

飞思考试中心

全国计算机等级考试考点分析、题解与模拟： 二级公共基础知识(最新版)

全国计算机等级考试命题研究中心

编著

飞思教育产品研发中心

联合监制

未来教育教学与研究中心

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书依据教育部考试中心最新发布的《全国计算机等级考试考试大纲》,在《全国计算机等级考试考点分析、题解与模拟(2012版)》的基础上修订而成。在编写过程中,一方面结合最新大纲和数套真卷,对重要考点进行了分析、讲解,并选取经典考题进行了深入剖析;另一方面配有同步练习和模拟试题,以逐步向考生详尽透析考试中的所有知识要点。“一书在手,通关无忧”。

本书适合作为全国计算机等级考试考前培训班辅导用书,也可作为应试人员的自学用书。

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考点分析、题解与模拟. 二级公共基础知识/全国计算机等级考试命题研究中心编著. —北京:电子工业出版社, 2007. 11

(飞思考试中心)

ISBN 978-7-121-05204-0

I. 全... II. 全... 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(2007)第 160316 号

责任编辑:王树伟

特约编辑:赵树刚

印 刷: 北京富博印刷有限公司
装 订:

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:880×1230 1/16 印张:10.75 字数:447.2 千字

印 次:2013 年 1 月第 1 次印刷

定 价:19.80 元(含光盘 1 张)

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 zlt@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

全国计算机等级考试自 1994 年由国家教育部考试中心推出以来,为评测全社会非计算机专业人员的计算机知识与技能,培养各行业的计算机应用人才开辟了一条新的道路,受到了用人单位和学习人员的热烈欢迎。全国计算机等级考试通过数年的发展,已经成为我国最大型的计算机类考试。

为了帮助更多的学习者顺利地通过考试,并掌握相应的操作技能,我们在深入调研、详尽分析考试大纲的基础上,组织国内著名高校的计算机专家和一线教师编写了本书。

本书共分为两大部分。

※ 考点分析/真题题解/同步练习

“考点分析”结合最新考试大纲、教材,对教材中考核的重点和难点进行了讲解,内容涵盖了大纲中所有的考点。

“真题题解”选取极具代表性的经典例题。例题符合考试命题规律的特征,对题目的讲解深入、透彻,循序渐进,极有条理。

“同步练习”提供了大量习题,对前面所学的理论知识进行温习和巩固,以练促学、学练结合。

※ 笔试全真模拟试卷

结合最新考试大纲,筛选与演绎出的典型试卷集,不论在形式上还是难度上,都与真题类似,解析详尽、透彻。

由于时间仓促,书中难免有不当之处,敬请指正。

联系方式

电 话: (010)82552266 68134545 88254160

电子邮件: support@fecit.com.cn eduwin@sina.com

未来教育考试网: <http://www.eduexam.cn>

飞思在线: <http://www.fecit.com.cn> <http://www.fecit.net>

通用网址: 计算机图书、飞思、飞思教育、飞思科技、FECIT

全国计算机等级考试命题研究中心
飞思教育产品研发中心
未来教育教学与研究中心

第 1 章 数据结构与算法

1.1 算法	1	1.6 树与二叉树	20
1.2 数据结构的基本概念	4	1.7 查找技术	29
1.3 线性表及顺序存储结构	7	1.8 排序技术	32
1.4 栈和队列	10	1.9 同步练习	41
1.5 线性链表	14	1.10 参考答案	48

第 2 章 程序设计基础

2.1 程序设计方法与风格	50	2.4 同步练习	60
2.2 结构化程序设计	52	2.5 参考答案	62
2.3 面向对象的程序设计	54		

第 3 章 软件工程基础

3.1 软件工程的基本概念	63	3.5 程序的调试	92
3.2 结构化分析方法	70	3.6 软件工程管理	94
3.3 结构化设计方法	77	3.7 同步练习	95
3.4 软件的测试	88	3.8 参考答案	108

第 4 章 数据库设计基础

4.1 数据库系统的基本概念	109	4.4 数据库的设计与管理	135
4.2 数据模型	121	4.5 同步练习	145
4.3 关系代数	132	4.6 参考答案	157

第 5 章 无纸化考试试题

5.1 无纸化考试试题(1)	158	5.4 无纸化考试试题(4)	160
5.2 无纸化考试试题(2)	159	5.5 无纸化考试试题(5)	161
5.3 无纸化考试试题(3)	160	5.6 参考答案及解析	162

第 1 章 数据结构与算法

考核知识点

- 算法的基本概念
- 数据结构的基本概念
- 线性表的定义
- 栈和队列
- 线性链表
- 循环链表
- 树的基本概念
- 查找技术
- 排序技术

重要考点提示

- 算法的基本概念;算法复杂度的概念和意义(时间复杂度与空间复杂度)
- 数据结构的定义;数据的逻辑结构与存储结构;数据结构的图形表示;线性结构与非线性结构
- 线性表的定义;线性表的顺序存储结构及其插入与删除运算
- 栈和队列的定义;栈和队列的顺序存储结构及其基本运算
- 线性单链表、双向链表与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(交换类排序、选择类排序、插入类排序)

1.1 算 法

考点 1 算法的基本概念

计算机解题的过程实际上是在实施某种算法,这种算法称为计算机算法。

算法(algorithm)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,同时是明确的;此顺序将在有限的次数后终止。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。

1. 算法的基本特征

- (1)可行性(effectiveness):针对实际问题而设计的算法,执行后能够得到满意的结果。
- (2)确定性(definiteness):算法中的每一个步骤都必须有明确的定义,不允许有模棱两可的解释和多义性。
- (3)有穷性(finiteness):算法必须在有限时间内做完,即算法必须能在执行有限个步骤之后终止。
- (4)拥有足够的情报:要使算法有效必须为算法提供足够的情报。当算法拥有足够的情报时,此算法才是有效的;而当提供的情报不够时,算法可能无效。

2. 算法的基本要素

(1)算法中对数据的运算和操作:每个算法实际上是按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。

计算机可以执行的基本操作是以指令的形式描述的。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,基本的运算和操作有以下4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”、“不等于”等运算。
- ④数据传输:主要包括赋值、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的执行顺序,而且也直接反映了算法的设计是否符合结构化原则。描述算法的工具通常有传统流程图、N-S结构化流程图、算法描述语言等。一个算法一般都可以用顺序、选择、循环3种基本控制结构组合而成。

3. 算法设计的基本方法

计算机算法不同于人工处理的方法,下面是工程上常用的几种算法设计,在实际应用时,各种方法之间往往存在一定的联系。

(1)列举法

列举法是计算机算法中的一个基础算法。列举法的基本思想是,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。

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,在用列举法设计算法时,使方案优化,尽量减少运算工作量,是应该重点注意的。

(2)归纳法

归纳法的基本思想是,通过列举少量的特殊情况,经过分析,最后找出一般的关系。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。

(3)递推

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。递推本质上属于归纳法,工程上许多递推关系式实际上是通过通过对实际问题的分析与归纳而得到的,因此,递推关系式往往是归纳的结果。对于数值型的递推算法必须要注意数值计算的稳定性问题。

(4)递归

人们在解决一些复杂问题时,为了降低问题的复杂程度(如问题的规模等),一般总是将问题逐层分解,最后归结为一些最简单的问题。这种将问题逐层分解的过程,实际上并没有对问题进行求解,而只是当解决了最后那些最简单的问题后,再沿着原来分解的逆过程逐步进行综合,这就是递归的基本思想。

递归分为直接递归与间接递归两种。

(5)减半递推技术

实际问题的复杂程度往往与问题的规模有着密切的联系。因此,利用分治法解决这类实际问题是有效的。工程上常用的分治法是减半递推技术。

所谓“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”的过程。

(6)回溯法

在工程上,有些实际问题很难归纳出一组简单的递推公式或直观的求解步骤,并且也不能进行无限的列举。对于这类问题,一种有效的方法是“试”。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,若试探成功,就得到问题的解;若试探失败,就逐步回退,换别的路线再逐步试探。

4. 算法设计的要求

通常一个好的算法应达到如下目标:

(1) 正确性(correctness)

正确性大体可以分为以下4个层次:

- ①程序不含语法错误;
- ②程序对于几组输入数据能够得出满足规格说明要求的结果;
- ③程序对于精心选择的典型、苛刻而带有刁难性的几组输入数据能够得出满足规格说明要求的结果;
- ④程序对于一切合法的输入数据都能产生满足规格说明要求的结果。

(2) 可读性(readability)

算法主要是为了方便人的阅读与交流,其次才是其执行。可读性好有助于用户对算法的理解;晦涩难懂的程序易于隐藏较多错误,难以调试和修改。

(3) 健壮性(robustness)

当输入数据非法时,算法也能适当地做出反应或进行处理,而不会产生莫名其妙地输出结果。

(4) 效率与低存储量需求

效率指的是程序执行时,对于同一个问题如果有多个算法可以解决,执行时间短的算法效率高;存储量需求指算法执行过程中所需要的最大存储空间。

考点 2 算法的复杂度

1. 算法的时间复杂度

算法的时间复杂度,是指执行算法所需要的计算工作量。同一个算法用不同的语言实现,或者用不同的编译程序进行编译,或者在不同的计算机上运行,效率均不同。这表明使用绝对的时间单位衡量算法的效率是不合适的。撇开这些与计算机硬件、软件有关的因素,可以认为一个特定算法“运行工作量”的大小,只依赖于问题的规模(通常用整数 n 表示),它是问题的规模函数。即

$$\text{算法的工作量} = f(n)$$

例如,在 $N \times N$ 矩阵相乘的算法中,整个算法的执行时间与该基本操作(乘法)重复执行的次数 n^3 成正比,也就是时间复杂度为 n^3 ,即

$$f(n) = O(n^3)$$

在有的情况下,算法中的基本操作重复执行的次数还随问题的输入数据集不同而不同。例如,在起泡排序的算法中,当要排序的数组 a 初始序列为自小至大有序时,基本操作的执行次数为0;当初始序列为自大至小有序时,基本操作的执行次数为 $n(n-1)/2$ 。对这类算法的分析,可以采用以下两种方法来分析。

(1) 平均性态(Average Behavior)

所谓平均性态是指利用各种特定输入下的基本运算次数的加权平均值来度量算法的工作量。

设 x 是有可能输入中的某个特定输入, $p(x)$ 是 x 出现的概率(即输入为 x 的概率), $t(x)$ 是算法在输入为 x 时所执行的基本运算次数,则算法的平均性态定义为

$$A(n) = \sum_{x \in D_n} P(x) t(x)$$

其中 D_n 表示当规模为 n 时,算法执行的所有可能输入的集合。

(2) 最坏情况复杂性(Worst-case Complexity)

所谓最坏情况分析,是指在规模为 n 时,算法所执行的基本运算的最大次数。

$$W(n) = \max_{x \in D_n} \{t(x)\}$$

2. 算法的空间复杂度

算法的空间复杂度是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的空间、输入的初始数据所占的存储空间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及某种数据结构所需要的附加存储空间。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

际问题中,为了减少算法所占的存储空间,通常采用压缩存储技术,以便尽量减少不必要的额外空间。

真题题解

【例1】算法的时间复杂度取决于()。

- A) 问题的规模
- C) 问题的难度

- B) 待处理的数据的初态
- D) A) 和 B)

【命题目的】影响算法时间复杂度的主要因素。

【解题要点】算法的时间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及输入各种数据或数据集的概率有关。

【错解分析】算法的时间复杂度与问题的难度无关,而是与问题的规模和数据的初态有关,所以 C) 选项错误,A) 选项和 B) 选项正确。

【考点链接】算法的空间复杂度及算法时间复杂度和空间复杂度的计算方法。

【答案】D)

1.2 数据结构的基本概念

考点 3 数据结构的定义

数据结构(data structure)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,即数据的组织形式。

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

讨论以上问题的目的是提高数据处理的效率,所谓提高数据处理的效率有两个方面:

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

数据(data):是对客观事物的符号表示,在计算机科学中是指所有能输入到计算机中并被计算机程序处理的符号的总称。

数据元素(data element):是数据的基本单位,在计算机程序中通常作为一个整体进行考虑和处理。

数据对象(data object):是性质相同的数据元素的集合,是数据的一个子集。

一般情况下,在具有相同特征的数据元素集合中,各个数据元素之间存在某种关系(即连续),这种关系反映了该集合中的数据元素所固有的一种结构。在数据处理领域中,通常把数据元素之间这种固有的关系简单地用前后件关系(或直接前驱与直接后继关系)来描述。

前后件关系是数据元素之间的一个基本关系,但前后件关系所表示的实际意义随具体对象的不同而不同。一般来说,数据元素之间的任何关系都可以用前后件关系来描述。

1. 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。更通俗地说,数据结构是指带有结构的数据元素的集合。所谓结构实际上就是指数据元素之间的前后件关系。

一个数据结构应包含以下两方面信息:

- (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。

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。

数据的逻辑结构包括集合、线性结构、树型结构和图形结构四种。

线性结构:数据元素之间构成一种顺序的线性关系,如图 1-1 所示。

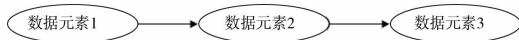


图 1-1 线性结构

树型结构:数据元素之间形成一种树型的关系,如图 1-2 所示。

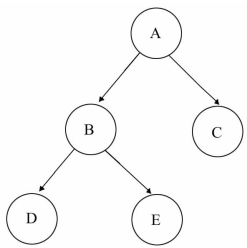


图 1-2 树型结构

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。

例如,复数是一种数据结构,在计算机科学中,复数可取如下定义:

$$B = (C, R)$$

其中, C 是含有两个实数的集合 $\{c_1, c_2\}$; R 是定义在集合 C 上的一种关系 $\{ \langle c_1, c_2 \rangle \}$, 其中有序偶 $\{ \langle c_1, c_2 \rangle \}$ 表示 c_1 是复数的实部, c_2 是复数的虚部。

2. 数据的存储结构

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,常用的存储结构有顺序、链接、索引等存储结构。而采用不同的存储结构,其数据处理的效率是不同的。因此,在进行数据处理时,选择合适的存储结构是很重要的。

考点 4 数据结构的图形表示

数据结构除了用二元关系表示外,还可以直观地用图形表示。

在数据结构的图形表示中,对于数据集合 D 中的每一个数据元素用中间标有元素值的方框表示,一般称为数据结点,并简称为结点;为了进一步表示各数据元素之间的前后件关系,对于关系 R 中的每一个二元组,用一条有向线段从前件结点指向后件结点。

例如,反映家庭成员间辈分关系的数据结构可以用如图 1-3 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(也称为叶子结点)。

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,可以在一个数据结构中增加一个新结点(称为插入运算),也可以删除数据结构中的某个结点(称为删除运算)。插入与删除是对数据结构的两种基本运算。除此之外,对数据结构的运算还有查找、分类、合并、分解、复制和修改等。

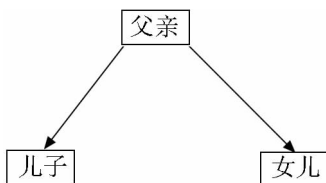


图 1-3 家庭成员辈分关系数据结构的图形表示

考点 5 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。

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。

非空数据结构满足:

- (1)有且只有一个根结点;
- (2)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。线性结构又称为线性表。一个线性表是 n 个数据元素的有限序列。至于每个元素的具体含义,在不同的情况下各不相同,它可以是一个数或一个符号,也可以是一页书,甚至其他更复杂的信息。如果一个数据结构不是线性结构,称为非线性结构。线性结构与非线性结构都可以是空的数据结构。对于空的数据结构,如果对该数据结构的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

【命题目的】考查数据存储结构的定义。

【解题要点】数据的存储结构是数据结构在计算机内存中的表示,它既保存数据元素也保存数据元素之间的关系。

【错解分析】数据结构是指相互之间存在一种或者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) 选项错误。数据的逻辑结构描述的是数据之间的逻辑关系,因此 C) 选项错误。数据元素之间的关系就是指数据的逻辑结构,因此 D) 选项错误。

【考点链接】数据结构和数据逻辑结构的定义。

【答案】A)

【例 2】在数据结构中,与所使用的计算机无关的数据结构是()。

- A) 逻辑
- B) 存储
- C) 逻辑和存储
- D) 物理

【命题目的】考查数据的逻辑结构和存储结构的特点。

【解题要点】存储结构、物理结构是同一概念的两个术语,都是数据结构在计算机内存中的表示,逻辑结构是数据元素间关系的描述,与所用的计算机无关。

【错解分析】数据的存储结构又称为数据的物理结构,是指数据在计算机内存中的表示,与所使用的计算机密切相关,因此 B) 选项、C) 选项和 D) 选项都是错误的。

【考点链接】数据的存储结构和逻辑结构的定义及特点。

【答案】A)

【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

【命题目的】考查数据结构从逻辑上的分类。

【解题要点】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)。

【错解分析】数据的逻辑结构就是指数据元素之间的相互关系,A) 选项、B) 选项和 D) 选项均不能表示数据元素之间的关系,只有 C) 选项可以。线性结构和非线性结构是根据数据元素之间关系的不同特性加以分类的,即从逻辑上加以分类。

【考点链接】数据存储结构的分类。

【答案】C)

1.3 线性表及顺序存储结构

考点 6 线性表的定义

线性表是 $n(n \geq 0)$ 个元素构成的有限序列 (a_1, a_2, \dots, a_n) 。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。即线性表是一个空表,或可以表示为

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

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

其中,每个元素可以简单到是一个字母或是一个数据,也可能是比较复杂的由多个数据项组成的。在复杂的线性表中,由若干数据项组成的数据元素称为记录(record),而由多个记录构成的线性表又称为文件(file)。在非空表中的每个数据元素都有一个确定的位置,如 a_1 是第一个元素, a_n 是最后一个数据元素, a_i 是第 i 个数据元素,称 i 为数据元素 a_i 在线性表中的位序。非空线性表有如下一些结构特征:

(1) 有且只有一个根结点 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。

考点 7 线性表的顺序存储结构

线性表的顺序表指的是用一组地址连续的存储单元依次存储线性表的数据元素。

线性表的顺序存储结构具备如下两个基本特征:

(1) 线性表中的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。

假设线性表的每个元素需要占用 k 个存储单元,并以所占的存储位置 $ADR(a_{i+1})$ 和第 i 个数据元素的存储位置 $ADR(a_i)$ 之间满足下列关系:

$$ADR(a_{i+1}) = ADR(a_i) + k$$

线性表第 i 个元素 a_i 的存储位置为

$$ADR(a_i) = ADR(a_1) + (i - 1) \times k$$

式中 $ADR(a_1)$ 是线性表的第一个数据元素 a_1 的存储位置,通常称做线性表的起始位置或基址。

线性表的这种表示称做线性表的顺序存储结构或顺序映像,这种存储结构的线性表为顺序表。表中每一个元素的存储位置都和线性表的起始位置相差一个和数据元素在线性表中的位序成正比例的常数。如图 1-4 所示。由此只要确定了存储线性表的起始位置,线性表中任一数据元素都可以随机存取,所以线性表

的顺序存储结构是一种随机存取的存储结构。

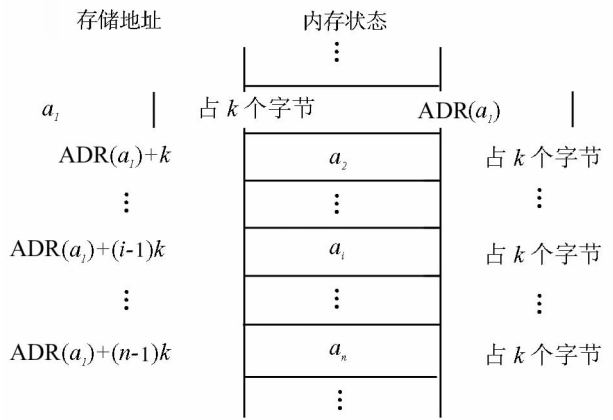


图 1-4 线性表顺序存储结构示意图

在程序设计语言中,通常定义一个一维数组来表示线性表的顺序存储空间。在用一维数组存放线性表时,该一维数组的长度通常要定义得比线性表的实际长度大一些,以便对线性表进行各种运算,特别是插入运算。在线性表的顺序存储结构下,可以对线性表做以下运算:

- (1)在线性表的指定位置处加入一个新的元素(即线性表的插入);
- (2)在线性表中删除指定的元素(即线性表的删除);
- (3)在线性表中查找某个(或某些)特定的元素(即线性表的查找);
- (4)对线性表中的元素进行整序(即线性表的排序);
- (5)按要求将一个线性表分解成多个线性表(即线性表的分解);
- (6)按要求将多个线性表合并成一个线性表(即线性表的合并);
- (7)复制一个线性表(即线性表的复制);
- (8)逆转一个线性表(即线性表的逆转)等。

考点 8 顺序表的插入运算

线性表的插入运算是指在表的第 $i(1 \leq i \leq n+1)$ 个位置上,插入一个新结点 x ,使长度为 n 的线性表

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

变成长度为 $n+1$ 的线性表

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

现在分析算法的复杂度。这里的问题规模是表的长度,设它的值为 n 。该算法的时间主要花费在循环的结点后移语句上,该语句的执行次数(即移动结点的次数)是 $n-i+1$ 。由此可看出,所需移动结点的次数不仅依赖于表的长度,而且还与插入位置有关。

