

<<妙趣横生的算法>>

图书基本信息

书名：<<妙趣横生的算法>>

13位ISBN编号：9787302216018

10位ISBN编号：7302216010

出版时间：2010-4

出版时间：清华大学出版社

作者：杨峰

页数：377

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>

<<妙趣横生的算法>>

前言

程序 = 数据结构 + 算法——著名的计算机科学家沃斯 (Nikiklaus Wirth) 自从著名的计算机科学家沃斯将程序设计形象地用上面的公式表示出来后, 这条“黄金定律”便成为了人们学习程序设计, 进行程序开发的准则。

要想成为一名真正专业的程序设计人员, 基本的数据结构基础和常用的算法知识是必须掌握的。

脱离了这两点, 编写出来的程序一定不是健壮的好程序。

然而单纯地掌握了一些数据结构基础和常用的算法知识也是远远不够的。

空洞地掌握所谓的数据结构和算法等理论知识只是纸上谈兵, 这些知识必须要依托于一门程序设计语言才具有真正的生命力, 才能够转化为真实的程序代码, 才能真正地解决实际问题。

本书就是将数据结构基础和常用的算法知识与目前广泛应用、最具群众基础的C语言相结合而产生的。

。

本书的写作思想是理论与实践相结合, 以实践为核心, 以实例为主要内容。

首先, 本书总结归纳了数据结构基础、常用的排序查找算法和经典的算法思想, 提纲挈领地阐述了核心理论知识。

这样可以使没有系统学习过或者不熟悉数据结构和算法等知识的读者对这部分知识有一个基本的了解, 并掌握基本的数据结构知识和常用而经典的算法思想, 以便更加深入地学习本书的其他内容。

其次, 本书列举了大量的编程实例, 这些题目都按照知识体系进行了内容上的划分。

本书列举的这些编程实例都是一些比较灵活有趣的题目, 有些题目渗透了巧妙的算法思想, 有些题目则必须借助特殊的数据结构才能更加容易解答。

通过这些题目的训练, 可以使读者开阔眼界, 启迪思维, 提高编程的兴趣。

最重要的是能够提高读者算法设计的本领, 提高读者灵活应用各种数据结构的本领, 提高读者编写程序解决实际问题能力。

本书有何特点1. 结构清晰, 知识全面本书分为两部分。

第1部分是基础知识介绍, 主要介绍数据结构的基础知识和一些常用的算法思想。

这部分内容为核心理论知识, 可以帮助读者学习和回顾数据结构和算法的知识, 使读者在理论水平上有所提高, 从而能够更加顺利地深入学习后续内容。

第2部分主要是编程实例的介绍, 通过一些非常有趣的编程实例使读者开阔眼界, 发散思维, 提高算法设计本领, 提高灵活应用各种数据结构的本领, 提高读者编写程序解决实际问题能力。

<<妙趣横生的算法>>

内容概要

本书理论与实践相结合，旨在帮助读者理解算法，并提高C语言编程能力，培养读者的编程兴趣，并巩固已有的C语言知识。

全书分为2个部分共10章，内容涵盖了编程必备的基础知识（如数据结构、常用算法等），编程实例介绍，常见算法和数据结构面试题等。

本书最大的特色在于实例丰富，题材新颖有趣，实用性强，理论寓于实践之中。

通过本书的学习，可以使读者开阔眼界，提高编程的兴趣，提高读者的编程能力和应试能力。

本书附带1张光盘，内容为本书源代码和作者为本书录制的5.5小时多媒体教学视频。

本书可作为算法入门人员的教程，也可以作为学习过C语言程序设计的人士继续深造的理想读物，也可作为具有一定经验的程序设计人员巩固和提高编程水平，查阅相关算法实现和数据结构知识的参考资料，同时也为那些准备参加与算法和数据结构相关的面试的读者提供一些有益的帮助。

