



出现频率最高的

100种

典型题型 精解精练

数据 结构

- ◆ 研究常考题型是考试过关的捷径
- ◆ 实战预测试卷是加分致胜的法宝

费 宁 汪胡青 张 宏 编著

2009

考研必备

巧学、巧练、巧过关



清华大学出版社

全国硕士研究生入学考试考题排行榜系列丛书

出现频率最高的 100 种典型题型

精解精练——数据结构

费宁 汪胡青 张宏 编著

地址：北京市清华大学学大入编

邮编：100084

电话：010-62786244

客户服务热线：010-62736969

字数：499千字

印次：2008年5月第1次印刷

责任编辑：吴立同

责任校对：范王李

责任印制：林梓

出版发行：清华大学出版社

http://www.tup.com.cn

c-service@tup.tsinghua.edu.cn

社址：010-62770175

社址：010-62770175

社址：北京市鑫山鑫印刷有限公司

印：010-62770175

印：010-62770175

印：010-62770175

印：010-62770175

印：010-62770175

印：010-62770175

清华大学出版社

北京 010-62770175

010-62770175

内 容 简 介

考研作为一种选拔性水平考试,试题规范,规律性很强,不少题型反复出现,把这些反复出现的题型按考试出现频率整理归类,并提供解题思路,可以帮助考生节省宝贵的复习时间,提高应试效率,对考生迎考大有帮助。本书正是基于这一思路,由资深考研辅导老师精心编写而成。

全书共分11章,第1~10章归纳整理了最常考的100种典型题型,具体内容包括:数据结构基本概念、线性表、栈与队列、数组与字符串、递归、树、集合、图、内排序、文件与外排序,第11章为全国硕士研究生入学考试数据结构全真预测试题及其参考解答。每种题型分为三个板块:真题分析、题型点睛和即学即练。真题分析以历届考研真题为实例进行分析,旨在让读者彻底明白这类题型的解法;题型点睛浓缩了该题型的要点,并加以讲解与点评,便于读者理解与记忆;即学即练中作者设计了部分试题,让读者即学即练,即练即会,以达到举一反三的功效。本书附录给出了各章即学即练试题的详细解析与参考答案。

本书以广大考研读者为主要对象,帮助考生在短时间内获取较大收益,同时可作为考研辅导班的培训教材以及高等院校相关师生的教学参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

出现频率最高的100种典型题型精解精练. 数据结构/费宁,汪胡青,张宏编著.
—北京:清华大学出版社,2007.12

(全国硕士研究生入学考试考题排行榜系列丛书)

ISBN 978-7-302-16679-5

I. 出… II. ①费…②汪…③张… III. 数据结构—研究生—入学考试—解题
IV.G643-44

中国版本图书馆CIP数据核字(2007)第200933号

责任编辑:何立兵

责任校对:李玉茹

责任印制:科海

出版发行:清华大学出版社

<http://www.tup.com.cn>

c-service@tup.tsinghua.edu.cn

社总机:010-62770175

投稿咨询:010-62772075

地址:北京清华大学学研大厦A座

邮编:100084

邮购热线:010-62786544

客户服务:010-62776969

印装者:北京市鑫山源印刷有限公司

经销:全国新华书店

开本:185×260 印张:20.5 字数:499千字

版次:2008年2月第1版 印次:2008年2月第1次印刷

印数:1~5000

定价:34.00元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:82896445 产品编号:027041-01

前言

读考题排行榜 走成功捷径路

全国硕士研究生入学考试试题是广大工作在教学一线的骨干教师和参加命题的专家教授的智慧 and 劳动结晶, 考试试题既反映了考试大纲对考生知识、能力等的要求, 又蕴含着考研命题的基本原则和趋势, 对于广大准备考研的考生而言, 也是一笔宝贵财富。

为了给广大考生提供一套高效实用的试题导航标准应试教材, 我们在广泛调研和充分论证的基础上, 听取资深专家及众多考生的建议, 组织编写了这套《全国硕士研究生入学考试考题排行榜系列丛书》, 其目的是帮助考生在复习阶段, 浓缩考试中出现频率最高的题型, “把书读薄”, 以做到成竹在胸, 引导考生在短时间内快速突破过关。