当 $i=n+1$ 时,由于循环变量的终值大于初值,结点后移语句将不进行;这是最好情况,其时间复杂度为 $O(1)$;

当 $i=1$ 时,结点后移语句,将循环执行 n 次,需移动表中所有结点,这是最坏情况,其时间复杂度为 $O(n)$ 。

由于插入可能在表中任何位置上进行,因此需分析算法的平均复杂度。

在长度为 n 的线性表中第 i 个位置上插入一个结点,令 $Eis(n)$ 表示移动结点的期望值(即移动的平均次数),则在第 i 个位置上插入一个结点的移动次数为 $n-i+1$ 。故

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

不失一般性,假设在表中任何位置($1 \leq i \leq n+1$)上插入结点的机会是均等的,则

$$p_1 = p_2 = p_3 = \dots = p_{n+1} = 1/(n+1)$$

因此,在等概率插入的情况下,

$$Eis(n) = \sum_{i=1}^{n+1} p_i (n-i+1) / (n+1) = n/2$$

也就是说,在顺序表上做插入运算,平均要移动表上一半的结点。当表长 n 较大时,算法的效率相当低。虽然 $Eis(n)$ 中 n 的系数较小,但就数量级而言,它仍然是线性级的。因此算法的平均时间复杂度为 $O(n)$,如图 1-5 所示。

插入位置	移动元素个数	内存状态	
1	n	<table><tr><td>a_1</td></tr></table>	a_1
a_1			
2	$n-1$	<table><tr><td>a_2</td></tr></table>	a_2
a_2			
\vdots	\vdots	<table><tr><td>\vdots</td></tr></table>	\vdots
\vdots			
i	$n-i+1$	<table><tr><td>a_i</td></tr></table>	a_i
a_i			
\vdots	\vdots	<table><tr><td>\vdots</td></tr></table>	\vdots
\vdots			
n	1	<table><tr><td>a_n</td></tr></table>	a_n
a_n			
$n+1$	0	<table><tr><td></td></tr></table>	

图 1-5 插入操作时间复杂度示意图

考点 9 顺序表的删除运算

线性表的删除运算是指将表的第 $i(1 \leq i \leq n)$ 个结点删除,使长度为 n 的线性表:

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

变成长度为 $n-1$ 的线性表:

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

该算法的时间分析与插入算法相似,结点的移动次数也是由表长 n 和位置 i 决定。若 $i=n$,则由于循环变量的初值大于终值,前移语句将不执行,无须移动结点;若 $i=1$,则前移语句将循环执行 $n-1$ 次,需移动表中除开始结点外的所有结点。这两种情况下算法的时间复杂度分别为 $O(1)$ 和 $O(n)$ 。

删除算法的平均性能分析与插入算法相似。在长度为 n 的线性表中删除一个结点,令 $Ede(n)$ 表示所需移动结点的平均次数,删除表中第 i 个结点的移动次数为 $n-i$,故

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

式子中, p_i 表示删除表中第 i 个结点的概率。在等概率的假设下,

$$p_1 = p_2 = p_3 = \cdots = p_n = 1/n$$

由此可得:

$$Ede(n) = \sum_{i=1}^n p_i (n-i) / n = (n-1) / 2$$

即在顺序表上做删除运算,平均要移动表中约一半的结点,平均时间复杂度也是 $O(n)$ 。

删除元素位置	移动元素个数	内存状态	
1	$n-1$	<table><tr><td>a_1</td></tr></table>	a_1
a_1			
2	$n-2$	<table><tr><td>a_2</td></tr></table>	a_2
a_2			
\vdots	\vdots	<table><tr><td>\vdots</td></tr></table>	\vdots
\vdots			
i	$n-i$	<table><tr><td>a_i</td></tr></table>	a_i
a_i			
\vdots	\vdots	<table><tr><td>\vdots</td></tr></table>	\vdots
\vdots			
n	0	<table><tr><td>a_n</td></tr></table>	a_n
a_n			
		<table><tr><td></td></tr></table>	

图 1-6 删除操作时间复杂度示意图

真题题解

【例1】用数组表示线性表的优点是()。

- A) 便于插入和删除操作
- B) 便于随机存取
- C) 可以动态地分配存储空间
- D) 不需要占用一片相邻的存储空间

【命题目的】考查数组的存储方式。

【解题要点】由于数组的存储方式是采用顺序存储的,即占用连续的存储空间,所以可以用数组的下标直接存取。

【错解分析】对顺序存储的线性表进行插入和删除操作时,需要移动大量的元素,因此不便于进行插入删除操作,A)选项错误;数组的存储空间大小一开始就已经分配好,不能动态分配存储空间,所以C)选项错误;数组正是占用一片相邻的存储空间,所以D)选项也错误。

【考点链接】线性表的存储方式。

【答案】B)

1.4 栈和队列

考点 10 栈及其基本运算

1. 什么是栈

栈实际也是线性表,只不过是一种特殊的线性表。栈(Stack)是只能在表的一端进行插入和删除运算的线性表,通常称插入、删除的这一端为栈顶(Top),另一端为栈底(Bottom)。当表中没有元素时称为空栈。栈顶元素总是后被插入的元素,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。

假设栈 $S = (a_1, a_2, a_3, \dots, a_n)$, 则 a_1 称为栈底元素, a_n 为栈顶元素。栈中元素按 $a_1, a_2, a_3, \dots, a_n$ 的次序进栈,退栈的第一个元素应为栈顶元素。换句话说,栈的修改是按后进先出的原则进行的。因此,栈称为先进后出表(FILO, First In Last Out),或“后进先出”表(LIFO, Last In First Out),如图1-7所示。

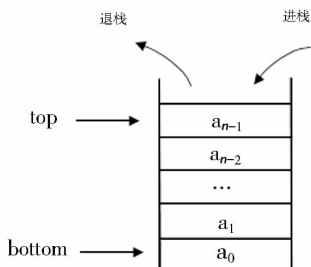


图 1-7 进栈出栈示意图

2. 栈的顺序存储及其运算

(1)入栈运算:入栈运算是指在栈顶位置插入一个新元素。首先将栈顶指针加1(即top加1),然后将新元素插入到栈顶指针指向的位置。当栈顶指针已经指向存储空间的最后一个位置时,说明栈空间已满,不可能再进行入栈操作。这种情况称为栈“上溢”错误,如图1-8所示。

(2)退栈运算:退栈是指取出栈顶元素并赋给一个指定的变量。首先将栈顶元素(栈顶指针指向的元素)赋给一个指定的变量,然后将栈顶指针减1(即top减1)。当栈顶指针为0时,说明栈空,不可进行退栈操作。这种情况称为栈的“下溢”错误,如图1-8所示。

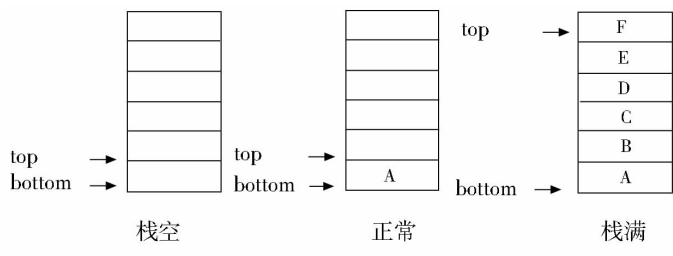


图 1-8 栈的状态

(3)读栈顶元素:读栈顶元素是指将栈顶元素赋给一个指定的变量。这个运算不删除栈顶元素,只是将它赋给一个变量,因此栈顶指针不会改变。当栈顶指针为 0 时,说明栈空,读不到栈顶元素。

考点 11 队列及其基本运算

1. 什么是队列

队列(queue)是只允许在一端删除,在另一端插入的顺序表,允许删除的一端叫做队头(front),允许插入的一端叫做队尾(rear),如图 1-9 所示。

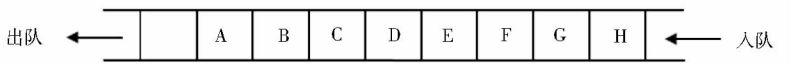


图 1-9 队列示意图

当队列中没有元素时称为空队列。在空队列中依次加入元素 a_1, a_2, \dots, a_n 之后, a_1 是队头元素, a_n 是队尾元素。显然退出队列的次序也只能是 a_1, a_2, \dots, a_n , 也就是说队列的修改是依先进先出的原则进行的。因此队列亦称做先进先出(FIFO, First In First Out)的线性表,或后进后出(LILO, Last In Last Out)的线性表。往队列队尾插入一个元素称为入队运算,从队列的排头删除一个元素称为退队运算,如图 1-10 所示。

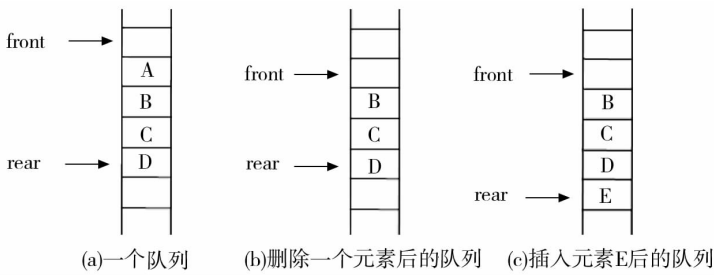


图 1-10 队列运算示意图

2. 循环队列及其运算

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,如图 1-11 所示。

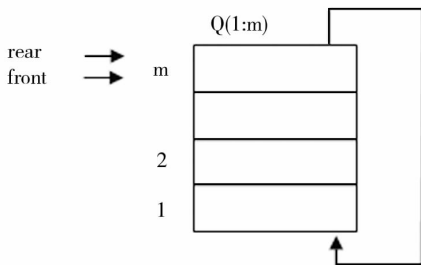


图 1-11 循环队列存储空间示意图

在循环队列中,用队尾指针 rear 指向队列中的队尾元素,用排头指针 front 指向排头元素的前一个位置。因此,从排头指针 front 指向的后一个位置直到队尾指针 rear 指向的位置之间所有的元素均为队列中的元素。

可以将向量空间想象为一个首尾相接的圆环,如图 1-12 所示,并称这种向量为循环向量,存储在其中的

队列称为循环队列(Circular Queue)。在循环队列中进行出队、入队操作时,头尾指针仍要加1,朝前移动。只不过当头尾指针指向向量上界($QueueSize - 1$)时,其加1操作的结果是指向向量的下界0。

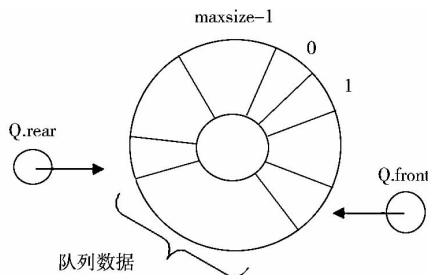


图 1-12 循环队列示意图

由于入队时尾指针向前追赶头指针,出队时头指针向前追赶尾指针,故队空和队满时头尾指针均相等。因此,我们无法通过 $front = rear$ 来判断队列“空”还是“满”。

在实际使用循环队列时,为了能区分队列满还是队列空,通常还需增加一个标志 s , s 值的定义如下:当 $s = 0$ 时表示队列空;当 $s = 1$ 时表示队列非空。

(1) 入队运算

入队运算是指在循环队列的队尾加入一个新元素。首先将队尾指针进1(即 $rear = rear + 1$),并当 $rear = m + 1$ 时置 $rear = 1$;然后将新元素插入到队尾指针指向的位置。当循环队列非空($s = 1$)且队尾指针等于队头指针时,说明循环队列已满,不能进行入队运算,这种情况称为“上溢”。

(2) 退队运算

退队运算是指在循环队列的队头位置退出一个元素并赋给指定的变量。首先将队头指针进1(即 $front = front + 1$),并当 $front = m + 1$ 时,置 $front = 1$;然后将排头指针指向的元素赋给指定的变量。当循环队列为空($s = 0$)时,不能进行退队运算,这种情况称为“下溢”。

真题题解

【例1】以下不是栈的基本运算的是()。

- A) 删除栈顶元素
- B) 删除栈底元素
- C) 判断栈是否为空
- D) 将栈置为空栈

【命题目的】考查有关栈结构的基本运算。

【解题要点】栈的基本运算有:入栈,出栈(删除栈顶元素),初始化、置空、判断栈是否为空或满、提取栈顶元素等,对栈的操作都是在栈顶进行的。

【错解分析】根据堆栈结构的后进先出的特点,每次删除堆栈元素时,总是先删除栈顶元素,而不是删除栈底元素。

【考点链接】栈结构的基本特点、基本运算和算法实现,队列结构的基本特点、基本运算和算法实现。

【答案】B)

【例2】若进栈序列为1,2,3,4,进栈过程中可以出栈,则下列不可能的一个出栈序列是()。

- A) 1,4,3,2
- B) 2,3,4,1
- C) 3,1,4,2
- D) 3,4,2,1

【命题目的】考查栈结构的先进后出的特点。

【解题要点】对栈这种数据结构,关键是要把握其中结点的后进先出的特点,选项C)中先出栈的是3,说明此时栈内必然有1,2,由于1先于2入栈,所以它不可能在2之前出栈,所以C)的出栈序列是不可能的。

【错解分析】解答此题,关键要把握“进栈过程中可以出栈”这句话。对于A)选项,1进栈,然后出栈,2、3、4依次进栈,然后4、3、2依次出栈;对于B)选项,1进栈,2进栈,2出栈,3进栈,3出栈,4进栈,4出栈,最后1出栈。对于D)选项,1、2、3依次进栈,然后3出栈,4进栈,4出栈,然后2、1依次出栈。

【考点链接】栈结构的出栈运算和入栈运算的算法。

【答案】C)

【例3】栈和队列的共同点是()。

- A) 都是先进先出
B) 都是先进后出
C) 只允许在端点处插入和删除元素
D) 没有共同点

【命题目的】考查栈结构和队列结构的特点。

【解题要点】栈和队列都是操作受限的线性表,只允许在表的端点处进行操作。所以答案为 C)。

【错解分析】堆栈结构特点是:后进先出,队列结构的特点是:先进先出,所以 A) 选项和 B) 选项错误。

【考点链接】堆栈结构和队列结构的区别。

【答案】C)

【例 4】若已知一个栈的进栈序列是 $1, 2, 3, \dots, n$, 其输出序列是 $p_1, p_2, p_3, \dots, p_n$, 若 $p_1 = n$, 则 $p_i (1 < i < n)$ 为()。

- A) i
B) $n-i$
C) $n-i+1$
D) 不确定

【命题目的】考查栈结构先进先出的特点和出栈算法。

【解题要点】当 $p_1 = n$, 即 n 是最先出栈的, 根据栈的原理, n 必定是最后进栈的, 所以输入序列必定是 $1, 2, 3, \dots, n$, 则出栈的序列是 $n, \dots, 3, 2, 1$ 。所以答案为 C)。

【错解分析】没有掌握栈的后进先出的特点,会导致选择错误答案。

【考点链接】栈的入栈和出栈算法。

【答案】C)

【例 5】判断一个栈 ST(最多元素为 MaxSize)为空的条件是()。

- A) ST -> top! == -1
B) ST -> top == -1
C) ST -> top! == MaxSize - 1
D) ST -> top == MaxSize - 1

【命题目的】考查判断栈是否为空的条件。

【解题要点】由顺序栈的结构可知,本题答案为 B)。

【错解分析】对于 D) 选项, $ST \rightarrow top == MaxSize - 1$ 是判断顺序栈为满的条件。

【考点链接】判断栈是否为满的条件。

【答案】B)

【例 6】下列关于线性表、栈和队列的叙述,错误的是()。

- A) 线性表是给定的 n (n 必须大于零) 个元素组成的序列
B) 线性表允许在表的任何位置进行插入和删除操作
C) 栈只允许在一端进行插入和删除操作
D) 队列允许在一端进行插入在另一端进行删除

【命题目的】考查线性表的概念;线性表、栈和队列的插入删除操作。

【解题要点】线性表可以是空表,即元素个数为零的表,A)选项说法是错误的,所以答案为A)。

【错解分析】线性表的插入和删除允许在任何位置进行,所以 B) 选项的说法是正确的;栈的操作只允许在栈顶进行,因此,栈称为先进后出表(FILO, First In Last Out),或“后进先出”表(LIFO, Last In First Out),所以 C) 选项说法也是正确的;队列(queue)是只允许在一端删除,在另一端插入的顺序表,允许删除的一端叫做队头(front),允许插入的一端叫做队尾(rear),因此队列亦称做“先进先出”(FIFO, First In First Out)的线性表,或“后进后出”(LILO, Last In Last Out)的线性表。所以 D) 选项说法也是正确的。

【考点链接】线性表、栈和队列的存储结构及它们的插入删除操作。

【答案】A)

【例7】一个队列的入队序列是1,2,3,4,则队列的输出序列是()。

- A) 4,3,2,1 B) 1,2,3,4 C) 1,4,3,2 D) 3,2,4,1

【命题目的】考查队列结构的先进先出的特点。

【解题要点】由队列的先进先出的特点可以知道,入队序列是1,2,3,4,出队序列只能是1,2,3,4。所以答案为B)。

【错解分析】没有掌握队列的先进先出的特点,容易误选答案。

【考点链接】堆栈结构和队列结构的区别。

【答案】B)

【例 8】设初始输入序列为 1,2,3,4,5,利用一个栈产生输出序列,下列()序列是不可能通过栈产生的。

- A) 1,2,3,4,5 B) 5,3,4,1,2 C) 4,3,2,1,5 D) 3,4,5,2,1

【命题目的】考查栈的进栈和出栈操作。

【解题要点】由于栈的压入和退出只能在栈顶进行,所以要使出栈的第一个数是序列的最后一个数5,只能先把序列所有元素都压入栈,但这时出栈序列只能是5,4,3,2,1,所以B)选项的出栈序列是错误的,应选B)。

【错解分析】当初始序列压入一个元素时,就退出一个元素,这样就得到A)选项的出栈序列1,2,3,4,5;先压入1,2,3,4四个元素,再退出所有元素,最后压入5,并退栈,这时就得到C)选项的出栈序列4,3,2,1,5;压入1,2后对后面的元素3,4,5分别压入一个退出一个,这时便得到D)选项的出栈序列3,4,5,2,1。

【考点链接】进栈和出栈操作。

【答案】B)

【例9】设栈S的初始状态为空,6个元素入栈的顺序为e1,e2,e3,e4,e5和e6。若出栈的顺序是e2,e4,e3,e6,e5,e1,则栈S的容量至少应该是()。

A) 6

B) 4

C) 3

D) 2

【命题目的】考查栈的入栈出栈操作。

【解题要点】由题意知,入栈出栈过程是,先压入e1,e2,再退出e2,再压入e3,e4,再退出e4,e3,再压入e5,e6,最后退出所有元素,驻留在栈中的元素最多有3个。

【错解分析】由以上分析知A)、B)、D)显然是错误的。

【考点链接】栈的入栈与出栈。

【答案】C)

【例10】为了减小栈溢出的可能性,可以让两个栈共享一片连续存储空间,两个栈的栈底分别设在这片空间的两端,这样只有当()时才可能产生上溢。

A) 两个栈的栈顶在栈空间的某一位置相遇

B) 其中一个栈的栈顶到达栈空间的中心点

C) 两个栈的栈顶同时到达栈空间的中心点

D) 两个栈均不空,且一个栈的栈顶到达另一个栈的栈底

【命题目的】考查栈的共享结构的特点和判断栈满的条件。

【解题要点】两栈共享连续存储空间,两个栈的栈底分别设在这个存储空间的两端的存储结构中,为了使两栈的存储空间能够做到互补余缺,减少溢出的可能性,两个栈的栈满溢出都不能按位置判别,仅当两栈的栈顶相遇时,才可能栈满溢出。B)选项、C)选项和D)选项都是按栈顶位置来判别的,因此都是错误的。A)选项符合上述判别原则,即两个栈的栈顶在栈空间的某一位置相遇时才可能产生上溢。所以答案为A)。

【错解分析】没有掌握和理解共享栈的数据结构,不容易理解此题。

【考点链接】判断共享栈结构是否为满的条件。

【答案】A)

1.5 线性链表

考点 12 线性单链表的结构及其基本运算

1. 什么是线性链表

(1) 线性表顺序存储的缺点

①在一般情况下,要在顺序存储的线性表中插入一个新元素或删除一个元素时,为了保证插入或删除后的线性表仍然为顺序存储,则在插入或删除过程中需要移动大量的数据元素。因此采用顺序存储结构进行插入或删除的运算效率很低。