<<妙趣横生的算法>>

书籍目录

第1部分 基础篇	第1章 数据结构基础	1.1 什么是数据结构	1.2 顺序表	1.2.1 顺序表的定义
	1.2.2 向顺序表中插入元素	1.2.3 从顺序表中删除元素	1.2.4 实例与分析	
	1.3 链表	1.3.1 创建一个链表	1.3.2 向链表中插入结点	1.3.3 从链表中删除结点
	1.3.4 销毁一个链表	1.3.5 实例与分析	1.4 栈	1.4.1 栈的定义
栈	1.4.3 入栈操作	1.4.4 出栈操作	1.4.5 栈的其他操作	1.4.2 创建一个
	1.4.6 实例与分析			
1.5 队列	1.5.1 队列的定义	1.5.2 创建一个队列	1.5.3 入队列操作	1.5.4 出队
列操作	1.5.5 销毁一个队列	1.5.6 循环队列的概念	1.5.7 循环队列的实现	
1.5.8 实例与分析	1.6 树结构	1.6.1 树的概念	1.6.2 树结构的计算机存储形式	
1.6.3 二叉树的定义	1.6.4 二叉树的遍历	1.6.5 创建二叉树	1.6.6 实例与分析	
1.7 图结构	1.7.1 图的概念	1.7.2 图的存储形式	1.7.3 邻接表的定义	1.7.4 图
的创建	1.7.5 图的遍历(1)——深度优先搜索	1.7.6 图的遍历(2)——广度优先搜索		
1.7.7 实例与分析	第2章 常用的查找与排序方法	2.1 顺序查找	2.2 折半查找	2.3 排
序的概述	2.4 直接插入排序	2.5 选择排序	2.6 冒泡排序	2.7 希尔排序
2.8 快速排	第3章 常用的算法思想	3.1 什么是算法	3.2 算法的分类表示及测评	3.2.1 算法的分
类	3.2.2 算法的表示	3.2.3 算法性能的测评	3.3 穷举法思想	3.3.1 基本概念
3.3.2 寻找给定区间的素数	3.3.3 TOM的借书方案	3.4 递归与分治思想	3.4.1 基本	
概念	3.4.2 计算整数的划分数	3.4.3 递归的折半查找算法	3.5 贪心算法思想	
3.5.1 基本概念	3.5.2 最优装船问题	3.6 回溯法	3.6.1 基本概念	3.6.2 四皇后问
题求解	3.7 数值概率算法	3.7.1 基本概念	3.7.2 计算定积分	第2部分 编程实例解析
第4章 编程基本功	第5章 数学趣题(一)	第6章 数学趣题(二)	第7章 数据结构趣题	第8章 数
值计算问题	第9章 综合题	第10章 算法设计与数据结构面试题精粹		

<<妙趣横生的算法>>

章节摘录

插图：其实阶乘的数学定义可以用递归函数来简单地描述：这样的函数称为递归函数，因为该函数本身直接或间接地调用了该函数本身。

基于阶乘的递归函数的描述，就不难设计出计算 n 的阶乘 $n!$

的递归算法。

可以看出，使用递归算法解决阶乘问题形式上更加简洁，更易于人们理解。

在设计递归算法时要注意以下几点。

(1) 每个递归函数都必须有一个非递归定义的初始值，作为递归结束标志，或递归结束的出口。

就像实例3.5所描述的递归算法中的if ($n=0$) ret 0; 如果一个递归算法中没有这个非递归定义的初始值，那么该递归调用是无法计算出具体值的（或无法得到结果），同时该递归调用也无法结束。

(2) 在设计递归算法时，要解决的问题需具有递归性。

例如要计算， 2 的阶乘 $n!$

， $n!$

的定义本身具有递归性。

这种所谓的递归性实际上就是一种反复调用自身过程的特性。

(3) 虽然采用递归算法解决问题，特别是一些复杂问题，更加方便且容易实现，但是递归方法的运行效率较低，时间和空间复杂度都比较高，因此对于一些对时间和空间要求较高的程序，建议使用非递归算法设计。

在实际的算法设计中，递归与分治如同一对兄弟，经常结合在一起使用。

这是因为，由分治的方法产生的子问题往往都是原问题的更小规模。

反复使用分治的手段，可使子问题与原问题类型一致，但规模不断缩小，最终使子问题比较容易求解。

既然子问题与原问题的类型一致，这就具有了所谓的递归性，因此可以使用递归的方法去解决原问题的算法去解决同类型的子问题。

在第2章中介绍的折半查找算法只是单纯地使用了分治的策略，在下面的实例分析中将使用递归与分治思想相结合的方法进行折半查找算法的设计。

<<妙趣横生的算法>>

编辑推荐

《妙趣横生的算法(C语言实现)》：5.5小时教学视频、86个趣味算法题、61个算法面试题，一学就会！

帮您开阔眼界，培养编程兴趣，提高编程能力，增强求职的竞争力！

特别提示《妙趣横生的算法(C语言实现)》配套多媒体教学视频和涉及的实例代码收录于《妙趣横生的算法(C语言实现)》配书光盘中。

另外，《妙趣横生的算法(C语言实现)》适合作为相关学校的教材使用。

为了方便老师授课，《妙趣横生的算法(C语言实现)》专门配备了相应的教学PPT。

《妙趣横生的算法(C语言实现)》内容生动有趣，寓教于乐，旨在帮您开阔眼界，培养编程兴趣，提高编程能力。

增强求职的竞争力。

如果您想在程序设计之路上走得更远，请翻开《妙趣横生的算法(C语言实现)》，仔细研读吧，它将助您一臂之力。

《妙趣横生的算法(C语言实现)》特色 提供了5.5小时多媒体教学视频，学习起来比较直观。

提供了24个数据结构和算法的基础知识实例，便于读者入门。

提供了86个经典、有趣、贴近生活、实用性强的算法实例。

提供了61个算法及数据结构的面试题。

增强求职者的竞争力。

内容梯度科学，既适合入门，也适合进一步提高和研究。

书中实例用C语言实现，便于读者验证及加深对C语言的理解。

既涵盖基本理论，又包含大量实例，寓理论于实践之中。

讲解由浅入深，通俗易懂，将复杂问题简单化，读者可以轻松掌握。

<<妙趣横生的算法>>

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>