◆ 丛书书目

丛书第一批推出 8 本:

- ◆ 出现频率最高的 100 种典型题型精解精练——数学一
- ◆ 出现频率最高的 100 种典型题型精解精练——数学二
- ◆ 出现频率最高的 100 种典型题型精解精练——数学三
- ◆ 出现频率最高的 100 种典型题型精解精练——数学四
- ◆ 出现频率最高的 100 种典型题型精解精练——数据结构
- ◆ 出现频率最高的 100 种典型题型精解精练——C 语言程序设计
- ◆ 出现频率最高的 100 种典型题型精解精练——操作系统
- ◆ 出现频率最高的 100 种典型题型精解精练——电路

◆ 关于本书

本书通过深入分析历年真题特点, 归纳整理出了硕士研究生入学考试“数据结构”联考的 100 种题型, 并依据《数据结构》(第二版) 的章节顺序, 将这 100 种题型分成 10 章进行解析与点评, 便于考生更快地了解和掌握复习的重点, 发现命题的规律, 明确复习方向, 节省宝贵的复习时间。由于某些题型几乎是年年出现, 所以本书可以令考生更高效地复习与掌握必考题型与知识点。这也正是本书的最大特色: 省时、高效、高命中率。

书中将全国几十所重点高校近 5 年考研试卷中的同一题型试题, 归纳整理成 100 种高频题型 (即 TOP1~TOP100), 对每种题型进行了详细分析并给出参考解答, 便于考生复习该内容时可以了解: 这种题型考过什么样的题目, 常与哪些知识点联系起来出题, 从哪个角度命题, 等等。每种题型具体分为如下三个板块:

- 真题分析。以重点高校近 5 年的考研真题为实例,分析解题思路,实际就是进行破题,最终找出解题方法。分析以后给出详细的解答,旨在让考生掌握解题方法和技巧,以及这些方法技巧在每个具体问题中的灵活运用,彻底明白这类题型的解法。
- 题型点睛。浓缩该题型的要点,给出该题型的相关知识点和解题的一般方法或步骤,并加以讲解分析,便于考生理解与记忆。
- 即学即练。给出部分试题,让考生学过“真题分析”和“题型点睛”后就进行做题练习,以便更快更好地掌握所练题型的相关知识点和解题的一般方法或步骤,以达到举一反三、触类旁通的功效。

本书还提供了 3 套全国硕士研究生入学考试数据结构全真预测试题并附有具体的参考解答,可以供考生在考前实战演练。为了让考生及时掌握自己的学习效果,书中最后还给出了“即学即练”中试题的具体解答,以便考生自查。

◆ 读者对象

本套丛书以全国硕士研究生入学考试的考生为主要读者对象,特别适合希望在较短时间内取得较大进步的广大应试考生,也可作为相关考试培训班的培训教材,以及高校师生的教学参考书。

◆ 关于作者

丛书由一线教学及考试研究专家分工编写。作者均长期从事这方面的教学和研究工作,积累了丰富的经验,对研究生入学考试颇有研究(其中大多数编写者多年参加真题阅卷及相关培训与辅导工作)。本书由费宁、汪胡青和张宏执笔编写而成。参与本丛书组织、指导、编写、审校和资料收集的人员有(以姓氏笔画为序):王伦夫、石雪梅、刘志高、孙建东、余雪勇、吴蕾、张宏、李千目、李勇智、李海、杜松、杨帮华、汪志宏、汪胡青、罗玮、费宁、徐倩、袁堃、彭宜青、葛武滇等,在此对诸位作者付出的辛勤劳动表示衷心的感谢。