②当为一个线性表分配顺序存储空间后,如果出现线性表的存储空间已满,但还需要插入新的元素时就会发生“上溢”错误。

③计算机空间得不到充分利用,并且不便于对存储空间的动态分配。

(2) 线性表链式的基本概念

在定义的链表中,若只含有一个指针域来存放下一个元素地址,称这样的链表为单链表或线性链表。

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),如图 1-13 所示。

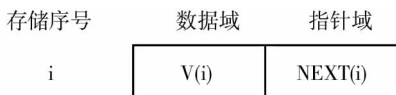


图 1-13 线性表的一个存储结点

2. 线性单链表的存储结构

用一组任意的存储单元来依次存放线性表的结点,这组存储单元既可以是连续的,也可以是不连续的,甚至是零散分布在内存中的任意位置上的。因此,链表中结点的逻辑次序和物理次序不一定相同。为了能正确表示结点间的逻辑关系,在存储每个结点值的同时,还必须存储指示其后件结点的地址(或位置)信息,这个信息称为指针(pointer)或链(link)。这两部分组成了链表中的结点结构,如图 1-14 所示。

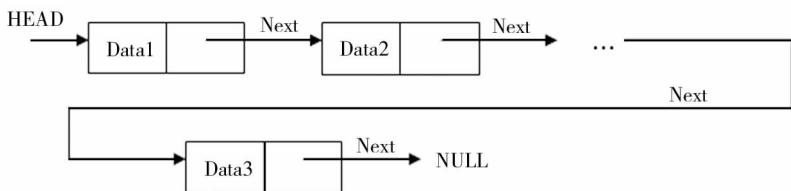


图 1-14 线性表的逻辑结构

链表正是通过每个结点的链域将线性表的 n 个结点按其逻辑次序链接在一起。由于上述链表的每一个结点只有一个链域,故将这种链表称为单链表(Single Linked)。

显然,单链表中每个结点的存储地址是存放在其前驱结点 Next 域中,而开始结点无前驱,故应设头指针 HEAD 指向开始结点。同时,由于终端结点无后件,故终端结点的指针域为空,即 NULL。

3. 带链的栈与队列

(1) 栈也是线性表,也可以采用链式存储结构。在实际应用中,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,如图 1-15 所示。

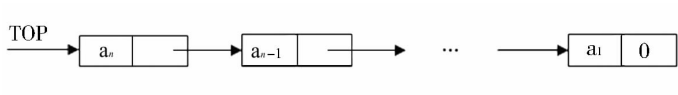


图 1-15 带链的栈

(2) 队列也是线性表,也可以采用链式存储结构,如图 1-16 所示。

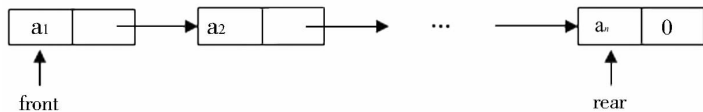


图 1-16 带链的队列

考点 13 线性链表的基本运算

线性链表的运算主要有以下几个:

- (1) 在线性链表中包含指定元素的结点之前插入一个新元素。
- (2) 在线性链表中删除包含指定元素的结点。
- (3) 将两个线性链表按要求合并成一个线性表。
- (4) 将一个线性链表按要求进行分解。

- (5) 逆转线性链表。
- (6) 复制线性链表。
- (7) 线性链表的排序。
- (8) 线性链表的查找。

1. 在线性链表中查找指定元素

对线性链表进行插入或删除的运算中,总是首先需要找到插入或删除的位置,这就需要对线性链表进行扫描查找,在线性链表中寻找包含指定元素的前一个结点。

在线性链表中,即使知道被访问结点的序号 i ,也不能像顺序表中那样直接按序号 i 访问结点,而只能从链表的头指针出发,顺链域 Next 逐个结点往下搜索,直到搜索到第 i 个结点为止。因此,链表不是随机存取结构。

在链表中,查找是否有结点值等于给定值 x 的结点,若有的话,则返回首次找到的其值为 x 的结点的存储位置;否则返回 NULL。查找过程从开始结点出发,顺着链表逐个将结点的值和给定值 x 作比较。

2. 线性链表的插入

线性链表的插入是指在链式存储结构下的线性链表中插入一个新元素。

插入运算是将值为 x 的新结点插入到表的第 i 个结点的位置上,即插入到 a_{i-1} 与 a_i 之间。因此,我们必须首先找到 a_{i-1} 的存储位置 p ,然后生成一个数据域为 x 的新结点 $*p$,并令结点 $*p$ 的指针域指向新结点,新结点的指针域指向结点 a_i ,如图 1-17 所示。

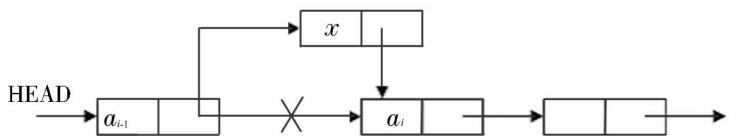


图 1-17 线性表的插入示意图

由线性链表的插入过程可以看出,由于插入的新结点取自于可利用栈,因此,只要可利用栈不空,在线性链表插入时总能取到存储插入元素的新结点,不会发生“上溢”的情况。而且,由于可利用栈是公用的,多个线性链表可以共享它,从而很方便地实现了存储空间的动态分配。另外,线性链表在插入过程中不发生数据元素移动的现象,只要改变有关结点的指针即可,从而提高了插入的效率。

3. 线性链表的删除

线性链表的删除是指在链式存储结构下的线性链表中删除包含指定元素的结点。

删除运算是将表的第 i 个结点删去。因为在单链表中结点 a_i 的存储地址是在其直接前趋结点 a_{i-1} 的指针域 Next 中,所以我们必须首先找到 a_{i-1} 的存储位置 p 。然后令 $p \rightarrow \text{Next}$ 指向 a_i 的直接后继结点,即把 a_i 从链上摘下。最后释放结点 a_i 的空间,如图 1-18 所示。

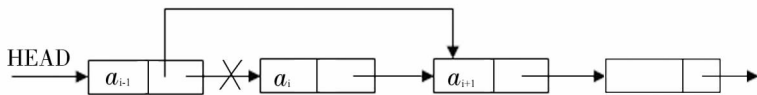


图 1-18 线性表的删除示意图

从线性链表的删除过程可以看出,从线性链表中删除一个元素后,不需要移动表中的数据元素,只要改变被删除元素所在结点的前一个结点的指针域即可。另外,由于可利用栈是用于收集计算机中所有的空闲结点,因此,当从线性链表中删除一个元素后,该元素的存储结点就变为空闲,应将空闲结点送回到可利用栈。

考点 14 线性双向链表的结构及其基本运算

1. 什么是双向链表

在单链表中,从某个结点出发可以直接找到它的直接后继,时间复杂度为 $O(1)$,但无法直接找到它的

构成了一个环状链。

在循环链表中,只要指出表中任何一个结点的位置,就可以从它出发访问到表中其他所有的结点,而线性单链表做不到这一点。

由于在循环链表中设置了一个表头结点,因此,在任何情况下,循环链表中至少有一个结点存在,从而使空表的运算统一。

真题题解

【例1】链表不具备的特点是()。

- A) 可随机访问任意一个结点
- B) 插入和删除不需要移动任何元素
- C) 不必事先估计存储空间
- D) 所需空间与其长度成正比

【命题目的】考查链表结构的数据结构的基本特点。

【解题要点】顺序表可以随机访问任意一个结点,而链表必须从第一个数据结点出发,逐一查找每个结点。所以答案为A)。

【错解分析】链表结构是一些逻辑上相邻,而空间上并不一定相邻的数据元素的集合,相邻的结点之间通过指针相互联系,在插入和删除元素时,只需修改结点指针即可,不需要移动数据元素。当存储空间不足时,可以动态为其分配内存空间,所以不必估计存储空间的大小。

【考点链接】链表的插入元素和删除元素的算法及其复杂度。

【答案】A)

【例2】对线性表,在下列情况下应当采用链表表示的是()。

- A) 经常需要随机地存取元素
- B) 经常需要进行插入和删除操作
- C) 表中元素需要占据一片连续的存储空间
- D) 表中元素的个数不变

【命题目的】考查链表的插入和删除。

【解题要点】由于链表不是顺序存储的,即当进行插入和删除操作时不需要移动大量的元素,所以当一些需要经常进行插入和删除操作的情况应当采用链表的方式存储,答案应是B)。

【错解分析】经常需要随机地存取元素的情况适合采用顺序存储的数组来表示,因为可以通过数组下标来随机访问,所以A)选项错误;表中元素需要占据一片连续的存储空间正是顺序存储的特点,所以C)选项也不正确;数组在一开始就分配固定个数的空间,不能动态分配,而链表的元素是可以动态改变的,所以D)选项也不正确。

【考点链接】链表的存储方式及它们的插入和删除操作。

【答案】B)

【例3】如果最常用的操作是取第*i*个结点及其前驱,最节省时间的存储方式是()。

- A) 单链表
- B) 双向链表
- C) 单循环链表
- D) 顺序表

【命题目的】考查几种存储结构的特点。

【解题要点】本题实现上查找第*i*-1个结点和第*i*个结点,在前3个选项中,查找结果的时间复杂度均为 $O(n)$,在顺序表中查找的时间复杂度为 $O(1)$ 。所以答案为D)。

【错解分析】由于链表结构在空间存储上的不连续性,在查找某个结点时,需要从当前结点开始向前或者向后逐个比较查找,浪费时间。所以A)选项、B)选项和C)选项都是错误的。

【考点链接】单链表、双向链表、单循环链表和顺序表查找某个结点的算法。

【答案】D)

【例4】与单链表相比,双向链表的优点之一是()。

- A) 插入、删除操作更加简单
- B) 可以随机访问
- C) 可以省略表头指针或表尾指针
- D) 顺序访问相邻结点更加灵活

【命题目的】考查双向链表与单向链表相比的优点。

【解题要点】在双向链表中由于每个结点包括两个指针域,其中一个指向该结点的前一个结点,另一个指向该结点的后一个结点。因此它既可以访问前驱结点,又可以访问后继结点,而单链表每个结点只有一个指针域,指向它的后继结点,所

以它只能直接访问它的下一个结点,而无法直接访问它的前一个结点。所以答案为 D)。

【错解分析】由于双向链表比单链表结构复杂,所以在插入和删除元素时,要修改更多的指针域,相对比较复杂,所以 A) 选项错误。单向链表和双向链表在空间存储上的不连续性决定了两者都不可以随机访问,所以 B) 选项错误。

【考点链接】访问单向链表和双向链表结点的算法,单循环链表和双向循环链表的特点。

【答案】D)

【例 5】不带头结点的单链表 head 为空的判定条件是()。

A) head = NULL

B) head -> next = NULL

C) head -> next = head

D) head! = NULL

【命题目的】考查判断链式结构栈(不带头结点)是否为空的条件。

【解题要点】在不带头结点的单链表 head 中,head 指向第一个元素结点,head = NULL 表示该链表为空,所以答案为 A)。

【错解分析】对于 B) 选项,表示带头结点的单链表为空的判断条件。对于带头结点的单链表,头结点所指的下一个结点表示单链表的第一个结点,如果为 NULL,则表示链表为空。对于 C) 选项,表示带头结点的单循环链表为空的判断条件。对于带头结点的单循环链表,如果 head 所指的下一个结点是它自身,则表示链表为空。

【考点链接】判断带头结点的链表是否为空的条件。

【答案】A)

【例 6】可以用带头结点的链表表示线性表,也可用不带头结点的链表表示线性表,前者最主要的好处是()。

A) 可以加快对表的遍历

B) 使空表和非空表的处理统一

C) 节省存储空间

D) 可以提高存取表元素的速度

【命题目的】考查带头结点的线性链表和不带头结点的线性链表的插入和删除操作。

【解题要点】用带头结点的链表表示线性表时,空表和非空表的插入、删除是相同的。当往空链表插入元素时,只要把待插入元素的指针域指向头结点的指针域,把头结点的指针域指向新增元素即可,当往非空链表插入元素时只要找到插入的位置,执行同样的操作即可完成插入。当链表只有一个元素时,删除操作只要修改指针指向下一个元素的指针所指的元素即可,跟一般的链表删除操作是一样的。

【错解分析】带头结点的链表并不能加快对链表的遍历,所以 A) 选项是错误的;带头结点的链表反而要增加一个用于存储头结点的空间,并不能节省存储空间,C) 选项是错误的;用带头结点的链表跟存取元素的速度无关,所以 D) 选项也是错误的。

【考点链接】线性链表的插入和删除操作。

【答案】B)

【例 7】带头结点的双向循环链表 L 为空的条件是()。

A) L = = NULL

B) L -> next = = NULL

C) L -> prior = = NUL

D) L -> next = = L

【命题目的】考查判断双向链式结构栈(带头结点)是否为空的条件。

【解题要点】带头结点的双向循环链表 L 为空的条件是:只有该头结点,即 L -> next = = L 或者 L -> prior = = L。所以答案为 D)。

【错解分析】由于是带头结点双向循环链表,至少存在一个头结点,所以 L、L -> prior 和 L -> next 都不可能为 NULL。

【考点链接】判断不带头结点的双向循环链表是否为空的条件。

【答案】D)

【例 8】向一个栈顶指针为 HS 的链式栈中插入一个 s 所指的结点时,则执行()。

A) HS -> next = s;

B) s -> next = HS -> next; HS -> next = s;

C) s -> next = HS; HS = s;

D) s -> next = HS; HS = HS -> next;

【命题目的】考查链式栈的入栈运算的算法。

【解题要点】向链式栈中插入一个结点,就是在单链表的表头插入一个结点,同时将新结点的位置赋予栈顶指针。所以答案为 C)。

【错解分析】对于链式栈,HS 为指向栈顶的指针。对于 A) 选项,实际上是将 s 赋值给了栈顶的下一个元素;对于 B) 选项,执行后原来的链式栈结构已经不再是线性结构;对于 D) 选项,HS 指向原来栈顶的下一个元素。

【考点链接】删除链式栈中栈顶元素的算法。

【答案】C)

【例 9】在一个链式队列中,假设 f 和 r 分别为队头和队尾指针,则插入 s 所指结点的运算是()。

A) $f \rightarrow next = s; f = s;$

B) $r \rightarrow next = s; r = s;$

C) $s \rightarrow next = r; r = s;$

D) $s \rightarrow next = f; f = s;$

【命题目的】考查链式队列入队运算的算法。

【解题要点】向链式队列中插入一个结点就是在单链表表尾插入一个结点,同时新插入的结点成为表尾结点。所以答案为 B)。

【错解分析】对于链式队列结构,插入元素是在队尾进行的,只需修改队尾指针,不需修改队头指针。而 A) 选项、D) 选项都修改了队头指针。

【考点链接】在链式队列中删除一个结点的算法。

【答案】B)

【例 10】在一个链式队列中,假设 f 和 r 分别为队头和队尾指针,则删除结点的运算是()。

A) $r = f \rightarrow next;$

B) $r = r \rightarrow next;$

C) $f = f \rightarrow next;$

D) $f = r \rightarrow next;$

【命题目的】考查链式队列出队运算的算法。

【解题要点】从链式队列中删除一个结点就是删除单链表的表头结点,同时原表头结点的下一个结点成为新的表头结点。所以答案为 C)。

【错解分析】对于链式队列结构,删除元素是在队头进行的,只需修改队头指针,不需修改队尾指针。而 A) 选项、B) 选项都修改了队尾指针。

【考点链接】在链式队列中插入一个新的结点的算法。

【答案】C)

【例 11】下面关于线性表的叙述中,错误的是()。

A) 线性表采用顺序存储,必须占用一片连续的存储单元

B) 线性表采用顺序存储,便于进行插入和删除操作

C) 线性表采用链接存储,不必占用一片连续的存储单元

D) 线性表采用链接存储,便于进行插入和删除操作

【命题目的】考查顺序存储和链接存储的差别。

【解题要点】采用顺序存储的线性表,当对其进行插入和删除操作时需要移动大量的元素,所以 B) 选项说法是错误的,答案选 B)。

【错解分析】顺序存储结构必须占用一片连续的存储单元,A) 选项说法是正确的;当线性表采用链接存储时,其占用的存储空间是不连续的,C) 选项说法也是正确的;正是由于链接存储不是占用一片连续的存储空间,所以便于进行插入和删除操作。

【考点链接】链表的存储结构及针对其进行的操作。

【答案】B)

【例 12】对于 n 个结点的单向链表(无表头结点),需要指针单元的个数至少为()。

A) $n - 1$

B) n

C) $n + 1$

D) $2n$

【命题目的】考查链表结构的掌握。

【解题要点】在 n 个结点的单向链表(无表头结点)中,每个结点都有一个指针单元(即指针域),加上头指针,至少需要 $n + 1$ 个指针单元。所以答案为 C)。

【错解分析】忽略了最后结点或头结点的指针。

【考点链接】单向链表的特点。

【答案】C)

1.6 树与二叉树

考点 16 树的定义

树是由 $n(n \geq 0)$ 个结点组成的有限集合。若 $n = 0$,称为空树;若 $n > 0$,则:

(1)有一个特定的称为根(root)的结点。它只有直接后件,但没有直接前件;

(2)除根结点以外的其他结点可以划分为 $m(m \geq 0)$ 个互不相交的有限集合 T_0, T_1, \dots, T_{m-1} , 每个集合 $T_i(i=0, 1, \dots, m-1)$ 又是一棵树, 称为根的子树, 每棵子树的根结点有且仅有一个直接前件, 但可以有 0 个或多个直接后件, 如图 1-23 所示。



图 1-23 树的结构示意图

树型结构具有如下特点：

- (1) 每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, 简称为树的根。
- (2) 每一个结点可以有多个后件, 它们都称为该结点的子结点。没有后件的结点称为叶子结点。
- (3) 一个结点所拥有的后件个数称为树的结点数。
- (4) 树的最大层次称为树的深度。

在计算机中, 可以用树结构来表示算术表达式, 用树来表示算术表达式的原则是:

- (1) 表达式中的每一个运算符在树中对应一个结点, 称为运算符结点。
- (2) 运算符的每一个运算对象在树中为该运算符结点的子树(在树中的顺序为从左到右)。
- (3) 运算对象中的单变量均为叶子结点。

树在计算机中通常用多重链表表示。

考点 17 二叉树的定义及其基本性质

1. 什么是二叉树

二叉树(binary tree)是由 $n(n \geq 0)$ 个结点的有限集合构成, 此集合或者为空集, 或者由一个根结点及两棵互不相交的左右子树组成, 并且左右子树都是二叉树。二叉树可以是空集合, 根可以有空的左子树或空的右子树。二叉树不是树的特殊情况, 它们是两个概念。

二叉树具有如下两个特点:

- (1) 非空二叉树只有一个根结点。
- (2) 每一个结点最多有两棵子树, 且分别称为该结点的左子树与右子树。

二叉树的每个结点最多有两个孩子, 或者说, 在二叉树中, 不存在度大于 2 的结点, 并且二叉树是有序树(树为无序树), 其子树的顺序不能颠倒, 因此, 二叉树有 5 种不同的形态, 如图 1-24 所示。

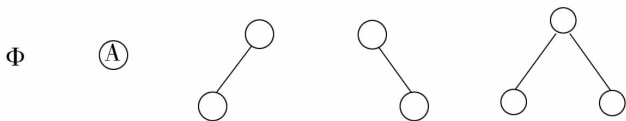


图 1-24 二叉树的 5 种不同形态

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, 该结点即是叶子结点。

2. 二叉树的基本性质

性质 1: 在二叉树的第 k 层上至多有 2^{k-1} 个结点 ($k \geq 1$)。

性质 2: 深度为 m 的二叉树至多有 $2^m - 1$ 个结点。

深度为 m 的二叉树的最大的结点数是为二叉树中每层上的最大结点数之和, 由性质 1 得到最大结点数。

$$\sum_{i=1}^m (k \text{ 层上最多结点数}) = \sum_{i=1}^m 2^{i-1} = 2^m - 1$$

性质3: 对任何一棵二叉树,度为0的结点(即叶子结点)总是比度为2的结点多一个。

如果叶子结点 n_0 ,度为2的结点数为 n_2 ,则 $n_0 = n_2 + 1$ 。

设二叉树中度为1的结点数为 n_1 ,二叉树中总结点数为 N ,因为二叉树中所有结点均小于或等于2,所以有

$$N = n_0 + n_1 + n_2 \quad (1)$$

再看二叉树中的分支数,除根结点外,其余结点都有一个进入分支,设 m 为二叉树中的分支总数,则有

$$N = m + 1 \quad (2)$$

又由于二叉树中这 m 个分支是分别由非叶子结点射出的。其中度为1的每个结点射出1个分支,度为2的每个结点射出2个分支。因此,二叉树中所有度为1与度为2的结点射出的分支总数为 $n_1 + 2n_2$,而在二叉树中,总的射出分支数应与总的进入分支数相等,即

$$m = n_1 + 2n_2 \quad (3)$$

将(3)代入(2)式有

$$N = n_1 + 2n_2 + 1$$

比较(1)和(4)并化简得

$$n_0 = n_2 + 1$$

性质4: 具有 n 个结点的完全二叉树的深度至少为 $\lceil \log_2 n \rceil + 1$,其中 $\lceil \log_2 n \rceil$ 表示 $\log_2 n$ 的整数部分。

3. 满二叉树与完全二叉树

(1) 满二叉树

满二叉树是指这样的一种二叉树:除最后一层外,每一层上的所有结点都有两个子结点。深度为 k 的二叉树具有 $2^k - 1$ 个结点。即在满二叉树的第 k 层上有 2^{k-1} 个结点。

