遍历右子树。例如，对图 1-1 中二叉树进行中序遍历的结果（或称为该二叉树的中序序列）为：D, B, E, A, C, F。

③后序遍历。

先遍历左子树，然后遍历右子树，最后访问根结点，并且在遍历左、右子树时，仍然先遍历左子树，然后遍历右子树，最后访问根结点。例如，对图 1-1 中二叉树进行后序遍历的结果（或称为该二叉树的后序序列）为：D, E, B, F, C, A。

8. 查找（基础，识记）

(1) 顺序查找

顺序查找是指在一个给定的数据结构中查找某个指定的元素。从线性表的第一个元素开始，依次将线性表中的元素与被查找的元素相比较，若相等则表示查找成功；若线性表中所有的元素都与被查找元素进行了比较但都不相等，则表示查找失败。

在下列两种情况下只能采用顺序查找：

①如果线性表为无序表，则不管是顺序存储结构还是链式存储结构，只能用顺序查找。

②即使是有序线性表，如果采用链式存储结构，也只能用顺序查找。

(2) 二分法查找

二分法查找，也称折半查找，是一种高效的查找方法。能使用二分法查找的线性表必须满足用顺序存储结构和线性表是有序表两个条件。

在本书中，为了简化问题，更方便讨论，“有序”是特指元素按非递减排列，即从小到大排列，但允许相邻元素相等。后面讲解的排序中，有序的含义也是如此。

对于长度为 n 的有序线性表，利用二分法查找元素 x 的过程如下。

步骤 1：将 x 与线性表的中间项比较；

步骤 2：如果 x 的值与中间项的值相等，则查找成功，结束查找；

步骤 3：如果 x 小于中间项的值，则在线性表的前半部分以二分法继续查找；

步骤 4：如果 x 大于中间项的值，则在线性表的后半部分以二分法继续查找。

顺序查找法每一次比较，只将查找范围减少 1，而二分法查找，每比较一次，可将查找范围减少为原来的一半，效率大大提高。

对于长度为 n 的有序线性表，在最坏情况下，二分法查找只需比较 $\log_2 n$ 次，而顺序查找需要比较 n 次。