◆ 特别致谢

在此,首先对丛书所选用的参考文献的著作者,以及丛书所引用试题的出题老师和相关单位表示真诚的感谢。

◆ 互动交流

由于时间仓促,书中不妥之处,敬请广大读者批评指正。读者的进步,是我们最大的心愿。您如果发现书中有任何疑惑之处,请与我们交流。

作者的联系方式: iteditor@126.com

出版社的联系方式: feedback@khp.com.cn

作者

2008 年 1 月

目 录

第 1 章 绪论	1
TOP1: 数据结构的基本概念	1
TOP2: 算法的时间复杂度和空间复杂度	3
TOP3: 数据的存储结构和逻辑结构	5
TOP4: 根据要求设计数据结构	7
第 2 章 线性表	9
TOP5: 链表的插入和删除	9
TOP6: 链表的逆置运算	11
TOP7: 链表的运算(并集、交集等)	13
TOP8: 循环链表和双向链表的操作	18
TOP9: 链表操作的时间复杂度和空间复杂度	22
TOP10: 多项式的算术运算	25
第 3 章 栈与队列	27
TOP11: 进栈和出栈	27
TOP12: 表达式的计算	31
TOP13: 队列及出入队操作	34
第 4 章 数组与字符串	39
TOP14: 数组的操作	39
TOP15: 数组的行优先、列优先顺序存储地址计算	46
TOP16: 稀疏矩阵的三元组及十字链表存储	47
TOP17: 对角矩阵	50
TOP18: 矩阵和数组的转换	52
TOP19: 字符串转换	55
TOP20: 模式匹配与 KMP 算法	59
第 5 章 递归	63
TOP21: 递归	63
TOP22: 广义表的运算	65
TOP23: 广义表的存储表示	66

第 6 章 树.....	71
TOP24: 树的定义与性质证明	71
TOP25: 二叉树的定义与性质证明	73
TOP26: 二叉树的存储表示	75
TOP27: 满二叉树	82
TOP28: 完全二叉树的判定	83
TOP29: 二叉树的高度	84
TOP30: 二叉树叶子结点和非叶子结点的计算	86
TOP31: 二叉树的前序遍历	87
TOP32: 二叉树的中序遍历	89
TOP33: 二叉树的后序遍历	90
TOP34: 二叉树遍历方式的转换	91
TOP35: 二叉树的遍历确定唯一二叉树	92
TOP36: 二叉树遍历的非递归算法	95
TOP37: 二叉树结点的操作	98
TOP38: 二叉树共同祖先的判断	100
TOP39: 二叉树的复制和相似判定	103
TOP40: 线索二叉树的性质和定义	104
TOP41: 前序线索二叉树	105
TOP42: 中序线索二叉树	106
TOP43: 后序线索二叉树	111
TOP44: 线索二叉树的遍历	112
TOP45: 线索二叉树的插入	113
TOP46: 森林与二叉树的转换	116
TOP47: 堆和堆排序	118
TOP48: 堆的插入和调整	119
TOP49: 哈夫曼树的性质与建立	121
TOP50: 哈夫曼编码	122
第 7 章 集合	124
TOP51: 并查集的定义和实现	124
TOP52: 二叉排序树的查找和遍历	125
TOP53: 二叉排序树的插入和删除	129
TOP54: 二叉平衡树的定义和性质证明	132
TOP55: 二叉平衡树的高度	134
TOP56: 二叉平衡树的平衡旋转	135



TOP57: B-树的定义和性质	136
TOP58: B-树的插入	137
TOP59: 键树的定义和性质	138
TOP60: 查找的一般方法	140
TOP61: 折半查找法	142
TOP62: 查找的时间复杂度和空间复杂度	145
TOP63: 哈希表和哈希函数	147
TOP64: 线性探查法解决冲突调节	149
TOP65: 哈希函数与二次探查法	151
第 8 章 图	153
TOP66: 图的基本概念	153
TOP67: 顶点的入度和出度	154
TOP68: 图的邻接矩阵表示法	155
TOP69: 图的邻接表表示法	157
TOP70: 图的表示方法的转换	163
TOP71: 图的深度优先遍历	164
TOP72: 图的宽度(广度)优先遍历	166
TOP73: 图遍历的非递归算法	169
TOP74: 拓扑排序	170
TOP75: 关键路径	173
TOP76: 普里姆算法	175
TOP77: 克鲁斯科尔(Kruskal)算法	178
TOP78: 迪杰斯特拉(Dijkstra)算法	180
TOP79: 弗洛伊德(Floyd)算法	182
第 9 章 内排序	185
TOP80: 排序的时间复杂度和空间复杂度	185
TOP81: 排序方法的比较	186
TOP82: 排序方法的选择和实际应用	188
TOP83: 简单选择排序	189
TOP84: 比较排序	192
TOP85: 直接插入排序	193
TOP86: 冒泡排序	195
TOP87: 快速排序的概念和应用	197
TOP88: 限定条件下快速排序的编程	200
TOP89: 归并排序	204