从上面满二叉树定义可知,必须是二叉树的每一层上的结点数都达到最大,否则就不是满二叉树。深度为 m 的满二叉树有 $2^m - 1$ 个结点。

(2) 完全二叉树

完全二叉树是指这样的二叉树: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。

如果一棵具有 n 个结点的深度为 k 的二叉树,它的每一个结点都与深度为 k 的满二叉树中编号为 $1 \sim n$ 的结点一一对应。

如图1-25所示,为满二叉树和完全二叉树的结构比较。

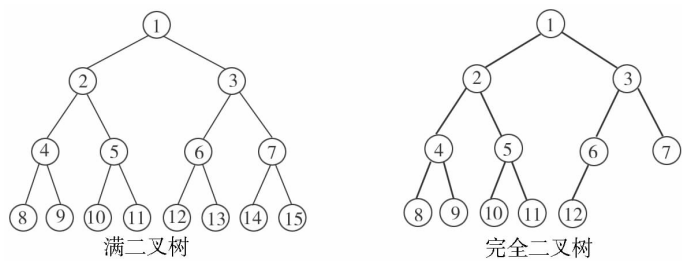


图 1-25 满二叉树与完全二叉树结构

从完全二叉树定义可知,结点的排列顺序遵循从上到下、从左到右的规律。所谓从上到下,表示本层结点数达到最大后,才能放入下一层。从左到右,表示同一层结点必须按从左到右排列,若左边空一个位置时不能将结点放入右边。完全二叉树除最后一层外每一层的结点数都达到最大值,最后一层只缺少右边的若干结点。

满二叉树也是完全二叉树,反之完全二叉树不一定是满二叉树。

性质5: 具有 n 个结点的完全二叉树深度为 $\lceil \log_2 n \rceil + 1$ 或 $\lceil \log_2 (n + 1) \rceil$ 。

性质6: 如果在一棵有 n 个结点的完全二叉树的结点按层序编号(从第1层到第 $\lceil \log_2 n \rceil + 1$ 层,每层从左到右),则对任一结点 i ($1 \leq i \leq n$),有:

(1) 如果 $i = 1$,则结点 i 无双亲,是二叉树的根;如果 $i > 1$,则其双亲是结点 $\lceil i/2 \rceil$;

(2) 如果 $2i \leq n$, 则结点 i 为叶子结点, 无左孩子; 否则, 其左孩子是结点 $2i$;

(3) 如果 $2i + 1 \leq n$, 则结点 i 无右孩子; 否则, 其右孩子是结点 $2i + 1$ 。

4. 二叉树的存储结构

在计算机中, 二叉树通常采用链式存储结构。用于存储二叉树中各元素的存储结点由两部分组成: 数据域与指针域。但在二叉树中, 由于每一个元素可以有两个后件(两个子结点), 因此, 用于存储二叉树的存储结点的指针域有两个: 一个用于指向该结点的左子结点的存储地址, 称为左指针域; 另一个用于指向该结点的右子结点的存储地址, 称为右指针域, 如图 1-26 所示。

Lchild	Value	Rchild
L(i)	V(i)	R(i)

图 1-26 二叉树的存储结构

考点 18 二叉树的遍历

所谓遍历二叉树, 就是遵从某种次序, 访问二叉树中的所有结点, 使得每个结点仅被访问一次。

1. 前序遍历

前序遍历是指在访问根结点、遍历左子树与遍历右子树这三者中, 首先访问根结点, 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。前序遍历描述为:

若二叉树为空, 则执行空操作。否则: ①访问根结点; ②前序遍历左子树; ③前序遍历右子树。

2. 中序遍历

中序遍历是指在访问根结点、遍历左子树与遍历右子树这三者中, 首先遍历左子树, 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。中序遍历描述为:

若二叉树为空, 则执行空操作。否则: ①中序遍历左子树; ②访问根结点; ③中序遍历右子树。

3. 后序遍历

后序遍历是指在访问根结点、遍历左子树与遍历右子树这三者中, 首先遍历左子树, 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。后序遍历描述为:

若二叉树为空, 则执行空操作。否则: ①后序遍历左子树; ②后序遍历右子树; ③访问根结点。

真题题解

【例 1】树最适合用来表示()。

A) 有序数据元素

B) 无序数据元素

C) 元素之间具有分支层次关系的数据

D) 元素之间无联系的数据

【命题目的】考查树结构的基本特点和数据结构的表示。

【解题要点】由树的一对多特点可知, 本题答案为 C)。

【错解分析】对树结构的认识和理解不够深入, 容易误选答案。

【考点链接】树的数据结构特点和表示以及树的一些相关基本概念。

【答案】C)

【例 2】下列有关树的概念错误的是()。

A) 一棵树中只有一个无前驱的结点

B) 一棵树的度为树中各个结点的度数之和

C) 一棵树中,每个结点的度数之和等于结点总数减 1

D) 一棵树中每个结点的度数之和与边的条数相等

【命题目的】考查对树的结点与树的度的掌握。

【解题要点】树的度是指树内各结点的度的最大值,B) 选项的说法是错误的。

【错解分析】一棵树中除根结点之外,每个结点都有一个前驱结点,A) 选项说法是正确的;结点拥有子树的个数称为结点的度,所以结点的度数之和即为除根结点外所有结点的个数,即每个结点的度数之和等于结点总数减 1,C) 选项的说法也是正确的;结点的度即是用于子树的个数,而结点与子树之间是以边连接的,所以一棵树中每个结点的度数之和与边的条数相等,D) 选项说法是正确的。

【考点链接】二叉树的概念、性质。

【答案】B)

【例 3】下面关于二叉树的叙述正确的是()。

A) 一棵二叉树中叶子结点的个数等于度为 2 的结点个数加 1

B) 一棵二叉树中的结点个数大于 0

C) 二叉树中任何一个结点要么是叶,要么恰有两个子女

D) 二叉树中,任何一个结点的左子树和右子树上的结点个数一定相等

【命题目的】考查二叉树的性质。

【解题要点】由二叉树的性质知二叉树叶子的个数 n_0 和度为 2 的结点个数 n_2 的关系为 $n_0 = n_2 + 1$ 。

【错解分析】二叉树的结点个数可以等于 0,所以 B) 选项错误;二叉树中有些不是叶子的结点只有一个子女,所以 C) 选项也不对;D) 选项显然是错误的。

【考点链接】二叉树的概念性质。

【答案】A)

【例 4】如图 1-27 所示的二叉树,其中序遍历的结果为()。

A) abcdef

B) abdefc

C) dbefac

D) defbca

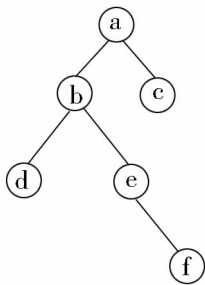


图 1-27 二叉树中序遍历

【命题目的】考查二叉树中序遍历的算法。

【解题要点】中序遍历的思想是,若二叉树为空,则空操作;否则

(1) 中序遍历左子树;

(2) 访问根结点;

(3) 中序遍历右子树。

【错解分析】混淆了二叉树的 3 种遍历方式的算法,容易误选答案。

【考点链接】二叉树的前序遍历、后序遍历和中序遍历。

【答案】C)

【例 5】设 n, m 为一棵二叉树上的两个结点,在中序遍历中, n 在 m 前的条件是()。

A) n 在 m 右子树上

B) n 是 m 的祖先

C) n 在 m 左子树上

D) n 是 m 的子孙

【命题目的】考查二叉树中序遍历的算法和特点。

【解题要点】中序遍历时,先访问左子树,再访问根结点。 n 在 m 前,则 n 必须在 m 的左子树中。

【错解分析】对于 A) 选项,如果 n 在 m 的右子树上,根据中序遍历算法,先访问根结点 m ,然后再访问右子树上的结点,所以 n 必然要在 m 后;对于 B) 选项, n 是 m 的祖先,则 m 可能在 n 的左子树上,也可能在 n 的右子树上,如果 m 在 n 的左子树

上,根据中序遍历算法,先访问 n 的左子树,然后访问 n 结点,所以 n 必然要在 m 后,所以 B) 选项错误;对于 D) 选项, n 是 m 的子孙,则 n 可能在 m 的左子树上,也可能在 m 的右子树上,如果 n 在 m 的右子树上,由 A) 选项的分析可知,D) 选项是错误的。

【考点链接】二叉树前序遍历和后序遍历的算法和特点。

【答案】C)

【例 6】已知某二叉树的后序遍历序列是 DACBE,中序遍历序列是 DEBAC,则它的前序遍历序列是()。

- A) ACBED B) DEABC C) DECAB D) EDBAC

【命题目的】二叉树 3 种遍历方式的特点。

【解题要点】后序遍历的顺序是“左子树 - 右子树 - 根结点”;中序遍历顺序是“左子树 - 根结点 - 右子树”;前序遍历顺序是“根结点 - 左子树 - 右子树”。根据各种遍历算法,不难得出前序遍历序列是 EDBAC。所以答案为 D)。

【错解分析】对二叉树的 3 种遍历方法没有正确掌握和理解,很难解决问题。

【考点链接】已知二叉树的两种遍历序列,求其第三种遍历序列。

【答案】D)

【例 7】一棵二叉树的前序遍历序列为 ABDGCFK,中序遍历序列为 DGBAFCK,则结点的后序遍历序列是()。

- A) ACFKDBG B) GDBFKCA C) KCFAGDB D) ABCDFKG

【命题目的】二叉树三种遍历方式的特点。

【解题要点】前序遍历的顺序是“根结点 - 左子树 - 右子树”,由题目可知,结点 A 必然为二叉树的根结点,中序遍历的顺序是“左子树 - 根结点 - 右子树”,所以 DGB 为左子树的结点,FCK 为右子树的结点,后序遍历的顺序是“左子树 - 右子树 - 根结点”,所以根结点必然为后序遍历序列的最后一个结点。所以本题答案为 B)。

【错解分析】对二叉树的三种遍历方法没有正确掌握和理解,很难解决问题。

【考点链接】已知二叉树的两种遍历序列,求其第三种遍历序列。

【答案】B)

【例 8】二叉树前序遍历和中序遍历序列如下:

前序遍历序列:EFHI'JK

中序遍历序列:HFIEJK'

则该二叉树根结点的右子树的根为:()。

- A) E B) F C) ' D) H

【命题目的】考查由二叉树的前序遍历序列和中序遍历序列判断二叉树的形状。

【解题要点】前序遍历是“根结点 - 左子树 - 右子树”,中序遍历是“左子树 - 根结点 - 右子树”。由前序遍历的序列可知 E 为该二叉树的树根结点,再结合中序遍历结点序列可知右子树的组成结点(E 后面的结点)为 JK',而这三个结点在前序遍历中'为第一个结点,即右子树的根结点。所以答案为 C)。

【错解分析】先序遍历根结点肯定在前面,E 是二叉树的根结点,所以 A) 选项错误。中序遍历则分左子树和右子树。F 和 H 在 E 的左边,所以 B) 选项和 D) 选项也是错误的。

【考点链接】已知二叉树的任意两种遍历序列,判断二叉树的形状。

【答案】C)

【例 9】设二叉树根结点的层次为 0,一棵树深为 h 的满二叉树中结点的个数是()。

- A) 2^h B) 2^{h-1} C) $2^h - 1$ D) $2^{h+1} - 1$

【命题目的】考查满二叉树树深与结点的关系。

【解题要点】当根结点的层次为 1 时,二叉树的结点数 n 和深度 h 之间的关系是 $n = 2^h - 1$,所以当二叉树的根结点的层次为 0 时,结点和树深的关系是 $n = 2^{h+1} - 1$,所以答案为 D)。

【错解分析】由以上分析知 A)、B) 和 C) 均是错误的。

【考点链接】二叉树的性质。

【答案】D)

【例 10】有关二叉树的下列说法正确的是()。

- A) 二叉树的度为 2 B) 一棵二叉树的度可以小于 2
C) 二叉树中任何一个结点的度都为 2 D) 任何一棵二叉树中至少有一个结点的度为 2

【命题目的】考查有关二叉树的基本概念和结构。

【解题要点】根据二叉树的一些概念,不难得知 A) 选项、C) 选项和 D) 选项均为错误答案。所以答案为 B)。

【错解分析】A) 选项错误, 因为二叉树不是度为 2 的树, 不能断章取义; C) 选项错误, 因为二叉树的结点的度可以为 1; D) 选项错误, 二叉树可以有所有结点为 1 的情况。

【考点链接】二叉树的概念、性质。

【答案】 B)

【例 11】如图 1-28 所示的 4 棵二叉树中,不是完全二叉树的是()。

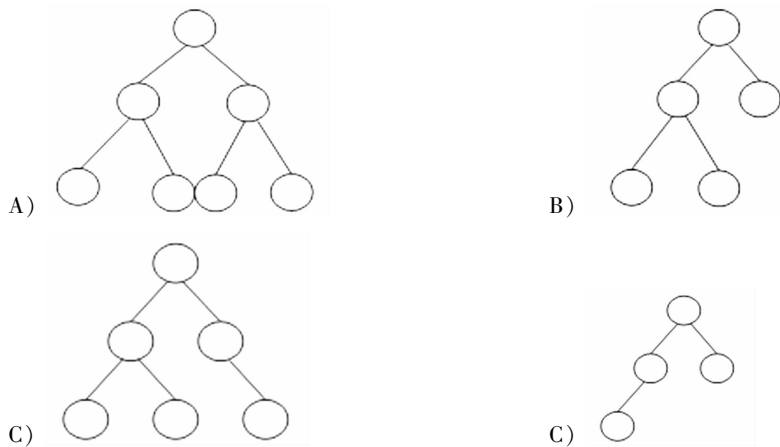


图 1-28 完全二叉树

【命题目的】考查完全二叉树的概念和结构。

【解题要点】如果一棵具有 n 个结点的深度为 k 的二叉树,它的每一个结点都与深度为 k 的满二叉树中编号为 $1 \sim n$ 的结点一一对应,则称这棵二叉树为完全二叉树。

【错解分析】只有 B) 选项不是完全二叉树, 由于没有掌握完全二叉树的概念, 导致选择其他错误答案。

【考点链接】完全二叉树和满二叉树的判断、区别和联系。

【答案】C)

【例 12】设深度为 h 的二叉树上只有度为 0 和度为 2 的结点, 则此二叉树中所包含的结点数至少为()。

- A) $2h$
B) $2h - 1$
C) $2h + 1$
D) $h + 1$

【命题目的】考查有关二叉树的基本结构及其结点度之间的关系。

【解题要点】结点最少的情况如图 1-29 所示,除根结点层只有 1 个结点外,其余 $h-1$ 层均有两个结点,结点总数 $= 2(h-1) + 1 = 2h-1$ 。

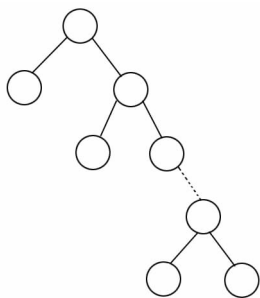


图 1-29 度只有 0 和 2 的二叉树

【错解分析】由二叉树的深度和结点数的关系可以知道 A) 选项、C) 选项和 D) 选项都是错误的。

【考点链接】二叉树叶子、结点、深度的关系计算。

【答案】B)

【例 13】如图 1-30 所示二叉树的中序遍历序列是()。

- A) abcdefg B) dfbagc
C) dbaefcg D) abcdefg

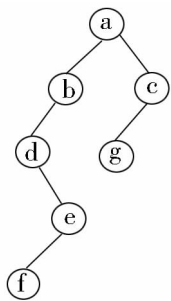


图 1-30 二叉树中序遍历

【命题目的】考查二叉树中序遍历的算法。

【解题要点】先遍历所有左子树,再遍历根结点,最后遍历右子树,所以本题的答案为 B 选项。

【错解分析】由二叉树的中序遍历算法知,A) 选项、C) 选项和 D) 选项都是错误的。

【考点链接】二叉树的前序遍历、后序遍历和中序遍历算法。

【答案】B)

【例 14】某二叉树的先序和后序遍历序列正好相反,则该二叉树一定是()。

A) 空或只有一个结点

B) 完全二叉树

C) 二叉排序树

D) 深度等于其结点数

【命题目的】考查对二叉树的遍历序列的熟练掌握。

【解题要点】对于深度等于其结点数的二叉树,每层只有一个结点,假设从上向下分别为 a_1, a_2, \dots, a_n ,则先序遍历序列为 a_1, a_2, \dots, a_n 。后序遍历序列为 a_n, a_{n-1}, \dots, a_1 。本题答案为 D) 选项。

【错解分析】A) 选项不存在相反不相反,B) 选项和 C) 选项的遍历序列左子树总是在根结点的左边,右子树总是在根结点的右边。

【考点链接】二叉树的中序、先序、后序遍历序列。

【答案】D)

【例 15】按照二叉树的定义,具有 3 个结点的二叉树有()种。

A) 3

B) 4

C) 5

D) 6

【命题目的】根据二叉树结点数目判断二叉树的种类。

【解题要点】具有 3 个结点的二叉树总数为 5,本题答案为 C。

【错解分析】A) 选项和 B) 选项遗漏了一些二叉树的可能组成情况,D) 选项把不是二叉树的情况也当成了是二叉树的情况了,都是错误的。

【考点链接】规定结点下二叉树的可能组合情况。

【答案】C)

【例 16】深度为 5 的二叉树至多有()个结点。

A) 16

B) 32

C) 31

D) 10

【命题目的】考查有关二叉树的基本知识。

【解题要点】最多结点的情况为满二叉树,由二叉树的性质知,此时结点个数 $= 2^5 - 1 = 31$ 。本题答案为 C)。

【错解分析】由满二叉树的结点和树深的换算关系 $n = 2^h - 1$,知 A) 选项、B) 选项和 D) 选项是错误的,只有 C) 选项才是正确的。

【考点链接】满二叉树树深与结点的关系。

【答案】C)

【例 17】假定根结点的层次是 0,含有 15 个结点的二叉树的最小树深是()。

A) 4

B) 5

C) 3

D) 6

【命题目的】考查完全二叉树的树深和结点关系的掌握情况。

【解题要点】要使二叉树在规定结点数下的深度最小,这样的二叉树只能是完全二叉树。当根结点的层次为 1 时,二叉树的结点 n 和深度 h 之间的关系是 $n = 2^h - 1$,所以当满二叉树的根结点的层次为 0 时,结点和树深的关系是 $n = 2^{h+1} - 1$,所以答案为 C)。

【错解分析】A) 选项是头结点层次为 1 的情况,题目假定头结点的层次为 0;B) 选项和 D) 选项显然不符合题目要求。

【考点链接】二叉树的性质。

【答案】C)

【例 18】在一非空二叉树的中序遍历序列中,根结点的右边()。

- A) 只有右子树上的所有结点 B) 只有右子树上的部分结点
C) 只有左子树上的部分结点 D) 只有左子树上的所有结点

【命题目的】考查二叉树中序遍历序列的特点。

【解题要点】在中序遍历序列中,根结点将左右树分开,左边为左子树中的结点,右边为右子树的结点。所以本题的答案为 A)。

【错解分析】二叉树的中序遍历序列中,左子树在根结点的左边,右子树在根结点的右边,所以 B) 选项、C) 选项和 D) 选项都是错误答案。

【考点链接】二叉树前序遍历和后序遍历序列的特点。

【答案】A)

【例 19】任何一棵二叉树的叶子结点在先序、中序和后序遍历序列中的相对次序()。

- A) 不发生改变 B) 发生改变
C) 不能确定 D) 以上都不对

【命题目的】考查二叉树三种遍历序列的特点。

【解题要点】在先序、中序、后序遍历序列中叶子结点总是从左向右的。本题的答案为 A)。

【错解分析】叶子结点的次序总是由左到右的次序排列的,因此只有 A) 选项正确,其他都是错误的。

【考点链接】二叉树先序、中序、后序遍历序列中各元素的相对次序。

【答案】A)

【例 20】对一个满二叉树, m 个树叶, n 个结点,深度为 h ,则()。

- A) $n = h + m$ B) $h + m = 2n$
C) $m = h - 1$ D) $n = 2^h - 1$

【命题目的】考查有关二叉树的一些基本结论。

【解题要点】对于深度为 h 的满二叉树, $n = 2^0 + 2^1 + 2^2 + \cdots + 2^{h-1} = 2^h - 1 = 2$ 。本题答案为 D)。

【错解分析】由满二叉树的树深和结点的关系知,A) 选项、B) 选项和 C) 选项都是错误的。

【考点链接】满二叉树树叶、结点、深度的关系计算公式。

【答案】D)

【例 21】以下各组序列不属于堆的是()。

- A) (100,85,98,77,80,60,82,40,20,10,66)
B) (100,98,85,82,80,77,66,60,40,20,10)
C) (10,20,40,60,66,77,80,82,85,98,100)
D) (100,85,40,77,80,60,66,98,82,10,20)

【命题目的】考查堆的概念、性质。

【解题要点】若有 n 个元素的序列 $h_1, h_2, h_3, \cdots, h_n$, 当满足如下条件:

$$\textcircled{1} \begin{cases} h_i \leq h_{2i} \\ h_i \leq h_{2i+1} \end{cases} \quad \text{或} \quad \textcircled{2} \begin{cases} h_i \geq h_{2i} \\ h_i \geq h_{2i+1} \end{cases} \quad \text{其中 } i = 1, 2, \cdots, n/2. \text{ 则称此 } n \text{ 个元素的序列 } h_1, h_2, h_3, \cdots, h_n \text{ 为一个堆。若将此排序码按}$$

