

图书基本信息

书名：<<程序员考试案例梳理、真题透解与强化训练>>

13位ISBN编号：9787121150289

10位ISBN编号：712115028X

出版时间：2012-1

出版时间：电子工业出版社

作者：爱拼人

页数：380

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

内容概要

本书紧扣《程序员考试大纲》的考核要求，深入研究了历次真题的命题风格和题型结构，依据考生在学习过程中所关注的3个要点：理考试重点、练历年真题和做模拟试卷进行梳理编写，致力于为读者在程序员案例分析方面提供务实有效的方法指导。

全书共10章。

对每一道真题都给出了详细的要点解析，其中不仅就试题进行了解题思路及步骤的讲解，而且对其考点及难点进行了扩展剖析。

解析细腻、重推理，针对性强，是本书的一大特色。

书籍目录

第1章 “程序员考试” 备考指南

- 1.1 考试目标
- 1.2 考试要求
- 1.3 考试形式
- 1.4 综合知识试卷
- 1.5 案例分析试卷
- 1.6 备考思路的调整

第2章 程序流程图填补题型

- 2.1 备考指南
- 2.2 真题透解
- 2.3 强化训练

第3章 C应用程序设计题型

- 3.1 备考指南
- 3.2 真题透解
- 3.3 强化训练

第4章 C程序设计改错题型

- 4.1 备考指南
- 4.2 真题透解
- 4.3 强化训练

第5章 数据结构与算法设计题型

- 5.1 备考指南
- 5.2 核心点
- 5.3 真题透解
- 5.4 强化训练

第6章 面向对象C++程序设计题型

第7章 面向对象JAVA程序设计题型

第8章 考前密押试卷

第9章 2010年下半年真题透解

第10章 2011年上半年真题透解

附录A 2011年下半年真题试卷

附录B 答题卡及答题纸示例

参考文献

章节摘录

版权页：插图：线性表顺序存储是最简单的存储方式，其特点是逻辑关系上相邻的两个元素在物理位置上也相邻。

通常使用一个足够大的数组，从数组的第1个元素开始，将线性表的结点依次存储在数组中。

线性表的第*i*个元素 $a[i]$ 的存储位置可以使用以下公式求得： $Loc(a_i) = Loc(a_1) + (i-1) \times L$ ，式中 $Loc(a_1)$ 是线性表的第1个元素 a_1 的存储位置，通常称为线性表的起始位置或基地址。

注意：在顺序表上实现基本的插入和删除操作时，由于C语言中数组的下标是从0开始的，因此在逻辑上所指的“第*k*个位置”实际上对应的是顺序表的“第*k*-1个位置”。

顺序存储方式的优点是能直接访问线性表中的任意结点。

其缺点有：表的大小固定，浪费大量的存储空间，不利于结点的增加和减少；执行线性表的插入和删除操作要移动其他元素，不够方便。

线性表链式存储是用链表来存储线性表的。

其特点是：每个链表都有一个头指针，整个链表的存取必须从头指针开始，头指针指向第1个数据元素的位置，最后的结点指针为空。

当链表为空时，头指针为空值。

当链表为非空时，头指针指向第1个结点。

链式存储的缺点是：由于要存储地址指针，因此浪费空间；直接访问结点不方便。

链式存储有单链表（线性链表）、循环链表和双向链表3种。

单链表从链表的第一个表元开始，将线性表的结点依次存储在链表的各表元中。

链表的每个表元除要存储线性表结点信息外，还要存储其后继结点的指针。

带头结点的单链表和不带头结点的单链表的区别主要体现在其结构和算法操作上。

在结构上，带头结点的单链表不管链表是否为空，均含有一个头结点，不带头结点的单链表不含头结点。

在操作上，带头结点的单链表的初始化为申请一个头结点。

无论插入或删除的位置是第一个结点还是其他结点，算法步骤都相同。

对于不带头结点的单链表，其算法步骤要分别考虑插入或删除的位置是第1个结点还是其他结点。

在单链表上实现线性表基本运算的初始化函数 $initlist(Slink *head)$ 用于创建一个头结点，由 $head$ 指向它，该结点的 $next$ 域为空， $data$ 域未设定任何值。

由于调用该函数时，指针 $head$ 在本函数中指向的内容发生了改变，为了返回改变的值，因此使用了应用型参数，其时间复杂度为 $O(1)$ 。

插入函数 $insert(Slink *head, int i, ElemType x)$ 的设计思想是：创建一个 $data$ 域值为 x 的新结点 p ，然后插入到 $head$ 所指向的单链表的第*i*个结点之前。

为保证插入正确有效，必须查找到指向第*i*个结点的前一个结点指针。

其主要的时间耗费在查找上，因而在长度为*n*的线性单链表中进行插入操作的时间复杂度为 $O(n)$ 。

。

编辑推荐

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>