TOP90: 基数排序	206
TOP91: 希尔排序	207
TOP92: 多关键字排序	208
TOP93: 排序的稳定性	209
第 10 章 文件与外排序	211
TOP94: 文件的基本概念和组织方式	211
TOP95: 外排序的基本过程	213
TOP96: 顺序文件	214
TOP97: 倒排文件	215
TOP98: B+树的定义和性质	216
TOP99: 竞赛树	217
TOP100: 动态存储管理	218
第 11 章 全国硕士研究生入学考试数据结构全真预测试题及其参考解答	221
硕士研究生入学考试数据结构全真预测试题一	221
硕士研究生入学考试数据结构全真预测试题一参考解答	226
硕士研究生入学考试数据结构全真预测试题二	232
硕士研究生入学考试数据结构全真预测试题二参考解答	236
硕士研究生入学考试数据结构全真预测试题三	242
硕士研究生入学考试数据结构全真预测试题三参考解答	248
附录 习题解答	253
参考文献	319

第1章 绪论

TOP1: 数据结构的基本概念

【真题分析】

【真题 1】(北京航空航天大学)

(1)“数据结构”课程是计算机专业的基础课还是专业课，或者专业基础课？

(2)学习“数据结构”课程需要哪些课程作为它的基础(举两门课程为例)？若没有这些知识，对学习“数据结构”课程可能会产生哪些影响？请举例说明(不超过 100 字)。

(3)“数据结构”课程将为哪些课程的学习奠定必要的基础？哪些课程(举两门课程为例)用到了“数据结构”课程的哪些知识(不超过 100 字)？

分析：此题主要考查“数据结构”课程的性质，并以此强调其在计算机科学中的重要性。

解答：

(1)“数据结构”课程是计算机专业的专业基础课。

(2)学习“数据结构”课程需要一些课程作为它的基础，如“C 语言”与“离散数学”。若没有“C 语言”或其他语言基础(如 Pascal 或 C++)，学生就难以理解描述数据结构及其算法的类 C、类 Pascal 或类 C++，更重要的是会造成学生上机环节的困难，影响该课程的学习。同样，“离散数学”课程也是“数据结构”课程的理论基础，其中的集合、树及图等理论知识为“数据结构”课程的学习提供了重要的理论基础。

(3)“数据结构”课程将为“编译原理”、“数据库系统”和“操作系统”等课程的学习奠定必要的基础。例如，“编译原理”的表达式求解用到“数据结构”的栈知识，“编译原理”的符号表管理技术用到“数据结构”的哈希查找知识，“数据库系统”的存储用到“数据结构”的 B+树知识，“操作系统”的设备链表管理用到“数据结构”的线性链表知识，“操作系统”的最短作业优先服务用到“数据结构”的队列知识。

【真题 2】(中山大学)数据元素之间的关系在计算机中有几种表示方法？各有什么特点？

分析：数据元素的存储方式是数据结构基本概念中的常考内容，考生需要掌握常见的存储方式和各自的特点，并且在学习的过程中能够和具体的数据结构相联系，对每种数据存储方式给出相应的实现方法。

解答：四种表示方法如下。



(1) 顺序存储方式。数据元素按顺序存放，每个存储结点只含一个元素。存储位置反映数据元素间的逻辑关系。存储密度大，但有些操作（如插入、删除）效率较低。

(2) 链式存储方式。每个存储结点除包含数据元素信息外，还包含一组（至少一个）指针。指针反映数据元素间的逻辑关系。这种方式不要求存储空间连续，便于动态操作（如插入、删除等），但存储空间开销大（用于指针），不能进行折半查找。

(3) 索引存储方式。除数据元素存储在一片地址连续的内存空间外，尚需建立一个索引表，索引表中的索引指示存储结点的存储位置（下标）或存储区间端点（下标），兼有静态和动态特性。