顺序组成一棵完全二叉树,则①称为小根堆(二叉树的所有根结点值小于或等于左右孩子的值),②称为大根堆(二叉树的所有根结点值大于或等于左右孩子的值)。

【错解分析】小根堆或大根堆的转换为二叉树,再由二叉树先序遍历时,数值的排列方式有两种情况,一种是由小到大排列或由大到小排列,另外一种则是当二叉树的左孩子的值大于右孩子的值时,会使由小到大或由大到小排列的顺序数据中有两个数据的顺序发生交换的现象如 A) 选项中的 85 和 98。由上所述知,A) 选项、B) 选项和 C) 选项都是堆。

【考点链接】堆与二叉树的转换;二叉树、堆到具体数据的转换。

【答案】D)

1.7 查找技术

考点 19 顺序查找与二分查找算法

1. 顺序查找

顺序查找是一种最简单的查找方法,它的基本思想是:从表的一端开始,顺序扫描线性表,依次将扫描到的结点关键字和待找的值 k 相比较,若相等,则查找成功;若整个表扫描完毕,仍未找到关键字等于 K 的元素,则查找失败。

顺序查找的优点是算法简单,对表结构无任何要求,无论是用向量还是用链表来存放结点,也无论结点之间是否按关键字有序或无序,它都同样适用。顺序查找的缺点是查找效率低,当 n 较大时,不宜采用顺序查找,而必须寻求更好的查找方法。

但在下列两种情况下只能采用顺序查找:

(1)如果线性表是无序表(即表中的元素是无序的),则不管是顺序存储结构还是链式存储结构,都只能用顺序查找。

(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. 二分查找

二分查找,也称折半查找,它是一种高效率的查找方法。但二分查找有条件限制:要求表必须用顺序存储结构,且表中元素必须按关键字有序(升序或降序均可)排列。我们不妨假设表中元素为升序排列。

设有序线性表长度为 n ,被查找元素为 x ,则二分查找的方法如下:

将 x 与线性表的中间项比较:

- (1)若 x 的值与中间项的值相等,则说明查到,查找结束;
- (2)若 x 小于中间项的值,则在线性表的前半部分以相同的方法查找;
- (3)若 x 大于中间项的值,则在线性表的后半部分以相同的方法查找。

若查找进行到子表长度为 0,说明线性表中没有这个元素。

假设给定有序表中关键字为 8,17,25,44,68,77,98,100,115,125,将查找 $k=17$ 的情况描述为如图 1-31 所示。

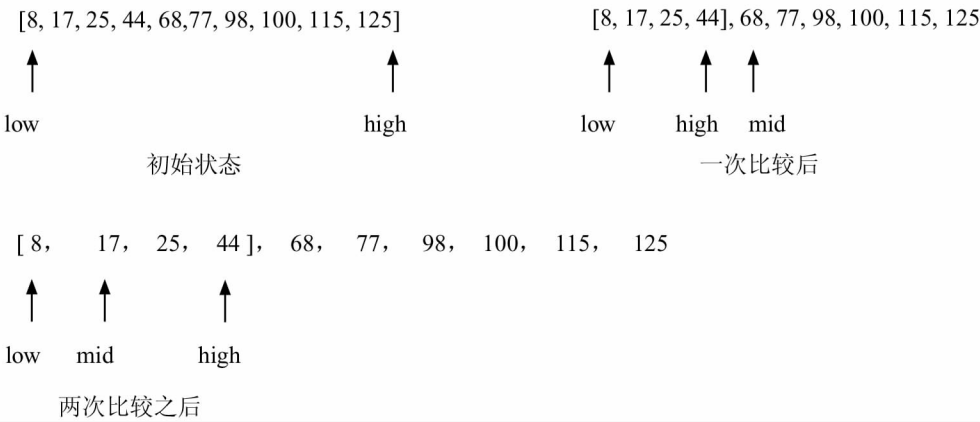


图 1-31 二分查找情况示意图

表按相同的方法继续查找。与11比较的关键码分别为15、8、10、12。比较的次数为4。所以答案为C)。

【错解分析】没有正确掌握二分法查找的过程,从而错选了A)、B)或D)。

【考点链接】线性表二分查找算法的时间复杂度和空间复杂度的计算。

【答案】C)

【例5】顺序查找适合于存储结构为()的线性表。

A) 散列存储

B) 顺序存储或链式存储

C) 压缩存储

D) 索引存储

【命题目的】考查对顺序查找使用范围的掌握。

【解题要点】顺序查找法适合于线性表,不论采用顺序存储还是链式存储。

【错解分析】散列存储顺序与查找无关,同样压缩存储、索引存储也与顺序查找无关。所以A)选项、C)选项和D)选项都是错误的。

【考点链接】各种查找方式对存储结构的要求。

【答案】B)

【例6】设有一个已按各元素的值排好序的线性表(长度大于2),对给定的值 k ,分别用顺序查找法和二分查找法查找一个与 k 相等的元素,比较的次数分别是 s 和 b ,在查找不成功的情况下, s 和 b 的关系是()。

A) $s = b$

B) $s > b$

C) $s < b$

D) $s \geq b$

【命题目的】考查顺序查找与二分查找在查找不成功时的关键字比较次数的情况。

【解题要点】对于顺序查找,查找不成功时和给定关键字比较的次数为 n 。二分查找不成功的关键字比较次数为 $\lceil \log_2 n \rceil + 1$ 。当 $n \geq 2$ 时,显然 $n + 1 > \lceil \log_2 n \rceil + 1$ 。

【错解分析】由以上分析知A)选项、C)选项和D)选项均是错误的。

【考点链接】各种查找方法的性能分析。

【答案】B)

【例7】采用顺序法查找法查找长度为 n 的线性表时,每个元素的平均查找长度为()。

A) n

B) $n/2$

C) $(n+1)/2$

D) $(n-1)/2$

【命题目的】考查顺序查找算法的平均查找长度的计算。

【解题要点】顺序查找法平均查找长度为 $(n+1)/2$ 。本题答案为C)。

【错解分析】没有对顺序查找法平均查找长度计算的熟练掌握,或混淆了各种查找方法时间复杂度的计算方法,导致错选答案。

【考点链接】算法平均查找长度的计算公式以及各种算法的平均查找长度。

【答案】C)

【例8】采用二分法查找长度为 n 的线性表时,算法的时间复杂度为()。

A) $O(n^2)$

B) $O(n \log_2 n)$

C) $O(n)$

D) $O(\log_2 n)$

【命题目的】考查二分查找算法的时间复杂度的计算。

【解题要点】二分查找法的时间复杂度为 $O(\log_2 n)$ 。本题答案为D)。

【错解分析】本题容易对二分查找法时间复杂度与其他方法的时间复杂度相混淆,从而错误选择了A)选项、B)选项或C)选项。

【考点链接】各种查找算法时间复杂度及其计算。

【答案】D)

【例9】有一个有序表为{1,3,9,12,32,41,45,62,75,77,82,95,100},当用二分法查找值82的结点时,()次比较后查找成功。

A) 1

B) 2

C) 4

D) 8

【命题目的】考查二分查找法的查找过程。

【解题要点】先找中间结点45,再找77、95,最后找到82,经过4次比较。本题答案为C)。

【错解分析】A)选项、B)选项显然是错误的。二分查找时间复杂度为 $O(\log_2 n)$,在此序列中,用二分法查找任何一个都不可能达到8次,所以D)选项也是错误的。

【考点链接】二分查找算法的时间复杂度和平均查找长度。

【答案】C)

【例 10】对有 18 个元素的有序表用二分法查找,则查找 A[3] 的比较序列的下标为()。

A) 1、2、3

B) 9、5、2、3

C) 9、5、3

D) 9、4、2、3

【命题目的】考查二分查找法的算法的查找过程。

【解题要点】第一次 $(1+18)/2=9$, 第二次 $(1+8)/2=4$, 第三次 $(1+3)/2=2$, 第四次 $(3+3)/2=3$ 。本题答案为 D)。

【错解分析】第一次查找时元素的下标应是中间元素的下标,所以 A) 选项错误。第二次查找也应该是剩下的需要查找的部分的中间的元素,所以 B) 选项、C) 选项也是错误的。

【考点链接】二分查找算法的时间复杂度和平均查找长度。

【答案】D)

【例 11】对一个长度为 10 的排好序的表用二分法查找,若查找不成功,至少需要比较的次数是()。

A) 6

B) 5

C) 4

D) 3

【命题目的】考查二分法的查找方法。

【解题要点】若要查找的值小于表中所有元素,查找过程如图 1-33 所示,若要查找的值大于表中的元素与此类似,所以选 D)。

【错解分析】由查找过程分析知 A) 选项、B) 选项和 C) 选项显然是错误的。

【考点链接】二分法的应用。

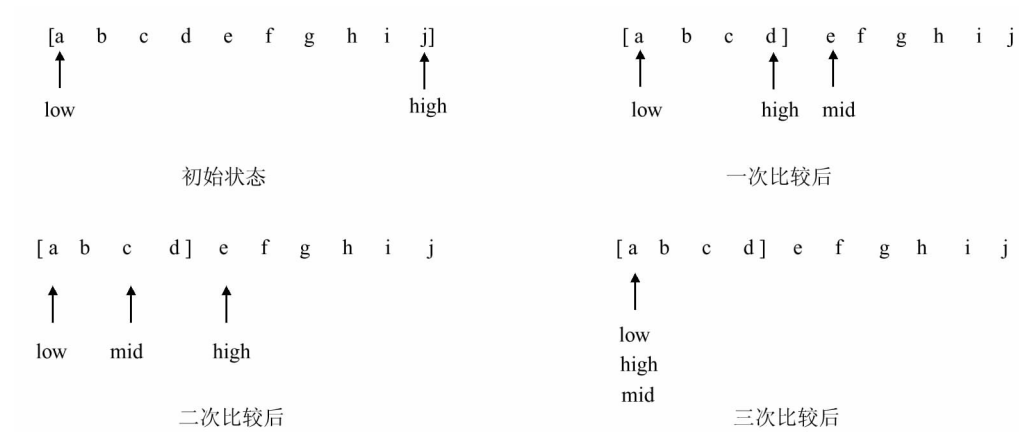


图 1-33 二分法查找过程示意图

【答案】D)

1.8 排序技术

排序(sort)或分类,所谓排序,就是要整理文件中的记录,使之按关键字递增(或递减)的次序排列起来。其确切定义如下:

输入: n 个记录 R_1, R_2, \dots, R_n , 其相应的关键字分别为 K_1, K_2, \dots, K_n 。

输出: $R_{i_1}, R_{i_2}, \dots, R_{i_n}$, 使得 $K_{i_1} \leq K_{i_2} \leq \dots \leq K_{i_n}$ 。(或 $K_{i_1} \geq K_{i_2} \geq \dots \geq K_{i_n}$ 。)

考点 20 交换类排序法

交换排序的基本思想是:两两比较待排序记录的关键字,发现两个记录的次序相反时即进行交换,直到没有反序的记录为止。

应用交换排序基本思想的主要排序方法有:冒泡排序和快速排序。

1. 冒泡排序

冒泡排序(Bubble Sorting)的基本思想是:通过对待排序序列从后向前或从前向后(从下标较大的元素

开始),依次比较相邻元素的排序码,若发现逆序则交换,使排序码较大的元素逐渐从前部移向后部或较小的元素逐渐从后部移向前部(从下标较大的单元移向下标较小的单元),就像水底下的气泡一样逐渐向上冒,如图 1-34 所示。

初始状态:	[15, 3, 23, 14, 21, 8]
从前往后结果:	[(3, 15, 14, 21, 8,) 23]
从后往前结果:	[3, 15, 14, 21, 8,) 23]
从前往后结果:	[3, 8, (14, 15,) 21, 23]
最终结果	[3, 8, (14, 15,) 21, 23]

图 1-34 冒泡法排序过程示意图

冒泡排序法的基本过程是:

- (1)从表头开始往后扫描线性表,在扫描过程中逐次比较相邻两个元素的大小。若相邻两个元素中,前面的元素大于后面的元素,则将它们互换,称为消去了一个逆序。显然在扫描过程中,不断地将两相邻元素中的大者往后移动,最后就将线性表中的最大者换到了表的最后。
- (2)从后到前扫描剩下的线性表,同样,在扫描过程中逐次比较相邻两个元素的大小。若相邻两个元素中,后面的元素小于前面的元素,则将它们互换,这样又消去了一个逆序。显然,在扫描过程中,不断地将两个相邻元素中的小者往前移动,最后就将剩下的线性表中的最小者换到了表的最前面。
- (3)对剩下的线性表重复上述过程,直到剩下的线性表变空为止,此时线性表已经变为有序。

2. 快速排序

快速排序(Quick Sorting)是迄今为止所有内排序算法中速度最快的一种。它的基本思想是:任取待排序序列中的某个元素作为基准(一般取第一个元素),通过一趟排序,将待排元素分为左右两个子序列,左子序列元素的排序码均小于或等于基准元素的排序码,右子序列的排序码则大于基准元素的排序码,然后分别对两个子序列继续进行排序,直至整个序列有序。快速排序是对冒泡排序的一种改进方法,算法中元素的比较和交换是从两端向中间进行的,排序码较大的元素一次就能够交换到后面单元,排序码较小的记录一次就能够交换到前面单元,记录每次移动的距离较远,因而总的比较和移动次数较少。

快速排序的过程为:把待排序区间按照第一个元素(即基准元素)的排序码分为左右两个子序列的过程叫做一次划分。设待排序序列为 $R[\text{left}] \sim R[\text{right}]$,其中 left 为下限, right 为上限, $\text{left} < \text{right}$, $R[\text{left}]$ 为该序列的基准元素,为了实现一次划分,令 i 和 j 的初值分别为 left 和 right 。

在划分过程中,首先让 j 从它的初值开始,依次向前取值,并将每一元素 $R[j]$ 的排序码同 $R[i]$ 的排序码进行比较,直到 $R[j] < R[i]$ 时,交换 $R[j]$ 与 $R[\text{left}]$ 的值,使排序码相对较小的元素交换到左子序列,然后让 i 从 $i + 1$ 开始,依次向后取值,并使每一元素 $R[i]$ 的排序码同 $R[j]$ 的排序码(此时 $R[j]$ 为基准元素)进行比较,直到 $R[i] > R[j]$ 时,交换 $R[i]$ 与 $R[j]$ 的值,使排序码大的元素交换到后面子区间;再接着让 j 从 $j - 1$ 开始,依次向前取值,重复上述过程,直到 i 等于 j ,即指向同一位置为止,此位置就是基准元素最终被存放的位置。如图 1-35 所示。

[46	55	13	42	94	05	17	70]
i ↑						j ↑	
[46	55	13	42	94	05	17	70]
i ↑						j ↑	
[17	55	13	42	94	05	46	70]
i ↑						j ↑	
[17	46	13	42	94	05	55	70]
i ↑					j ↑		
[17	05	13	42	94	46	55	70]
		i ↑			j ↑		
[17	05	13	42	94	46	55	70]
			i ↑		j ↑		
[17	05	13	42	94	46	55	70]
				i ↑	j ↑		
[17	05	13	42]	46	94	55	70]
				i ↑	j ↑		

图 1-35 快速排序的一次划分

考点 21 选择类排序法

选择排序(Selection Sort)的基本思想是:每一趟从待排序的记录中选出关键字最小的记录,顺序放在已排好序的子文件的最后,直到全部记录排序完毕。

常用的选择排序方法有简单选择排序和堆排序。

1. 简单选择排序法

简单选择排序法的思想为:扫描整个线性表,从中选出最小的元素,将它交换到表的最前面;然后对剩下的子表采用同样的方法,直到子表空为止,如图 1-36 所示。

初始状态:	8	3	2	1	7	4	6	5
第 1 次:	1	3	2	8	7	4	6	5
第 2 次:	1	2	3	8	7	4	6	5
第 3 次:	1	2	3	8	7	4	6	5
第 4 次:	1	2	3	4	7	8	6	5
第 5 次:	1	2	3	4	5	8	6	7
第 6 次:	1	2	3	4	5	6	8	7
第 7 次:	1	2	3	4	5	6	7	8

图 1-36 简单选择排序的过程示例

对于长度为 n 的序列,选择排序需要扫描 $n-1$ 遍,每遍扫描均从剩下的子表中选出最小的元素,然后将该最小的元素,与子表中的第一个元素进行交换。最坏情况下简单选择排序法需要比较 $n(n-1)/2$ 次。

2. 堆排序法

若有 n 个元素的序列 $h_1, h_2, h_3, \dots, h_n$, 当满足如下条件:

$$(1) \begin{cases} h_i \leq h_{2i} \\ h_i \leq h_{2i+1} \end{cases} \quad \text{或} \quad (2) \begin{cases} h_i \geq h_{2i} \\ h_i \geq h_{2i+1} \end{cases}$$

其中 $i=1, 2, \dots, \lfloor n/2 \rfloor$, 则称此 n 个元素的序列 $h_1, h_2, h_3, \dots, h_n$ 为一个堆。

若将此排序码按顺序组成一棵完全二叉树, 则(1)称为小根堆(二叉树的所有根结点值小于或等于左右孩子的值), (2)称为大根堆(二叉树的所有根结点值大于或等于左右孩子的值)。本节只讨论后者。

例如, 序列(91, 85, 53, 36, 47, 30, 24, 12)是一个堆, 它所对应的完全二叉树如图 1-37 所示。

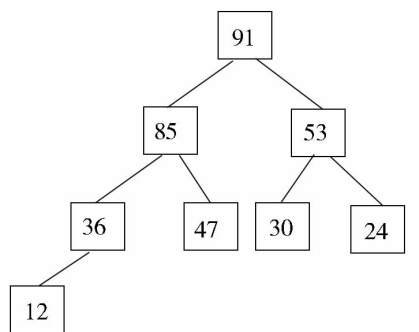


图 1-37 大根堆

在调整建堆的过程中,总是将根结点值与左、右子树的根结点值进行比较,若不满足堆的条件,则将左、右子树根结点值中的大者与根结点值进行交换。这个调整过程一直做到所有子树均为堆为止。

堆排序的方法为:

(1) 首先将一个无序序列建成堆。

(2) 然后将堆顶元素(序列中的最大项)与堆中最后一个元素交换(最大项应该在序列的最后)。不考虑已经换到最后的那个元素,只考虑前 $n-1$ 个元素构成的子序列,将该子序列调整为堆。反复做步骤(2),直到剩下的子序列空为止。

在最坏情况下,堆排序法需要比较的次数为 $O(n\log_2 n)$ 。

考点 22 插入类排序法

插入排序(Insertion Sort)的基本思想是:每次将一个待排序的记录,按其关键字大小插入到前面已经排好序的子文件中的适当位置,直到全部记录插入完成为止。

本节介绍两种插入排序方法:简单插入排序和希尔排序。

1. 简单插入排序法

所谓插入排序,是指将无序序列中的各元素依次插入到已经有序的线性表中。

简单插入排序法的基本思想:把 n 个待排序的元素看做一个有序表和一个无序表,开始时有序表中只包含一个元素,无序表中包含有 $n-1$ 个元素,排序过程中每次从无序表中取出第一个元素,把它的排序码依次与有序表元素的排序码进行比较,将它插入到有序表中的适当位置,使之成为新的有序表,如图 1-38 所示。

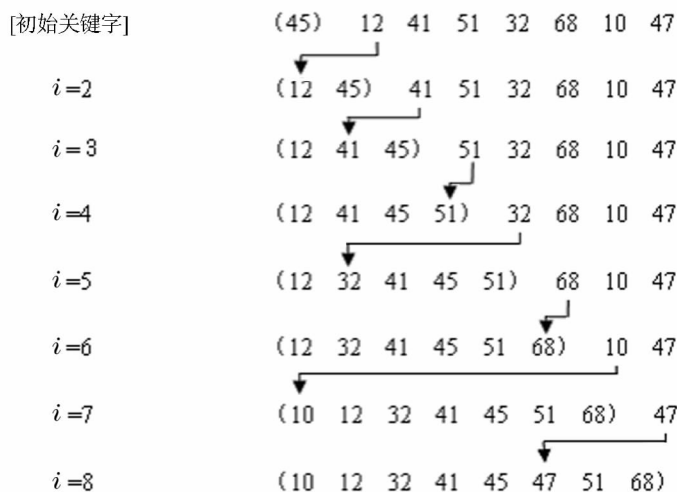


图 1-38 简单插入排序示例

在简单插入排序法中,每一次比较后最多移掉一个逆序,因此,这种排序方法的效率与冒泡排序法相

同。在最坏的情况下,简单插入排序需要 $n(n-1)/2$ 次比较。

2. 希尔排序法

希尔排序法(Shell Sort)的基本思想是:先将整个待排元素序列分割成若干个子序列(由相隔某个“增量”的元素组成的)分别进行直接插入排序,待整个序列中的元素基本有序(增量足够小)时,再对全体元素进行一次直接插入排序。因为直接插入排序在元素基本有序的情况下(接近最好情况),效率是很高的。

增量序列一般取 $h_i = n/2^k (k = 1, 2, \dots, \lceil \log_2 n \rceil)$, 其中 n 为待排序序列的长度,如图 1-39 所示。

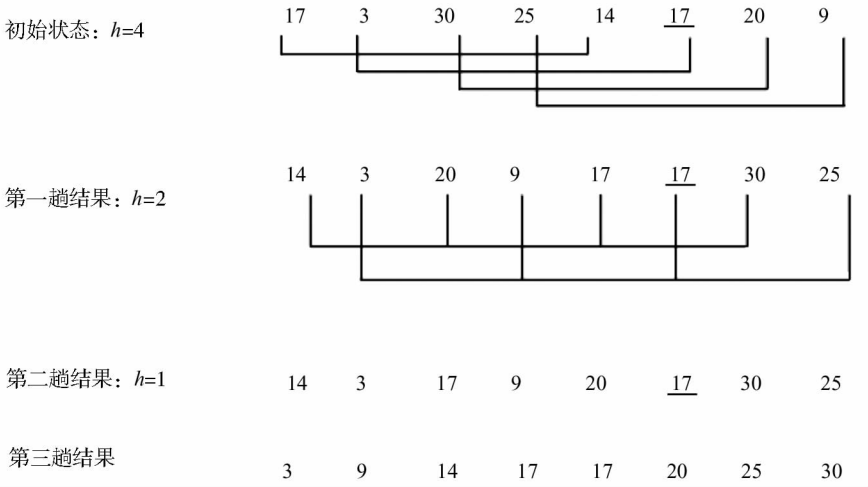


图 1-39 希尔排序算法的执行过程

希尔排序的效率与所选取的增量序列有关。在最坏情况下,希尔排序的时间复杂度为 $O(n\log_2 n)$ 。

附:各种排序方法时间复杂度、空间复杂度对比见表 1-1。

表 1-1 各种排序方法时间、空间复杂度对比

排序方法	时间复杂度			空间复杂度	复杂性
	平均情况	最坏情况	最好情况		
直接插入排序	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	简单
冒泡排序	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	简单
希尔排序	$O(n\log_2 n)$	$O(n\log_2 n)$		$O(1)$	较复杂
快速排序	$O(n\log_2 n)$	$O(n^2)$	$O(n\log_2 n)$	$O(n\log_2 n)$	较复杂
直接选择排序	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	简单
堆排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(1)$	较复杂

真题题解

【例 1】设关键码序列(16,9,4,25,15,2,13,18,17,5,8,24),要按关键码值递增的次序排列,采用直接选择排序法,一趟扫描后的结果是()。

- A) (15,2,4,18,16,5,8,24,17,9,13,25)
- B) (2,9,4,25,15,16,13,18,17,5,8,24)
- C) (9,4,16,15,2,13,18,17,5,8,24,25)
- D) (9,16,4,25,21,5,13,18,5,17,8,24)

【命题目的】考查直接选择排序的基本算法。

【解题要点】直接选择排序法的思想是:以无序表的第一个元素作为比较标准,依次同后面的元素进行比较,如果有一个元素比第一个元素小则记录这个元素的下标,然后以新的最小元素继续往下比较,有更小的元素再记录该下标,再比较……当对整个数组扫描一趟后就可以得到最小元素的下标,然后与无序表的第一个元素交换位置。本题很明显第一趟扫描结果

最小元素是2,与第一个元素交换位置后得到B结果。所以答案为B)。

【错解分析】混淆了直接选择排序和其他排序算法,容易误选答案。

【考点链接】冒泡排序算法、快速排序算法和堆排序算法。

【答案】B)

【例2】设有1000个无序的元素,希望用最快的速度挑选出其中前10个最大的元素,最好选用()排序法。

- A) 冒泡排序 B) 快速排序 C) 堆排序 D) 选择排序

【命题目的】堆排序方法的应用。

【解题要点】由于堆排序每扫描一趟就排好一个记录,只挑选出其中的前10个最大的元素时,使用堆排序为好。

【错解分析】每扫描一趟就排好一个记录是堆排序所特有的,A)选项、B)选项和D)选项都不是正确的答案。

【考点链接】冒泡排序、快速排序、堆排序、选择排序的比较,在哪些场合使用哪种排序。

【答案】C)

【例3】对任意7个关键字进行排序,至少要进行()次关键字之间的两两比较。

- A) 13 B) 14 C) 15 D) 16

【命题目的】本题考查对各种借助“比较”进行排序的算法中最好情况下的时间复杂度。

【解题要点】任何一个借助“比较”进行排序的算法,在最好情况下所需要进行的比较次数至少为 $\lceil \log_2(n!) \rceil$,对于本题 $\lceil \log_2(7!) \rceil = 15$ 。

【错解分析】“至少”则应该是最好情况下的比较次数,因此由公式计算知A)选项、B)选项和D)选项是错误的。

【考点链接】几个排序方法的时间复杂度的比较。

【答案】C)

【例4】排序方法中,将整个无序序列分割成若干小的子序列并分别进行插入排序的方法,称为()。

- A) 希尔排序 B) 冒泡排序 C) 插入排序 D) 选择排序

【命题目的】考查希尔排序算法的基本思想。

【解题要点】希尔排序的基本思想是:先将整个待排元素序列分割成若干个子序列(由相隔某个“增量”的元素组成的)分别进行直接插入排序,待整个序列中的元素基本有序(增量足够小)时,再对全体元素进行一次直接插入排序。

【错解分析】本题考查的是基本概念,由希尔排序的概念知B)选项、C)选项、D)选项都是错误的。

【考点链接】各种排序方法的基本思想和算法。

【答案】A)

【例5】在文件“局部有序”或文件长度较小的情况下,最佳内排序方法是()。

- A) 直接插入排序 B) 冒泡排序
C) 直接选择排序 D) 希尔排序

【命题目的】考查对直接插入排序最好情况下的时间复杂度的掌握。

【解题要点】直接插入排序的过程是,每一趟将一个待排序的记录,按其关键字的大小插入到已经排序的子文件中适当位置上,直到全部记录插入完成为止。当文件“局部有序”或文件长度较小的情况下,每趟的比较次数大为降低,也即 $n-1$ 趟比较的时间复杂度由 $O(n^2)$ 降至 $O(n)$ 。所以本题答案为A)。

【错解分析】这是直接插入排序所具有的特征,B)选项、C)选项和D)选项都是错误的。

【考点链接】直接插入排序在最好情况下、平均情况下和最坏情况下的时间复杂度。

【答案】A)

【例6】排序方法中,从未排序序列中依次取出元素与已排序序列(初始时空)中的元素作比较,将其放入已排序序列的正确位置上的方法,称为()。

- A) 希尔排序 B) 选择排序
C) 冒泡排序 D) 插入排序

【命题目的】考查插入排序的概念。

【解题要点】插入排序的基本思想是:把 n 个待排序的元素看做一个有序表和一个无序表,开始时有序表中只包含一个元素,无序表中包含有 $n-1$ 个元素,排序过程中每次从无序表中取出第一个元素,把它的排序码依次与有序表元素的排序码进行比较,将它插入到有序表中的适当位置,使之成为新的有序表。

【错解分析】希尔排序的基本思想是:先将整个待排元素序列分割成若干个子序列(由相隔某个“增量”的元素组成的)分别进行直接插入排序,待整个序列中的元素基本有序(增量足够小)时,再对全体元素进行一次直接插入排序,所以A)选项错

误;选择排序的思想为:扫描整个线性表,从中选出最小的元素,将它交换到表的最前面;然后对剩下的子表采用同样的方法,直到子表空为止,所以 B) 选项错误;冒泡排序的基本思想是:通过对待排序序列从后向前(从下标较大的元素开始),依次比较相邻元素的排序码,若发现逆序则交换,使排序码较小的元素逐渐从后部移向前部,所以 C) 选项也不正确。

【考点链接】各种排序方法概念和基本思想。

【答案】D)

【例 7】用直接插入排序方法对下列 4 个表由小到大进行排序,比较次数最少的是()。

A) (94,32,40,90,80,46,21,69)

B) (21,32,46,40,80,69,90,94)

C) (32,40,21,46,69,94,90,80)

D) (90,69,80,46,21,32,94,40)

【命题目的】考查影响直接插入排序时间复杂度的因素。

【解题要点】在直接插入排序过程中,当待排序列中记录按关键字非递减有序排序时,所需进行关键字比较的次数最小,为 $n-1$,即记录不需移动;反之,当待排序列中记录按关键字非递增有序排序时,总的比较次数达到最大值 $(n+2)(n-1)/2$ 。由 A) 选项、B) 选项、C) 选项和 D) 选项 4 个选项中知 B) 选项已经基本有序,需要比较的次数最少。所以答案选 B)。

【错解分析】A) 选项、C) 选项和 D) 选项中记录混乱无序,而 B) 选项中只需将 46 和 40,以及 80 和 69 的位置交换即可完成排序,关键字比较分别只需比较两次,其他关键字比较只需比较一次即可。显然 A) 选项、C) 选项和 D) 选项都是错误的。

【考点链接】直接排序时间复杂度计算。

【答案】B)

【例 8】用二分法插入排序方法进行排序,被排序的表(或序列)应采用的数据结构是()。

A) 单链表

B) 数组

C) 双向链表

D) 散列表

【命题目的】考查二分法插入排序对被排序表的存储结构要求。

【解题要点】二分法插入排序有条件限制:要求表必须用顺序存储结构,且表中元素必须按关键字有序(升序或降序均可)。数组在内存中是顺序存储的。

【错解分析】单链表的存储结构不是顺序存储的,所以 A) 选项不符合题目要求;同样道理 C) 选项和 D) 选项也是错误答案。

【考点链接】二分查找方法的基本思想。

【答案】B)

【例 9】在待排序的元素序列基本有序的前提下,效率最高的排序方法是()。

A) 插入排序

B) 选择排序

C) 快速排序

D) 冒泡排序

【命题目的】考查对插入排序最好情况下的时间复杂度的掌握情况。

【解题要点】插入排序是将待排序的记录插入到前面已排好序的子文件中,即考虑已排好序的子文件。所以本题的答案为 A)。

【错解分析】由各排序算法知只有 A) 选项是正确的,其他都是错误的。

【考点链接】插入排序、选择排序、快速排序、冒泡排序的效率分析。

【答案】A)

【例 10】用直接插入排序方法对序列{15,11,9,10,13}进行排序,关键码比较次数是()。

A) 10

B) 8

C) 4

D) 9

【命题目的】考查直接插入排序方法的应用。

【解题要点】直接插入排序过程如图 1-40 所示。

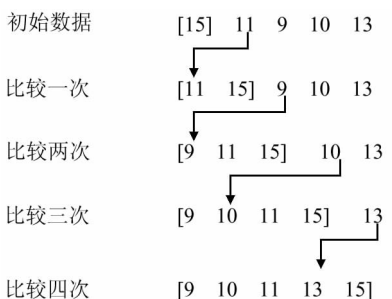


图 1-40 直接插入排序过程示意图

【错解分析】由以上分析知 B) 选项、C) 选项和 D) 选项均是错误的。

【考点链接】各种排序方法的具体应用。

【答案】A)

【例 11】在快速排序过程中,每次划分,将被划分的表(或子表)分成左、右两个子表,考虑这两个子表,下列结论一定正确的是()。

- A) 左、右两个子表都已各自排好序
- B) 左边子表中的元素都不大于右边子表中的元素
- C) 左边子表的长度小于右边子表的长度
- D) 左、右两个子表中元素的平均值相等

【命题目的】考查快速排序方法的排序过程。

【解题要点】快速排序基本思想是:任取待排序表中的某个元素作为基准(一般取第一个元素),通过一趟排序,将待排元素分为左右两个子表,左子表元素的排序码均小于或等于基准元素的排序码,右子表的排序码则大于基准元素的排序码,然后分别对两个子表继续进行排序,直至整个表有序。

【错解分析】对于 A) 选项,只有左子表排好序,右子表还没排好序;对于 C) 选项,左子表的长度在排序过程中可能大于、等于或小于右子表的长度;对于 D) 选项,没有这种说法。

【考点链接】各种排序方法的排序过程。

【答案】B)

【例 12】在下列排序算法中,()算法可能出现下列情况:在最后一趟开始之前,所有的元素都不在其最终位置上。

- A) 堆排序
- B) 冒泡排序
- C) 插入排序
- D) 快速排序

【命题目的】考查插入排序算法的排序过程。

【解题要点】在插入排序中,如果待排序列中的最后一个元素的关键字最小,则在最后一趟开始之前,前 $n-1$ 个排好序的元素都不在其最终位置上(都相差一个位置)。本题答案为 C)。

【错解分析】插入排序的一次排序可以改变所有元素的位置,即当最后一个元素插入到第一个位置时所有元素的位置将改变。只有 C) 选项正确。

【考点链接】堆排序、冒泡排序和快速排序的特点和排序过程。

【答案】C)

【例 13】对记录的关键字为{50,26,38,80,70,90,8,30,40,20}进行排序,各趟排序结束时的结果为:

50,26,38,80,70,90,8,30,40,20
 50,8,30,40,20,90,26,38,80,70
 26,8,30,40,20,80,50,38,90,70
 8,20,26,30,38,40,50,70,80,90

其使用的排序方法是()。

- A) 快速排序
- B) 冒泡排序
- C) 希尔排序
- D) 插入排序

【命题目的】考查希尔排序算法的特点和排序过程。

【解题要点】由排序结果可知,该排序是增量序列为 5、3、1 的希尔排序。本题答案为 C)。

【错解分析】第一趟中 26 和 8 交换了位置,38 和 30 交换了位置,80 和 40 交换了位置,90 和 20 交换了位置,它们的增量分别为 5。同理,第二趟的增量为 3,由后往前排。第三趟的增量为 1,由前往后排。这是希尔排序特有的排序方法,所以 A) 选项、B) 选项和 D) 选项均错误。

【考点链接】快速排序、冒泡排序和插入排序算法特点和过程。

【答案】C)

【例 14】对 n 个元素进行冒泡排序的过程中,最好情况下的时间复杂度为()。

- A) $O(1)$
- B) $O(\log_2 n)$
- C) $O(n^2)$
- D) $O(n)$

【命题目的】考查有关冒泡排序算法的时间复杂度。

【解题要点】冒泡排序平均情况时间复杂度为 $O(n^2)$;最坏情况时间复杂度为 $O(n^2)$;最好情况时间复杂度为 $O(n)$ 。本题答案为 D)。

【错解分析】容易把冒泡排序最好的情况下与最坏情况或与平均情况或与其他排序的时间复杂度相混淆。

【考点链接】由最好情况下冒泡排序的时间复杂度知 A) 选项、B) 选项和 C) 选项均是错误的。

【答案】D)

【例 15】一组记录的关键码为(46,79,56,38,40,84),则利用快速排序的方法,以第一个记录为基准得到的一次划分结果为()。

A) 38,40,46,56,79,84

B) 40,38,46,79,56,84

C) 40,38,46,56,79,84

D) 40,38,46,84,56,79

【命题目的】考查快速排序算法的特点和排序过程。

【解题要点】第一次划分的过程如下:

46,79,56,38,40,84 //40 小于 46。两者交换得到 40,79,56,38,46,84, $i=1, j=5$

40,79,56,38,46,84 //79 大于 46。两者交换得到 40,46,56,38,79,84, $i=2, j=5$

40,46,56,38,79,84 //38 小于 46。两者交换得到 40,38,56,46,79,84, $i=2, j=4$

40,38,56,46,79,84 //56 大于 46。两者交换得到 40,38,46,56,79,84, $i=3, j=4$

$i=j=4$,退出得到第一次划分后的序列。本题答案为 C)。

【错解分析】由快速排序的分析过程知,只有 C) 选项是正确的。

【考点链接】选择排序、冒泡排序和插入排序的算法特点和排序过程。

【答案】C)

【例 16】快速排序方法在()情况下最不利于发挥其长处。

A) 要排序的数据量太大

B) 要排序的数据中含有多个相同值

C) 要排序的数据已基本有序

D) 要排序的数据个数为奇数

【命题目的】考查快速排序算法的特点和适用范围。

【解题要点】要排序的数据(个数为 n)已基本有序,采用快速排序则需要 $n-1$ 趟,其时间复杂度升至 $O(n^2)$ 。本题答案为 C)。

【错解分析】A) 选项、B) 选项和 D) 选项跟排序的优越性无影响。

【考点链接】其他各种排序算法的特点和适用范围。

【答案】C)

【例 17】快速排序在最坏情况下时间复杂度是 $O(n^2)$,比()的性能差。

A) 堆排序

B) 冒泡排序

C) 选择排序

D) 以上都不对

【命题目的】考查各种排序算法的时间复杂度的比较。

【解题要点】堆排序最坏情况时间复杂度为 $O(n\log_2 n)$,比快速排序最坏情况下好,本题答案为 A)。

【错解分析】没有正确掌握各种排序方法在不同条件下性能的比较从而错误地选择其他答案。冒泡排序和选择排序最坏情况下的时间复杂度都是 $O(n^2)$,所以 B) 选项和 C) 选项不能选。而堆排序最坏情况下的时间复杂度是 $O(n\log_2 n)$,所以 D) 选项也是错误的。

【考点链接】各种排序算法的时间复杂度和空间复杂度。

【答案】A)

【例 18】采用直接选择排序,比较次数与移动次数分别是()。

A) $O(n), O(n)$

B) $O(\log_2 n), O(n^2)$

C) $O(n^2), O(n)$

D) $O(\log_2 n), O(n)$

【命题目的】考查直接选择排序算法的排序过程和时间复杂度。

【解题要点】直接选择排序过程是,每趟从 $n-i+1$ 个记录中选取关键字最小的记录(每趟比较的时间复杂度为 $O(n)$,并与第 i 个记录交换(每趟移动的时间复杂度为 $O(1)$),因此,总的比较次数与移动次数分别是 $O(n^2)$ 和 $O(n)$ 。本题答案为 C)。

【错解分析】常见错误是,把直接选择排序的一趟排序的时间复杂度与总的时间复杂度混淆认为都是 $O(n)$,实际上, A) 选项、B) 选项和 D) 选项都是错误的。

【考点链接】快速排序、冒泡排序算法的排序过程和时间复杂度。

【答案】C)

【例 19】如果对 n 个元素进行直接选择排序,则进行任一趟排序的过程中,为寻找最小值元素所需要的时间复杂度为()。

A) $O(1)$

B) $O(\log_2 n)$

C) $O(n^2)$

D) $O(n)$

【命题目的】考查直接选择排序一趟排序过程中的时间复杂度。

【解题要点】直接选择排序每趟从 $n-i+1$ 个记录中选取关键字最小的记录,其时间复杂度为 $O(n)$ 。本题的答案为 D)。

【错解分析】本题考查直接选择排序的时间复杂度,只有 D) 选项是正确的。

【考点链接】直接选择排序时间复杂度计算。

【答案】D)

【例 20】在所有排序方法中,关键字比较的次数与记录的初始排列次序无关的是()。

- A) 希尔排序 B) 快速排序 C) 插入排序 D) 选择排序

【命题目的】考查学生对选择排序的掌握。

【解题要点】由于选择排序每趟从待排序的记录中选中关键字最小的记录,每个记录都要比较,不考虑已排好序的子文件,因此,关键字比较的次数与记录的初始排列次序无关。

【错解分析】快速排序不考虑已排好序的子记录,所以 D) 选项才是答案。

【考点链接】各种排序方法的具体应用。

【答案】D)

【例 21】在对 n 个元素的序列进行排序时,堆排序所需要的附加存储空间是()。

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

【命题目的】考查对堆排序的空间复杂度的掌握。

【解题要点】堆排序的空间复杂度为 $O(1)$;时间复杂度在最好情况下为 $O(n\log_2 n)$,平均情况为 $O(n\log_2 n)$,最坏情况为 $O(n\log_2 n)$ 。本题答案为 B)。

【错解分析】本题容易把堆排序的空间复杂度与堆排序的最好情况、最坏情况或平均情况下的时间复杂度相混淆,从而错误地选择了 A) 选项、C) 选项或 D) 选项。

【考点链接】冒泡排序、选择排序、插入排序等排序方法的时间、空间复杂度。

【答案】B)

1.9 同步练习

1. 算法指的是()。

- A) 计算机程序 B) 解决问题的计算方法
C) 排序算法 D) 解决问题的有限运算序列

2. 算法能正确地实现预定功能的特性称为算法的()。

- A) 正确性 B) 易读性 C) 健壮性 D) 高效率

3. 数据的不可分割的基本单位是()。

- A) 元素 B) 结点 C) 数据类型 D) 数据项

4. 一个递归的定义可以用递归过程求解,也可以用非递归过程求解,但单从运行时间来看,通常递归过程比非递归过程()。

- A) 较快 B) 较慢 C) 相同 D) 无法确定

5. 以下有关数据结构的叙述,正确的是()。

- A) 线性表的线性存储结构优于链式存储结构
B) 二叉树的第 i 层上有 2^{i-1} 个结点,深度为 k 的二叉树上有 $2^k - 1$ 个结点
C) 二维数组是其数据元素为线性表的线性表
D) 栈的操作方式是先进先出