(4) 散列存储方式。通过散列函数（也叫哈希函数）和解决冲突的方法将关键字散列在连续、有限的地址空间内，并将散列函数的值解释成关键字所在元素的存储地址，这种存储方式称为散列存储。其特点是存取速度快，只能按关键字随机存取，不能顺序存取，也不能折半存取。

【真题 3】（北京邮电大学）数据类型和抽象数据类型是如何定义的？两者有何相同和不同之处？抽象数据类型的主要特点是什么？使用抽象数据类型的主要好处是什么？

分析：本题旨在考查数据类型和抽象数据类型的区别，需要考生掌握抽象数据类型的特点和应用范围。

解答：数据类型是程序设计语言中的一个概念，它是一个值的集合和定义在这个值集上的一组操作的总称，如 C 语言中的整型、实型、字符型等。整型值的范围（对具体机器都应有整数范围），其操作有加、减、乘、除、求余等。实际上，数据类型是厂家提供给用户的已实现了的数据结构。“抽象数据类型（ADT）”指一个数学模型及定义在该模型上的一组操作。“抽象”的意义在于数据类型的数学抽象特性。抽象数据类型的定义仅取决于它的逻辑特性，而与其在计算机内部如何表示和实现无关。无论数据类型的内部结构如何变化，只要它的数学特性不变，就不影响它的外部使用。抽象数据类型和数据类型实质上是同一个概念。此外，抽象数据类型的范围更广，它已不再局限于机器已定义和实现的数据类型，还包括用户在设计软件系统时自行定义的数据类型。使用抽象数据类型定义的软件模块含定义、表示和实现三部分，这三部分封装在一起，对用户透明（提供接口），而不必了解其实现细节。抽象数据类型的出现使程序设计不再是“艺术”，而是向“科学”迈进了一步。

【题型点睛】

深刻理解“数据结构”课程的性质和内容，就不难掌握和理解它与其他课程之间的关系。如数据结构算法的描述和实现需要借助一门高级语言，而目前比较流行的高级语言有 C、C++、Java、Pascal 等，因而 C 和 Pascal 可作为“数据结构”课程的前续课程。同理，“操作系统”、“编译原理”及“数据库系统原理”等课程中也需用到大量的数据结构知识，如“操作系统”中的作业队列、就绪进程队列及设备队列等，“编译原理”中借用栈来进行表达式求值等。

【即学即练】

【习题 1】(四川大学) 算法的计算量的大小称为计算的 ()。

- A. 效率
C. 现实性

- B. 复杂性
D. 难度

【习题 2】(燕山大学) 数据结构是一门研究什么内容的学科?

TOP2: 算法的时间复杂度和空间复杂度

【真题分析】

【真题 1】(清华大学) 斐波那契数列 F_n 定义如下:

$$F_0=0, \quad F_1=1, \quad F_n=F_{n-1}+F_{n-2}, \quad n=2, 3, \dots$$

请就此斐波那契数列回答下列问题。

- (1) 在递归计算 F_n 时, 需要对较小的 $F_{n-1}, F_{n-2}, \dots, F_1, F_0$ 精确计算多少次?
- (2) 如果用大 O 表示法, 试给出递归计算 F_n 时, 递归函数的时间复杂度为多少?

分析: 本题主要考查递归算法的时间复杂度。考生应该比较熟悉斐波那契数列, 只要具备一定的数学功底, 解决该题应该没有什么困难。但是, 通过该题需要掌握将实际问题转换成数据结构, 进而能够通过数学分析得到时间复杂度的能力。

解答:

(1) 由斐波那契数列的定义可得

$$\begin{aligned} F_n &= F_{n-1} + F_{n-2} \\ &= 2F_{n-2} + F_{n-3} \\ &= 3F_{n-3} + 2F_{n-4} \\ &= 5F_{n-4} + 3F_{n-5} \\ &= 8F_{n-5} + 5F_{n-6} \\ &\dots \\ &= pF_1 + qF_0 \end{aligned}$$

设 F_m 的执行次数为 B_m ($m=0, 1, 2, \dots, n-1$), 由以上等式可知, F_{n-1} 被执行一次, 即 $B_{n-1}=1$; F_{n-2} 被执行两次, 即 $B_{n-2}=2$; 直至 F_1 被执行 p 次, F_0 被执行 q 次, 即 $B_1=p, B_0=q$ 。 B_m 的执行次数为前两等式第一因式系数之和, 即 $B_m=B_{m-1}+B_{m-2}$, 再有 $B_{n-1}=1$ 和 $B_{n-2}=2$, 这也是一个斐波那契数列。可以解得



$$B_m = \frac{\sqrt{5}}{5} \left[\left(\frac{1+\sqrt{5}}{2} \right)^{n-m+2} - \left(\frac{1-\sqrt{5}}{2} \right)^{n-m+2} \right] \quad (m=0,1,2,\dots,n-1)$$

(2) 时间复杂度为 $O(n)$ 。

【真题2】(南京理工大学) 下面程序段的时间复杂度为_____。

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

分析: 该题考查单循环条件限制下时间复杂度的计算, 比较简单。比较复杂的是多重循环条件限制下时间复杂度的计算, 需要注意循环结束的条件。

解答: $O(n)$ 。

【真题3】(北京航空航天大学) 有实现同一功能的两个算法 A1 和 A2, 其中 A1 的时间复杂度为 $T1=O(2^n)$, A2 的时间复杂度为 $T2=O(n^2)$, 仅就时间复杂度而言, 请具体分析这两个算法哪一个更好。

分析: 根据数据结构中对大 O 的定义, 本题很容易解答。比较时间复杂度的题型很常见, 其中往往有对数和次方运算, 需要借助于一些简单的数学转换来分析。

解答: 对算法 A1 和 A2 的时间复杂度 T1 和 T2 取对数, 得 $\log_2 2^n$ 和 $2\log_2 n$ 。显然, 算法 A2 好于 A1。

【题型点睛】

1. 一般情况下, 算法中基本操作重复执行的次数是问题规模 n 的某个函数, 用 $T(n)$ 表示。若有某个辅助函数 $f(n)$, 使得当 n 趋近于无穷大时, $T(n)/f(n)$ 的极限值为不等于零的常数, 则称 $f(n)$ 是 $T(n)$ 的同数量级函数, 记作 $T(n)=O(f(n))$, 称 $O(f(n))$ 为算法的渐进时间复杂度, 简称时间复杂度。与时间复杂度类似, 空间复杂度是指算法在计算机内执行时所需存储空间的度量。

2. 数据结构的时间复杂度和空间复杂度是数据结构中非常重要的内容, 也是数据结构的常考内容。考生需要理解时间复杂度大 O 法的计算方法, 掌握时间复杂度的比较和分析方法, 能够计算递归或者循环算法的时间复杂度, 并且能够将实际算法的程序步转化为大 O 表示法; 同时, 能够分析常见数据存储方式的空间复杂度以及在特定空间复杂度下的算法转换方法。

【即学即练】

【习题1】(南京理工大学) 下面说法错误的是()。

- (1) 算法原地工作的含义是指不需要任何额外的辅助空间
- (2) 在相同的规模 n 下, 复杂度为 $O(n)$ 的算法在时间上总是优于复杂度为 $O(2n)$ 的算法
- (3) 所谓时间复杂度是指最坏情况下, 估算算法执行时间的一个上界
- (4) 同一个算法, 其实现语言的级别越高, 执行效率就越低

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

【习题 2】(吉林工业大学) 有下列运行时间函数:

- (1) $T_1(n) = 1000$; (2) $T_2(n) = n^2 + 1000n$; (3) $T_3(n) = 3n^3 + 100n^2 + n + 1$;

分别写出相应的大 O 表示的运算时间。

【习题 3】(哈尔滨工业大学) 下列函数中, 渐进时间复杂度最小的是 ()。

- A. $T_1(n) = n \log_2 n + 5000n$
C. $T_3(n) = n \log_2 2n - 6000n$
- B. $T_2(n) = n^2 - 8000n$
D. $T_4(n) = 2n \log_2 n - 7000n$

TOP3: 数据的存储结构和逻辑结构

【真题分析】

【真题 1】(浙江大学) 回答下列问题:

(1) 在“数据结构”课程中, 数据的逻辑结构, 数据的存储结构及数据的运算之间存在着怎样的关系?

(2) 若逻辑结构相同但存储结构不同, 则为不同的数据结构。这样的说法对吗? 举例说明之。

(3) 在给定的逻辑结构及其存储表示上可以定义不同的运算集合, 从而得到不同的数据结构。这样说法对吗? 举例说明之。

(4) 评价各种不同数据结构的标准是什么?

分析: 数据结构的逻辑结构和存储结构的特点是数据结构基本概念中的重要内容, 该题重在考查考生对这两个基本概念的掌握以及对它们之间的关系的理解。

解答:

(1) 数据的逻辑结构反映数据元素之间的逻辑关系(即数据元素之间的关联方式或邻接关系)。数据的存储结构是数据结构在计算机中的表示, 包括数据元素的表示及其关系的表示。数据的运算是对于数据定义的一组操作, 运算是定义在逻辑结构上的, 和存储结构无关, 而运算的实现则依赖于存储结构。

(2) 逻辑结构相同但存储结构不同, 可以是不同的数据结构。例如, 线性表的逻辑结构属于线性结构, 采用顺序存储结构称为顺序表, 而采用链式存储结构称为线性链表。

(3) 栈和队列的逻辑结构相同, 其存储表示也可相同(顺序存储和链式存储), 但由于其运算集合不同而成为不同的数据结构。

(4) 数据结构的评价标准非常复杂, 可以考虑两个方面: 一是所选数据结构是否准确、完整地刻画了问题的基本特征; 二是是否容易实现(如对数据的分解是否恰当; 逻辑结构的选择是否适合于运算的功能, 是否有利于运算的实现; 基本运算的选择是否恰当)。



【真题2】(中山大学)连续存储设计时,存储单元的地址()。

- A. 一定连续
B. 一定不连续
C. 不一定连续
D. 部分连续,部分不连续

分析:存储地址是程序在执行时由系统分配的,与采用何种设计方式没有关系,事实上是可能连续,也可能不连续,可能部分连续;也可能全部连续。

解答:C

【真题3】(华南理工大学)对数据结构基本操作的设计,其最重要的准则是实现应用程序与存储结构的独立。()

分析:数据结构这门科学研究的初衷就是分离数据的操作和实际存储方式,所以此题正确。不过该题有一点歧义,实际上,应用程序无法完全独立于存储结构。

解答:正确。

【真题4】(北京理工大学)试举一例,说明对相同的逻辑结构,同一种运算在不同的存储方式下实现,其运算效率不同。

分析:对同样的逻辑结构、不同的存储方式的研究贯穿了整个“数据结构”课程,这样的案例很多,在线性表、树和查找等章节中均有详细阐述。

解答:线性表中的插入、删除操作,在顺序存储方式下平均移动近一半的元素,时间复杂度为 $O(n)$;而在链式存储方式下,插入和删除的时间复杂度都是 $O(1)$ 。

【题型点睛】

1. 数据的逻辑结构是指结构定义中的“关系”,描述的是数据元素之间的逻辑关系。而数据的物理结构(又称存储结构)是数据结构在计算机中的表示(又称映像),它包括数据元素的表示和关系的表示。这样使得逻辑上相邻的数据元素,物理上未必相邻。例如,父子关系可以看成是逻辑关系,逻辑上是相邻的;但他们未必生活在同一个地方,即在物理上可能生活在不同的地方,甚至在不同的国家。

2. 从数学的角度来看,数据的物理结构是数据的逻辑结构的函数映射,是由逻辑地址到物理地址的映射。逻辑上相邻的地址可能映射到物理上不相邻的存储地址,但数据元素的内在关系保持不变。

3. 数据的存储方式有四种,分别是顺序存储方式、链式存储方式、索引存储方式和散列存储方式,考生需掌握这四种方式的特点和区别,还需要掌握这几种存储方式的优点和缺点,以及在具体情况下应该采取何种存储方式。

【即学即练】

【习题1】(北方交通大学)以下哪一个术语与数据的存储结构无关?()

- A. 栈
B. 哈希表
C. 线索树
D. 双向链表

【习题2】(山东大学)数据的物理结构是指数据在计算机内的实际存储形式。()

【习题3】(上海交通大学)数据的逻辑结构说明数据元素之间的顺序关系,它依赖于计算机的储存结构。()

TOP4: 根据要求设计数据结构

【真题分析】

【真题1】(西安电子科技大学,北京科技大学)当为解决某一问题而选择数据结构时,应从哪些方面考虑?

分析:算法的设计主要需要考虑时间复杂度和空间复杂度,可以根据实际情况在两者之间作权衡。

解答:通常要考虑算法所需要的存储空间量和时间量,后者又涉及到四个方面,分别是程序运行时所需输入的数据总量、对源程序进行编译所需的时间、计算机执行每条指令所需的时间和程序中指令重复执行的次数。

【真题2】(吉林工业大学)在编制管理通讯录的程序时,什么样的数据结构合适?为什么?

分析:该题主要考查考生对于具体情况的分析能力。由于通讯录包含的内容很多,对于通讯录中不同的数据可能需要不同的数据结构来表现,所以选用的数据结构应能够满足用户操作时的需要。

解答:应从两方面进行讨论,若通讯录变动较少(如城市私人电话号码),主要用于查询,则以顺序存储较方便,既可顺序查找也可随机查找;若通讯录经常有增删操作,用链式存储结构较为合适,将每个人的情况作为一个元素(即一个结点存放一个人),设姓名作关键字,链表安排成有序表,这样可提高查询速度。

【真题3】(山东大学)若有100个学生,每个学生有学号、姓名、平均成绩,采用什么样的数据结构最方便?写出这些结构。

分析:考生需要能够用常见语言来描述数据结构,能够灵活应用考试所要求语言的数据结构的表示方法,包括整型、浮点型、数组、结构和指针的应用。虽然数据结构主要以考查算法思想为主,但是掌握考试所需要的语言是必需的。

解答:将学号、姓名、平均成绩看成一个记录(元素,含三个数据项),将100个这样的记录存于数组中。因一般无增删操作,故宜采用顺序存储。

```
typedef struct
{
    int num;           //学号
    char name[8];      //姓名
    float score;       //平均成绩
}node;
node student[100];
```

**【题型点睛】**

1. 在熟练掌握数据结构基本概念的基础上,考生需要具备能够根据具体情况设计数据结构,并且能够用语言表达出来的能力。其实,在学习数据结构的过程中,考生几乎会涉及每一种数据结构,并且课本中都已经给出具体语言的实现方法,所以只需要仔细分析题目要求,并结合课本中学习过的数据结构选取合适的方法,再根据具体情况做相应调整即可。由于此类问题的答案可能并不唯一,所以需要比较可行的数据结构并选择最优的,在回答问题的同时需要把自己的思想阐述清楚。

2. 解决实际问题时,往往需要采用数据类型来描述数据结构,这主要是基于以下三点考虑。

(1) 数据类型(Data Type)是一个值的集合和定义在这个值集上的一组操作的总称。解决现实问题就必须进行数据处理,而数据处理包括对数据进行查找、插入、删除、合并、排序、统计以及简单计算等操作过程。

(2) 数据类型是高级程序设计语言中的一个基本概念,它和数据结构的概念密切相关。一方面,在程序设计语言中,每一个数据都属于某种数据类型,数据类型明显或隐含地规定了数据的取值范围、存储方式以及允许进行的运算;可以认为,数据类型是在程序设计中已经实现了的数据结构。另一方面,在程序设计过程中,当需要引入某种新的数据结构时,总是借助编程语言所提供的数据类型来描述数据的存储结构。

(3) 用高级语言数据类型来描述数据结构,既避免了低级语言的复杂性,增加了可读性和简洁性,又有利于算法的实现。

3. 数据结构正逐步采用抽象数据类型描述,使其能更紧密地与面向对象的程序设计联系起来。抽象数据类型(Abstract Data Type, ADT)是指一个数学模型以及定义在该模型上的一组操作。

【即学即练】

【习题 1】(浙江大学)设计一个数据结构,用来表示某一银行储户的基本信息:账号、姓名、开户年月日、储蓄类型、存入累加数、利息和账面数据。