6. 在数据结构的讨论中把数据结构从逻辑上分为()。

- A) 内部结构与外部结构 B) 静态结构与动态结构
C) 线性结构与非线性结构 D) 紧凑结构与非紧凑结构

7. 数据的逻辑关系是指数据元素的()。

- A) 关联 B) 结构
C) 数据项 D) 存储方式

8. 下列关于数据结构的叙述中,正确的是()。

- A) 数组是同类型值的集合
- B) 递归算法的程序结构比迭代算法的程序结构更为精炼
- C) 树是一种线性结构
- D) 用一维数组存储二叉树,总是以先序遍历的顺序存储各结点

9. 执行下面程序段时,执行 S 语句的次数为()。

```
for(int i = 1; i <= n; i++)
    for(int j = 1; j <= i; j++)
        S;
```

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

10. 下面程序段的时间复杂度是()。

```
for(int i = 0; i < n; i++)
    for(int j = 1; j < m; j++)
        A[i][j] = 0;
```

- A) $O(n)$
- B) $O(m+n+1)$
- C) $O(m+n)$
- D) $O(m \times n)$

11. 下列算法 suanfa 的时间复杂度为()。

```
int suanfa(int n)
{
    int t = 1;
    while(t <= n)
        t = t * 2;
    return t;
}
```

- A) $O(\log_2 n)$
- B) $O(2^n)$
- C) $O(n^2)$
- D) $O(n)$

12. 单链表的每个结点中包括一个指针 link,它指向该结点的后继结点。现要将指针 q 指向的新结点插入到指针 p 指向的单链表结点之后,下面的操作序列中()是正确的。

- A) $q \rightarrow \text{link} = p \rightarrow \text{link}; p \rightarrow \text{link} = q \rightarrow \text{link}$
- B) $p \rightarrow \text{link} = q \rightarrow \text{link}; q \rightarrow \text{link} = p \rightarrow \text{link}$
- C) $q \rightarrow \text{link} = p \rightarrow \text{link}; p \rightarrow \text{link} = q$
- D) $p \rightarrow \text{link} = q; q \rightarrow \text{link} = p \rightarrow \text{link}$

13. 以下数据结构中,()是线性结构。

- A) 有向图
- B) 栈
- C) 线索二叉树
- D) B 树

14. 下列关于链式存储结构的叙述中,()是不正确的。

- A) 结点除自身信息外还包括指针域,因此存储密度小于顺序存储结构
- B) 逻辑上相邻的结点物理上不必邻接
- C) 可以通过计算直接确定第 i 个结点的存储地址
- D) 插入、删除运算操作方便,不必移动结点

15. 下列关于顺序存储结构的叙述中,()是不正确的。

- A) 存储密度大
- B) 逻辑上相邻的结点物理上不必邻接
- C) 可以通过计算机直接确定第 i 个结点的存储地址
- D) 插入、删除运算操作不方便

16. 下列关于数据的逻辑结构的叙述中,()是正确的。

- A) 数据的逻辑结构是数据间关系的描述
- B) 数据的逻辑结构反映了数据在计算机中的存储方式
- C) 数据的逻辑结构分为顺序结构和链式结构
- D) 数据的逻辑结构分为静态结构和动态结构

17. 线性表是一个()。

- A) 有限序列,可以为空
- B) 有限序列,不能为空

- C) 无限序列,可以为空 D) 无限序列,不能为空
18. 某线性表采用顺序存储结构,每个元素占4个存储单元,首地址为100,则第12个元素的存储地址为()。
- A) 144 B) 145 C) 147 D) 148
19. 若长度为 n 的线性表采用顺序存储结构,那么删除它的第 i 个数据元素之前,需要它依次向前移动()个数据元素。
- A) $n-i$ B) $n+i$ C) $n-i-1$ D) $n-i+1$
20. 若长度为 n 的线性表采用顺序存储结构,在第 i 个位置插入一个元素,需要它依次向后移动()数据元素。
- A) $n-i$ B) $n-i+1$ C) $n-i-1$ D) i
21. 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()个元素。
- A) 8 B) 63.5 C) 63 D) 7
22. 线性表中,只有直接前驱而无后继的元素是()。
- A) 尾元素 B) 首元素 C) 所有中间的元素 D) 全部元素
23. 线性表 $L=(a_1, a_2, \dots, a_n)$ 中, a_{i+1} 称为 a_i 的()。
- A) 直接后继 B) 直接前驱 C) 孩子 D) 兄弟
24. 若事先不知道线性表的长度,则处理线性表时较好的存储结构是()。
- A) 单链表 B) 静态链表 C) 顺序表 D) B)和C)
25. 设线性表 $L=(a_1, a_2, \dots, a_n)$,下列关于线性表的叙述正确的是()。
- A) 每个元素都有一个直接前驱和一个直接后继
B) 线性表中至少要有有一个元素
C) 表中元素排列顺序必须按由小到大或由大到小
D) 除第一个和最后一个元素外,其余每个元素都有且只有一个直接前驱和一个直接后继
26. 下列所述不是树型结构特点的是()。
- A) 每一个结点可以有多个后件,它们都称为该结点的子结点
B) 每个结点有多个前件
C) 一个结点所拥有的后件个数称为该结点的度
D) 树的最大层次称为树的深度
27. 设线性表的顺序存储结构中,每个元素占用1个存储单元,表的第1个元素的存储地址为 d ,则第 i 个元素($1 \leq i \leq n$, n 为表长)的存储地址为()。
- A) $d+(i-1) \times 1$ B) $d+i \times 1$ C) $d+(i+1) \times 1$ D) $d+i \times 1-1$
28. 链表表示线性表的优点是()。
- A) 便于随机存取 B) 花费的存储空间比顺序表少
C) 便于插入与删除 D) 数据元素的物理顺序与逻辑顺序相同
29. 线性表采用链式存储时,结点的存储地址()。
- A) 必须是不连续的 B) 连续与否均可
C) 必须是连续的 D) 和头结点的存储地址相连续
30. 一维数组与线性表的区别是()。
- A) 前者长度固定,后者长度可变 B) 后者长度固定,前者长度可变
C) 两者长度均固定 D) 两者长度均可变
31. 线性表的链接实现有利于()。
- A) 插入运算 B) 读表元运算 C) 查找运算 D) 定位运算
32. 线性链表不具有的特点是()。
- A) 随机访问 B) 不必事先估计所需存储空间大小
C) 插入与删除时不必移动元素 D) 所需空间与线性表长度成正比
33. 若进栈序列为3,5,7,9,进栈过程中可以出栈,则()不可能是一个出栈序列。
- A) 7,5,3,9 B) 9,7,5,3 C) 7,5,9,3 D) 9,5,7,3
34. 数组 $Q[0 \dots n-1]$ 用来表示一个环形队列, f 为当前队头元素的前一位置, r 为队尾元素的位置,假定队列中元素的个数总小于 n ,计算队列中元素个数的公式为()。
- A) $r-f$ B) $n+f-r$ C) $n+r-f$ D) $(n+r-f) \bmod n$

35. 4 个元素 a_1 、 a_2 、 a_3 和 a_4 依次入栈,入栈过程中允许栈顶元素出栈,假设某时刻栈的状态是 a_3 (栈顶)、 a_2 、 a_1 (栈底),则不可能的出栈序列是()。
- A) a_4, a_3, a_2, a_1 B) a_3, a_2, a_4, a_1 C) a_3, a_1, a_4, a_2 D) a_3, a_4, a_2, a_1
36. 与数据元素本身的形式、内容、相对位置、个数无关的是数据的()。
- A) 存储结构 B) 存储实现 C) 逻辑结构 D) 运算实现
37. 设用一数组 $A[1 \dots n]$ 来存储一个栈,令 $A[n]$ 为栈底,用整型变量 T 指示当前栈顶位置, $A[T]$ 为栈顶元素。当从栈中弹出一个元素时,变量 T 的变化为()。
- A) $T = T + 1$ B) $T = T - 1$ C) T 不变 D) $T = n$
38. 有 6 个元素按 1、2、3、4、5、6 的顺序进栈,下列()不是合法的出栈序列。
- A) 2、3、4、1、6、5 B) 3、2、4、6、5、1 C) 4、3、1、2、5、6 D) 5、4、6、3、2、1
39. 一个栈的入栈序列是 a、b、c、d、e,则栈的不可能的输出序列是()。
- A) e、d、c、b、a B) d、e、c、b、a C) d、c、e、a、b D) a、b、c、d、e
40. 设计一个判别表达式左、右括号是否配对出现的算法,采用()数据结构最佳。
- A) 线性表的顺序存储结构 B) 栈
C) 队列 D) 线性表的链式存储结构
41. 下面 4 种内排序方法中,要求内存容量最大的是()。
- A) 插入排序 B) 选择排序 C) 快速排序 D) 归并排序
42. 栈的插入和删除操作在()进行。
- A) 栈顶 B) 栈底 C) 任意位置 D) 指定位置
43. 向顺序栈中压入新元素时,应当()。
- A) 先移动栈顶指针,再存入元素 B) 先存入元素,再移动栈顶指针
C) 先后次序无关紧要 D) 同时进行
44. 链式栈与顺序栈相比,一个比较明显的优点是()。
- A) 插入操作更加方便 B) 通常不会出现栈满的情况
C) 不会出现栈空的情况 D) 删除操作更加方便
45. 设一数列的顺序为 1、2、3、4、5、6,通过栈操作可以得到()的输出序列。
- A) 3、2、5、6、4、1 B) 1、2、3、4、5、6 C) 6、5、4、3、2、1 D) 4、5、3、2、6、1
46. 设一个栈的输入序列为 A、B、C、D,则借助一个栈所得到输出序列不可能是()。
- A) A、B、C、D B) D、C、B、A C) A、C、D、B D) D、A、B、C
47. 若用一个大小为 6 的数组来实现循环队列,且当前 rear 和 front 的值分别为 0 和 3。从当前队列中删除一个元素,再加入两个元素后,rear 和 front 的值分别为()。
- A) 1 和 5 B) 2 和 4 C) 4 和 2 D) 5 和 1
48. 由两个栈共享一个向量空间的好处是()。
- A) 减少存取时间,降低下溢发生的概率 B) 节省存储空间,降低下溢发生的概率
C) 减少存取时间,降低上溢发生的概率 D) 节省存储空间,降低下溢发生的概率
49. 设一数列的顺序为 1、2、3、4、5、6,通过队列操作可以得到()的输出序列。
- A) 3、2、5、6、4、1 B) 1、2、3、4、5、6 C) 6、5、4、3、2、1 D) 4、5、3、2、6、1
50. 从一个顺序队列中删除元素时,首先要()。
- A) 前移一位队首指针 B) 后移一位队首指针
C) 取出队首指针所指位置上的元素 D) 取出队尾指针所指位置上的元素
51. 在一个顺序存储的循环队列中,队头指针指向队头元素的()。
- A) 前一个位置 B) 后一个位置
C) 队头元素位置 D) 队尾元素的前一位置
52. 两栈共享数组存储空间,前一个栈的栈顶指针为 p ,后一个栈的栈顶指针为 q ,能进行正常入栈操作的条件是()。
- A) $p \leq q$ B) $p < q$ C) $p < q - 1$ D) $p = q - 2$
53. 将长度为 n 的单链表链接在长度为 m 的单链表之后的算法的时间复杂度为()。
- A) $O(1)$ B) $O(n)$ C) $O(m)$ D) $O(m + n)$

54. 在一个单链表中,若 q 所指结点是 p 所指结点的前驱,若在 q 与 p 之间插入一个 s 所指的结点,则执行()。
- A) $s \rightarrow \text{link} = p \rightarrow \text{link}; p \rightarrow \text{link} = s;$ B) $p \rightarrow \text{link} = s; s \rightarrow \text{link} = q;$
 C) $p \rightarrow \text{link} = s \rightarrow \text{link}; s \rightarrow \text{link} = p;$ D) $q \rightarrow \text{link} = s; s \rightarrow \text{link} = p;$
55. 下列关于数据结构的叙述中,正确的是()。
- A) 数组是同类型值的集合
 B) 递归算法的程序结构比迭代算法的程序结构更为精练
 C) 树是一种线性结构
 D) 用一维数组存储二叉树,总是以先序遍历的顺序存储各个结点
56. 在一个顺序循环队列中,队尾指针指向队尾元素的()位置。
- A) 后两个 B) 后一个 C) 当前 D) 前一个
57. 在单链表中,指针 p 指向元素为 x 的结点,实现“删除 x 的后继”的语句是()。
- A) $p = p \rightarrow \text{next};$ B) $p \rightarrow \text{next} = p \rightarrow \text{next} \rightarrow \text{next};$
 C) $p \rightarrow \text{next} = p;$ D) $p = p \rightarrow \text{next} \rightarrow \text{next};$
58. 在表头指针为 head 且表长大于 1 的单向循环链表中,指针 p 指向表中的某个结点,若 $p \rightarrow \text{next} \rightarrow \text{next} = \text{head}$,则()。
- A) p 指向头结点 B) p 指向尾结点
 C) $*p$ 的直接后继是头结点 D) $*p$ 的直接后继是尾结点
59. 单链表中,增加一个头结点的目的是()。
- A) 使单链表至少有一个结点 B) 标识表结点中首结点的位置
 C) 方便运算的实现 D) 说明单链表是线性表的链式存储
60. 循环链表的主要优点是()。
- A) 不在需要头指针了
 B) 已知某个结点的位置后,能够容易找到它的直接前驱
 C) 在进行插入、删除运算时,能更好地保证链表不断开
 D) 从表中的任意结点出发都能扫描到整个链表
61. 若 $(a-b) * (c+d)$ 是中序表达式,则其后序表达式是()。
- A) $abcd + * -$ B) $ab - cd + *$ C) $ab - * cd +$ D) $a - bcd + *$
62. 设森林 F 对应的二叉树为 B ,它有 m 个结点, B 的根为 P , P 的右子树上的结点个数为 n ,森林 F 中第一棵树的结点个数是()。
- A) $m - n - 1$ B) $n + 1$ C) $m - n + 1$ D) $m - n$
63. 在完全二叉树中,若一个结点是叶结点,则它没有()。
- A) 左子结点 B) 右子结点
 C) 左子结点和右子结点 D) 左子结点、右子结点和兄弟结点
64. 设二叉树根结点的层次为 0,一棵深度为 k 的满二叉树和同样深度的完全二叉树各有 f 个结点和 c 个结点,下列关系不正确的是()。
- A) $f \geq c$ B) $c > f$ C) $f = 2^{k+1} - 1$ D) $c > 2^k - 1$
65. 有一棵非空二叉树(第 0 层为根结点),其第 i 层上至多有()个结点。
- A) $2i$ B) $2^i - 1$ C) $2^{i+1} - 1$ D) i
66. 在下列存储形式中,()不是树的存储形式。
- A) 双亲表示法 B) 孩子链表表示法
 C) 孩子兄弟表示法 D) 顺序存储表示法
67. 对二叉树中的一个结点 x ,在先根序列中的序号为 $\text{pre}(x)$,在后根序列中的序号为 $\text{post}(x)$ 。若二叉树中结点 x 是结点 y 的祖先,下列 4 个条件()正确。
- A) $\text{pre}(x) < \text{pre}(y)$ 和 $\text{post}(x) < \text{post}(y)$
 B) $\text{pre}(x) < \text{pre}(y)$ 和 $\text{post}(x) > \text{post}(y)$
 C) $\text{pre}(x) > \text{pre}(y)$ 和 $\text{post}(x) < \text{post}(y)$
 D) $\text{pre}(x) > \text{pre}(y)$ 和 $\text{post}(x) > \text{post}(y)$

68. 在树 T 中, 结点 x 的度为 $k(k > 1)$, 结点 y 是结点 x 的最右边一个子女, 在与树 T 对应的二叉树中, 下列结论成立的是()。
- A) y 一定是 x 的左子女
B) y 一定是 x 的右子女
C) y 的左子树一定是空二叉树
D) y 的右子树一定是空二叉树
69. 对一棵 70 个结点的完全二叉树, 它有()个非叶结点。
- A) 35
B) 40
C) 30
D) 44
70. 如果 T_2 是由有序树 T 转换而来的二叉树, 那么 T 中结点的后序就是 T_2 中结点的()。
- A) 前序
B) 中序
C) 后序
D) 层次序
71. 顺序查找法适用于存储结构为()的线性表。
- A) 散列存储
B) 压缩存储
C) 顺序存储或链接存储
D) 索引存储
72. 设要将序列 {Q, H, C, Y, P, A, M, S, R, D, F, X} 中的关键码按字母升序重新排序, 下面()是冒泡排序一趟扫描的结果。
- A) F, H, C, D, P, A, M, Q, R, S, Y, X
B) P, A, C, S, Q, D, F, X, R, H, M, Y
C) A, D, C, R, F, Q, M, S, Y, P, H, X
D) H, C, Q, P, A, M, S, R, D, F, X, Y
73. 对一个由 n 个关键码组成的序列, 借助排序过程选出其中最大的关键码, 要求关键码比较次数和移动次数最少, 应当用()方法。
- A) 归并排序
B) 直接插入排序
C) 直接选择排序
D) 快速排序
74. 对数组 $R[1 \cdots n]$ 中的 n 个元素进行排序的某一种方法描述如下:
- 第一步: 令 $h = n$;
第二步: 进行 $h - 1$ 次比较, 从 $R[1], R[2], \dots, R[h]$ 中找出最大的元素 $R[i] (1 \leq i \leq h)$;
第三步: 若 $i \neq h$, 则交换 $R[i]$ 和 $R[h]$;
第四步: 令 $h = h - 1$;
第五步: 若 $h = 1$, 则排序完成, 否则转向第二步。
- 对上述排序方法, 下列选项中不正确的是()。
- A) 排序过程中, 元素的交换次数至少为 0 次
B) 排序过程中, 元素的交换次数至多为 $n - 1$ 次
C) 方法是稳定的
D) 方法是不稳定的
75. 对上题中描述的排序方法, 下列选项中正确的是()。
- A) 排序过程中, 比较的次数与元素的初始排列顺序无关
B) 排序过程中, 比较的次数与元素的初始排列顺序有关
C) 确定数组中最大元素需要比较 $n(n - 1)/2$ 次
D) 确定数组中最小元素需要比较 $n - 1$ 次
76. 对下列 4 个序列用快速排序方法进行排序, 以序列的第一个元素为划分的基准。在第一趟划分过程中, 元素移动次数最多的序列是()。
- A) 70, 75, 82, 90, 23, 16, 10, 68
B) 70, 75, 65, 23, 10, 16, 90, 82
C) 82, 75, 70, 16, 10, 90, 68, 23
D) 23, 10, 16, 70, 82, 75, 68, 90
77. 对以下关键字序列用快速排序法进行排序, () 的速度最慢。
- A) {19, 23, 3, 15, 21, 28, 7}
B) {23, 21, 28, 15, 19, 37}
C) {19, 7, 15, 28, 23, 21, 3}
D) {3, 7, 15, 19, 21, 23, 28}
78. 若查找每个元素的概率相等, 则在长度为 n 的顺序表上查找任一元素的平均查找长度为()。
- A) n
B) $n + 1$
C) $(n - 1)/2$
D) $(n + 1)/2$
79. 对长度为 n 的单链有序表, 若查找每个元素的概率相等, 则查找任一元素的平均查找长度为()。
- A) $n/2$
B) $(n + 1)/2$
C) $(n - 1)/2$
D) $n/4$
80. 对于长度为 18 的顺序存储的有序表, 若采用二分查找, 则查找第 15 个元素的查找长度为()。
- A) 3
B) 4
C) 5
D) 6
81. 对于顺序存储的有序表 {5, 12, 20, 26, 37, 42, 46, 50, 64}, 若采用二分查找, 则查找元素 26 的查找长度为()。
- A) 2
B) 3
C) 4
D) 5
82. 对具有 n 个元素的有序表采用二分查找, 则算法的时间复杂性为()。
- A) $O(n)$
B) $O(n^2)$
C) $O(1)$
D) $O(\log_2 n)$

83. 字符 A、B、C 依次进入一个栈,按出栈的先后顺序组成不同的字符串,则至多可以组成()个不同的字符串。
A) 14 B) 5 C) 6 D) 8

84. 如图 1-47 所示,该二叉树结点的前序遍历的序列为()。
A) E、G、F、A、C、D、B B) E、A、G、C、F、B、D
C) E、A、C、B、D、G、F D) E、G、A、C、D、F、B

85. 如图 1-41 所示,该二叉树结点的中序遍历的序列为()。

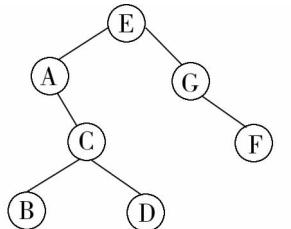


图 1-41 二叉树的遍历图

- A) A、B、C、D、E、G、F B) E、A、G、C、F、B、D
C) E、A、C、B、D、G、F D) B、D、C、A、F、G、E
86. 在具有 n 个结点的二叉链表中,非空的链域个数为()。
A) $n-1$ B) $2n-1$ C) $n+1$ D) $2n+1$
87. 下列排序算法中,()算法在进行一趟相应的排序处理结束后不一定能选出一个元素放到其最终位置上。
A) 直接选择排序 B) 冒泡排序 C) 归并排序 D) 堆排序
88. 若一个元素序列基本有序,则选用()方法较快。
A) 直接插入排序 B) 直接选择排序 C) 堆排序 D) 快速排序
89. 若要从 1000 个元素中得到 10 个最小值元素,最好采用()方法。
A) 直接插入排序 B) 直接选择排序 C) 堆排序 D) 快速排序
90. 若要对 1000 个元素排序,要求既快又稳定,则最好采用()方法。
A) 直接插入排序 B) 归并排序 C) 堆排序 D) 快速排序
91. 若要对 1000 个元素排序,要求既快又节省存储空间,则最好采用()方法。
A) 直接插入排序 B) 归并排序 C) 堆排序 D) 快速排序
92. 在下列排序方法中,空间复杂性为 $O(\log_2 n)$ 的方法为()。
A) 直接选择排序 B) 归并排序 C) 堆排序 D) 快速排序
93. 在平均情况下速度最快的排序方法为()。
A) 直接选择排序 B) 归并排序 C) 堆排序 D) 快速排序
94. 若对 n 个元素进行直接插入排序,则进行第 i 趟排序过程前,有序表中的元素个数为()。
A) n B) $n+1$ C) $n-1$ D) $i-1$
95. 若对 n 个元素进行直接插入排序,则进行任一趟排序的过程中,为寻找插入位置而需要的时间复杂性为()。
A) $O(1)$ B) $O(n)$ C) $O(n^2)$ D) $O(\log_2 n)$
96. 下面 4 种排序方法中,平均查找长度最小的是()。
A) 插入排序 B) 选择排序 C) 快速排序 D) 归并排序
97. 设有关键码序列 (16,9,4,25,15,2,13,18,17,5,8,24),要按关键码值递增的次序排序,采用起泡排序法,第二趟起泡后的结果为()。
A) (15,2,4,18,16,5,8,24,17,9,13,25) B) (4,9,15,2,13,16,17,5,8,18,24,25)
C) (9,4,16,15,2,13,18,17,5,8,24,25) D) (9,16,4,25,2,15,13,18,5,17,8,24)
98. 设有关键码序列 (16,9,4,25,15,2,13,18,17,5,8,24),要按关键码值递增的次序排序,采用初始增量为 4 的希尔排序法,一趟扫描后的结果为()。
A) (15,2,4,18,16,5,8,24,17,9,13,25) B) (2,9,4,25,15,16,13,18,17,5,8,24)
C) (9,4,16,15,2,13,18,17,5,8,24,25) D) (9,16,4,25,2,15,13,18,5,17,8,24)
99. 下列与数据元素有关的叙述中,()是不正确的。
A) 数据元素是数据的基本单位,即数据集合中的个体

- B) 数据元素是有独立含义的数据最小单位
C) 数据元素又称做结点
D) 数据元素又称做记录
100. 下面 4 种排序方法中,要求内存容量最大的是()。
A) 插入排序 B) 选择排序 C) 快速排序 D) 归并排序
101. 堆是()。
A) 完全二叉树 B) 线性表 C) 二叉排序树 D) 平衡二叉树
102. 如果要求一个线性表既能较快地查找,又能适应动态变化的要求,则可采用的方法是()。
A) 分块法 B) 顺序法 C) 二分法 D) 散列法
103. 在对 n 个元素进行直接插入排序的过程中,共需要进行()趟。
A) n B) $n+1$ C) $n-1$ D) $2n$
104. 对 n 个元素进行直接插入排序时间复杂度为()。
A) $O(1)$ B) $O(n)$ C) $O(n^2)$ D) $O(\log_2 n)$
105. 在对 n 个元素进行冒泡排序的过程中,最好情况下的时间复杂度为()。
A) $O(1)$ B) $O(\log_2 n)$ C) $O(n^2)$ D) $O(n)$
106. 在对 n 个元素进行冒泡排序的过程中,至少需要()趟完成。
A) 1 B) n C) $n-1$ D) $n/2$
107. 在对 n 个元素进行快速排序的过程中,最好情况下需要进行()趟。
A) n B) $n/2$ C) $\log_2 n$ D) $2n$
108. 在对 n 个元素进行快速排序的过程中,最坏情况下需要进行()趟。
A) n B) $n-1$ C) $n/2$ D) $\log_2 n$
109. 在对 n 个元素进行快速排序的过程中,平均情况下的时间复杂度为()。
A) $O(1)$ B) $O(\log_2 n)$ C) $O(n^2)$ D) $O(n\log_2 n)$
110. 在对 n 个元素进行快速排序的过程中,最坏情况下的时间复杂度为()。
A) $O(1)$ B) $O(\log_2 n)$ C) $O(n^2)$ D) $O(n\log_2 n)$
111. 在对 n 个元素进行直接插入排序的过程中,算法的空间复杂度为()。
A) $O(1)$ B) $O(\log_2 n)$ C) $O(n^2)$ D) $O(n\log_2 n)$
112. 假定对元素序列(7,3,5,9,1,12,8,15)进行快速排序,则进行第一次划分后,得到的左区间中元素的个数为()。
A) 2 B) 3 C) 4 D) 5
113. 在对 n 个元素进行堆排序的过程中,时间复杂度为()。
A) $O(1)$ B) $O(\log_2 n)$ C) $O(n^2)$ D) $O(n\log_2 n)$
114. 在对 n 个元素进行堆排序的过程中,空间复杂度为()。
A) $O(1)$ B) $O(\log_2 n)$ C) $O(n^2)$ D) $O(n\log_2 n)$
115. 假定对元素序列(7,3,5,9,1,12)进行堆排序,并且采用小根堆,则由初始数据构成的初始堆为()。
A) 1,3,5,7,9,12 B) 1,3,5,9,7,12
C) 1,5,3,7,9,12 D) 1,5,3,9,12,7

1.10 参考答案

- | | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|
| 1. D) | 2. A) | 3. D) | 4. B) | 5. C) | 6. C) | 7. A) | 8. B) |
| 9. D) | 10. D) | 11. A) | 12. C) | 13. B) | 14. C) | 15. B) | 16. A) |
| 17. A) | 18. A) | 19. A) | 20. B) | 21. B) | 22. A) | 23. A) | 24. A) |
| 25. D) | 26. B) | 27. A) | 28. C) | 29. B) | 30. A) | 31. A) | 32. A) |
| 33. D) | 34. D) | 35. C) | 36. C) | 37. A) | 38. C) | 39. C) | 40. B) |
| 41. C) | 42. A) | 43. A) | 44. B) | 45. C) | 46. D) | 47. B) | 48. B) |

49. B)	50. B)	51. C)	52. C)	53. C)	54. D)	55. A)	56. B)
57. B)	58. D)	59. C)	60. D)	61. B)	62. D)	63. C)	64. B)
65. A)	66. D)	67. B)	68. D)	69. A)	70. B)	71. C)	72. D)
73. C)	74. C)	75. D)	76. A)	77. D)	78. D)	79. B)	80. B)
81. C)	82. D)	83. B)	84. C)	85. A)	86. A)	87. C)	88. A)
89. B)	90. B)	91. C)	92. D)	93. D)	94. D)	95. B)	96. C)
97. B)	98. A)	99. B)	100. D)	101. B)	102. A)	103. C)	104. C)
105. D)	106. A)	107. C)	108. B)	109. D)	110. C)	111. A)	112. B)
113. D)	114. A)	115. B)					

第 2 章 程序设计基础

考核知识点

- 程序设计方法与风格
- 结构化程序设计
- 面向对象的程序设计方法、对象、属性及继承与多态性

重要考点提示

- 形成良好的程序设计风格应注意的因素
- 结构化程序设计的主要原则
- 结构化程序的基本结构与特点
- 结构化程序设计的原则和方法的应用
- 面向对象程序设计的主要优点
- 面向对象方法的基本概念(对象、类和实例、消息、继承和多态性)

2.1 程序设计方法与风格

考点 1 程序设计经历的阶段

程序设计是一门技术,需要相应的理论、技术、方法和工具来支持,程序设计主要经历了结构化设计和面向对象的程序设计阶段。

考点 2 良好的编程风格应注意的因素

1. 源程序的文档化

(1)符号名的命名规则:符号名的命名应具有一定的实际含义,以便理解程序功能。

(2)正确的程序注释:注释一般为序言性注释和功能性注释。序言性注释常位于程序开头部分,它包括程序标题、程序功能说明、主要算法、接口说明、程序位置、开发简历、程序设计者、复审者、复审日期及修改日期等。功能性注释一般嵌在源程序体之中,用于描述其后的语句或程序的主要功能。

(3)视觉组织:在程序中利用空格、空行、缩进等技巧使程序层次清晰。

2. 数据说明的方法

(1)数据说明的次序规范化。

(2)说明语句中变量安排有序化。

(3)使用注释来说明复杂数据的结构。

3. 语句的结构

为使程序更简单易懂,语句构造应该简单直接,应注意如下原则:

(1)在一行内只写一条语句。

(2)程序编写应优先考虑清晰性。

(3)程序编写要做到清晰第一,效率第二。

- (4)在保证程序正确的基础上再要求提高效率。
- (5)避免使用临时变量而使程序的可读性下降。
- (6)避免不必要的转移。
- (7)尽量使用库函数。
- (8)避免采用复杂的条件语句。
- (9)尽量减少使用“否定”条件语句。
- (10)数据结构要有利于程序的简化。
- (11)要模块化,使模块功能尽可能单一化。
- (12)利用信息隐蔽,确保每一个模块的独立性。
- (13)从数据出发去构造程序。
- (14)不要修补不好的程序,要重新编写。

4. 输入输出

输入输出方式和风格应尽可能方便用户的使用,应考虑如下原则:

- (1)对输入数据要检验数据的合法性。
- (2)检查输入项的各种重要组合的合理性。
- (3)输入格式要简单,使得输入的步骤和操作尽可能简单。
- (4)输入数据时,应允许使用自由格式。
- (5)应允许默认值。
- (6)输入一批数据时,最好使用输入结束标志。
- (7)在以交互式输入/输出方式进行输入时,要在屏幕上使用提示符明确提示输入的请求,同时在数据输入过程中和输入结束时,应在屏幕上给出状态信息。
- (8)当程序设计语言对输入格式有严格要求时,应保持输入格式与输入语句的一致性。

真题题解

【例1】对建立良好的程序设计风格,下面描述正确的是()。

- | | |
|--------------------|----------------|
| A) 程序应力求简单、清晰、可读性好 | B) 符号的命名只要符合语法 |
| C) 充分考虑程序的执行效率 | D) 程序的注释可有可无 |

【命题目的】本题主要考查程序设计应该遵循的一些原则。

【解题要点】良好的程序设计风格可以减少编码错误,减少程序阅读时间,从而提高软件的开发效率。程序是由人来编写的,为了测试和维护程序,往往还要阅读和跟踪程序,因此程序设计的风格总体而言应该强调简单和清晰,程序必须是可以理解的。所以本题正确答案为A)。

【错解分析】程序设计中,符号的命名不但要符合语法,还要具有一定的实际含义,以便于对程序功能的理解。所以B)选项错误。“清晰第一,效率第二”已经成为当今主导的程序设计风格,所以在充分考虑程序的执行效率时,一定要保证程序清晰、可读。所以C)选项错误。程序的注释能够帮助读者理解程序,良好的程序设计风格应该包括完整的注释。所以D)选项错误。

【考点链接】良好的程序设计应该遵循的原则。

【答案】A)

【例2】序言性注释的主要内容不包括()。

- | | | | |
|----------|----------|----------|----------|
| A) 模块的接口 | B) 数据的描述 | C) 模块的功能 | D) 数据的状态 |
|----------|----------|----------|----------|

【命题目的】考查序言性注释的主要内容。

【解题要点】序言性注释通常位于每个程序的开头部分,它给出程序的整体说明,主要描述内容可以包括:程序标题、程序功能说明、主要算法、接口说明、程序位置、开发简历、程序设计者、复审者、复审日期、修改日期等。所以本题正确答案为D)。

【错解分析】对序言性注释掌握得不够准确导致本题错误。

【考点链接】功能性注释的主要内容。

【答案】D)

【例3】模块的功能性注释的主要内容不包括()。

- A) 程序段的功能 B) 语句的功能 C) 模块的功能 D) 数据的状态

【命题目的】考查功能性注释的主要内容。

【解题要点】功能性注释的位置一般嵌在源程序体中,主要描述其后的语句或者程序的主要功能。所以本题正确答案为 C)。

【错解分析】对功能性注释掌握得不够准确导致本题错误。

【考点链接】序言性注释的主要内容。

【答案】C)

2.2 结构化程序设计

考点 3 结构化程序设计的原则

一个结构化程序就是用高级语言表示的结构化算法,这种程序便于编写、阅读、修改和维护,这就减少了程序出错的机会,提高了程序的可靠性,保证了程序的质量。结构化程序设计强调程序设计风格和程序结构的规范化,提倡清晰的结构,如图 2-1 所示,结构化程序设计采取如下方法:

(1)自顶向下:即先考虑总体,后考虑细节;先考虑全局目标,后考虑局部目标。这种程序结构按功能划分为若干个基本模块,这些模块形成一个树状结构。

(2)逐步求精:对复杂问题,应设计一些子目标做过渡,逐步细化。

用这种方法逐步分解,直到作者认为可以直接将各小段表达为文字语句为止。这种方法就叫做“自顶而下,逐步求精”。这种设计方法的过程是将问题求解由抽象逐步具体化的过程。用这种方法便于验证算法的正确性,在向下一层展开之前应仔细检查本层设计是否正确,只有上一层是正确的才能向下细化。如果每一层设计都没有问题,则整个算法就是正确的。这种方法容易保证整个算法的正确性。

(3)模块化:模块化是把程序要解决的总目标分解为分目标,再进一步分解为具体的小目标,把每个小目标称为一个模块。

(4)限制使用 GOTO 语句。

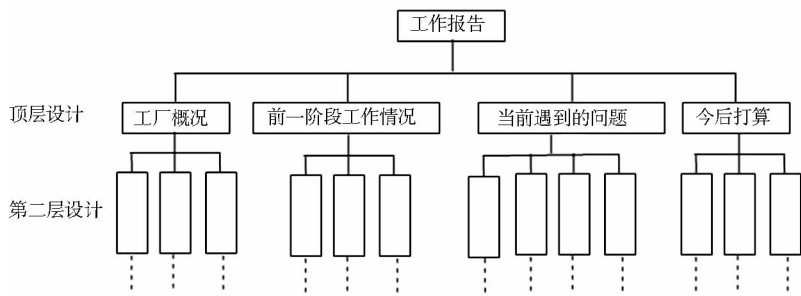


图 2-1 自顶而下、逐步求精方法示意图

考点 4 结构化程序的基本结构与特点

1966 年,Boehm 和 Jacopini 证明了程序设计语言仅仅使用顺序、选择和重复三种基本控制结构就足以表达出各种其他形式结构的程序设计方法。遵循程序结构化的设计原则,按结构化程序设计方法设计出的程序易于理解、使用和维护;可以提高编程工作的效率,降低软件的开发成本。

(1)顺序结构:顺序结构是一种简单的程序设计结构,顺序结构自始至终严格按照程序中语句的先后顺序逐条执行,是最基本、最普遍的结构形式,如图 2-2 所示。

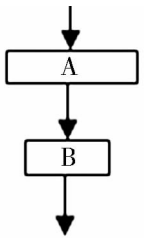


图 2-2 顺序结构

(2) 选择结构:又称为分支结构,它包括简单选择和多分支选择结构,如图 2-3 所示。

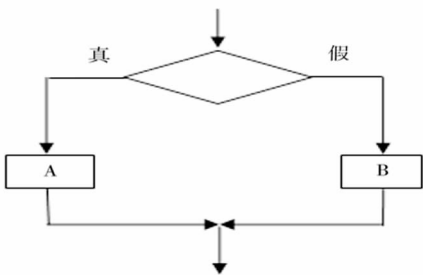


图 2-3 选择结构

(3) 重复结构:重复结构又称为循环结构,它根据给定的条件,判断是否需要重复执行某一相同的或类似的程序段。

在程序设计语言中,重复结构对应两类循环语句,对先判断后执行的循环体称为当型循环结构,对先执行循环体后判断的称为直到型循环结构,如图 2-4 所示。

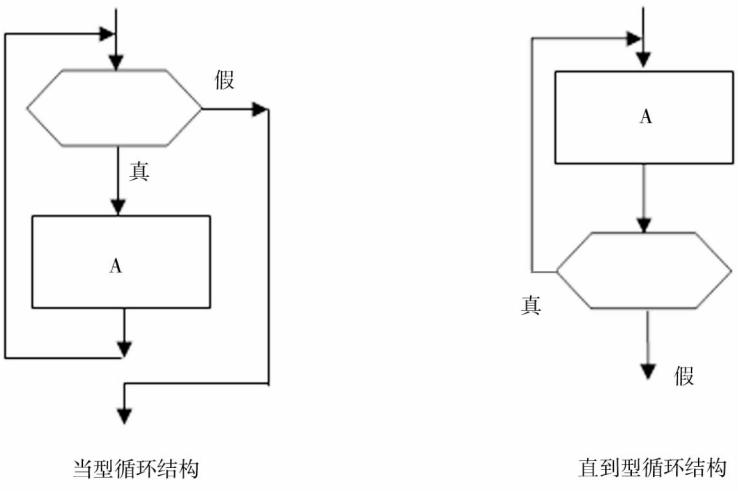


图 2-4 两种循环结构

考点 5 结构化程序设计原则和方法的应用

结构化程序设计中,要注意把握如下要素:

- (1) 使用程序设计语言中的顺序、选择、循环等有限的控制结构表示程序的控制逻辑。
- (2) 选用的控制结构只准许有一个入口和一个出口。
- (3) 程序语言组成容易识别的块,每块只有一个入口和一个出口。
- (4) 复杂结构应该用嵌套的基本控制结构进行组合嵌套来实现。
- (5) 语言中所没有的控制结构,应该采用前后一致的方法来模拟。
- (6) 尽量避免 GOTO 语句的使用。

虽然结构化程序设计方法具有很多优点,但它仍是一种面向过程的程序设计方法。它把数据和处理数

据的过程分离为相互独立的实体,当数据结构改变时,所有相关的处理过程都要进行相应的修改,每一种相对于老问题的新方法都要带来额外的开销,程序的可重用性差。一个好的软件应该随时响应用户的任何操作,而不是请用户按照既定的步骤循规蹈矩地使用,因此对这种软件的功能很难用过程来描述和实现,如果仍用面向过程的方法,开发和维护都将很困难。

真题题解

【例 1】结构化程序设计方法的结构不包括()。

- A) 顺序结构 B) 分支结构 C) 循环结构 D) 跳转结构

【命题目的】考查结构化设计的基本知识。

【解题要点】结构化设计是指采用自顶向下、逐步求精、模块化和尽量避免 GOTO 跳转语句的设计方法,其程序结构按功能划分为若干个基本模块,这些模块形成一个树状结构,每一个模块均采用顺序、选择(分支)和循环 3 种基本结构组成的。

【错解分析】由以上分析知 A) 选项、B) 选项和 C) 选项均是结构化程序设计的一部分,只有 D) 选项不是结构化程序设计的思想。

【考点链接】结构化程序设计方法。

【答案】D)

【例 2】结构化程序设计的 3 种基本结构是()。

- A) 顺序、选择、重复 B) 递归、嵌套、调用
C) 过程、子过程、主程序 D) 顺序、转移、调用

【命题目的】考查程序设计的 3 种基本结构。

【解题要点】经过推理证明,程序设计语言仅仅使用顺序、选择和重复 3 种基本控制结构就足以表达出各种其他形式结构的程序设计方法。所以本题正确答案为 A)。

【错解分析】B) 选项、C) 选项和 D) 选项都不是程序设计的 3 种基本结构。

【考点链接】结构化程序设计的特点和应用。

【答案】A)

2.3 面向对象的程序设计

考点 6 面向对象的方法

客观世界中任何一个事物都可以被看成是一个对象,面向对象方法的本质就是主张从客观世界固有的事物出发来构造系统,提倡用人类在现实生活中常用的思维方法来认识、理解和描述客观事物,强调最终建立的系统能够映射问题域,也就是说,系统中的对象以及对象之间的关系能够如实地反映问题域中固有事物及其关系。从计算机的角度来看,一个对象应该包括两个要素:一是数据;二是需要进行的操作。对象就是一个包含数据以及与这些数据有关的操作的集合,如图 2-5 所示。面向对象就是运用对象、类、继承、封装、消息、结构与连接等面向对象的概念对问题进行分析、求解的系统开发技术。

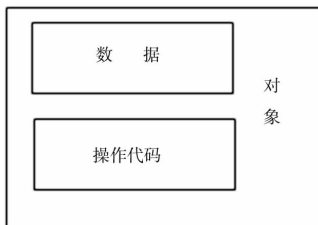


图 2-5 对象的组成

面向对象有如下主要优点: