

<<数据结构和程序设计>>

图书基本信息

书名：<<数据结构和程序设计>>

13位ISBN编号：9787030362230

10位ISBN编号：7030362233

出版时间：2013-1

出版时间：Robert Kruse、C.L.Tondo、Bruce Leung 科学出版社 (2013-01出版)

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>

<<数据结构和程序设计>>

作者简介

作者:(美)克鲁斯 著作

书籍目录

PREFACE
 Synopsis
 Changes in the Second Edition
 Course Structure
 Book Production
 Acknowledgments
 CHAPTER 1 Programming Principles
 1.1 Introduction
 1.2 The Game of Life
 1.2.1 Rules for the Game of Life
 1.2.2 Examples
 1.2.3 The Solution
 1.2.4 Life: The Main Program
 1.3 Programming Style
 1.3.1 Names
 1.3.2 Documentation and Format
 1.3.3 Refinement and Modularity
 1.4 Coding, Testing, and Further Refinement
 1.4.1 Stubs
 1.4.2 Counting Neighbors
 1.4.3 Input and Output
 1.4.4 Drivers
 1.4.5 Program Tracing
 1.4.6 Principles of Program Testing
 Pointers and Pitfalls
 Review Questions
 References for Further Study
 C Programming Principles
 The Game of Life
 CHAPTER 2 Introduction to Software Engineering
 2.1 Program Maintenance
 2.1.1 Review of the Life Program
 2.1.2 A Fresh Start and a New Method for Life
 2.2 Algorithm Development: A Second Version of Life
 2.2.1 Lists: Specifications for a Data Structure
 2.2.2 The Main Program
 2.2.3 Information Hiding
 2.2.4 Refinement: Development of the Subprograms
 2.2.5 Verification of Algorithms
 2.3 Coding
 2.3.1 The List Functions
 2.3.2 Error Processing
 2.3.3 Demonstration and Testing
 2.4 Coding the Life Functions
 2.5 Program Analysis and Comparison
 2.6 Conclusions and Preview
 2.6.1 The Game of Life
 2.6.2 Program Design
 2.6.3 C Pointers and Pitfalls
 Review Questions
 References for Further Study
 CHAPTER 3 Stacks and Recursion
 3.1 Stacks
 3.1.1 Introduction
 3.1.2 First Example: Reversing a Line
 3.1.3 Information Hiding
 3.1.4 Specifications for a Stack
 3.1.5 Implementation of Stacks
 3.1.6 Linked Stacks
 3.2 Introduction to Recursion
 3.2.1 Stack Frames for Subprograms
 3.2.2 Tree of Subprogram Calls
 3.2.3 Factorials: A Recursive Definition
 3.2.4 Divide and Conquer: The Towers of Hanoi
 3.3 Backtracking: Postponing the Work
 3.3.1 Solving the Eight-Queens Puzzle
 3.3.2 Example: Four Queens
 3.3.3 Backtracking
 3.3.4 Refinement: Choosing the Data Structures
 3.3.5 Analysis of Backtracking
 3.4 Principles of Recursion
 3.4.1 Designing Recursive Algorithms
 3.4.2 How Recursion Works
 3.4.3 Tail Recursion
 3.4.4 When Not to Use Recursion
 3.4.5 Guidelines and Conclusions
 Pointers and Pitfalls
 Review Questions
 References for Further Study
 CHAPTER 4 Queues and Linked Lists
 4.1 Definitions
 4.2 Implementations of Queues
 4.3 Circular Queues in C
 4.4 Application of Queues: Simulation
 4.4.1 Introduction
 4.4.2 Simulation of an Airport
 4.4.3 The Main Program
 4.4.4 Steps of the Simulation
 4.4.5 Pseudo-Random Numbers
 4.4.6 Sample Results
 4.5 Pointers and Linked Lists
 4.5.1 Introduction and Survey
 4.5.2 Pointers and Dynamic Memory in C
 4.5.3 The Basics of Linked Lists
 4.6 Linked Queues
 4.7 Application: Polynomial Arithmetic
 4.7.1 Purpose of the Project
 4.7.2 The Main Program
 4.7.3 Data Structures and Their Implementation
 4.7.4 Reading and Writing Polynomials
 4.7.5 Addition of Polynomials
 4.7.6 Completing the Project
 4.8 Abstract Data Types and Their Implementations
 4.8.1 Introduction
 4.8.2 General Definitions
 4.8.3 Refinement of Data Specification
 Pointers and Pitfalls
 Review Questions
 References for Further Study
 CHAPTER 5 General Lists
 5.1 List Specifications
 5.2 Implementation of Lists
 5.2.1 Contiguous Implementation
 5.2.2 Simply Linked Implementation
 5.2.3 Variation: Keeping the Current Position
 5.2.4 Doubly Linked Lists
 5.2.5 Comparison of Implementations
 5.3 Strings
 5.4 Application: A Text Editor
 5.4.1 Specifications
 5.4.2 Implementation
 5.5 Linked Lists in Arrays
 5.6 Generating Permutations
 Pointers and Pitfalls
 Review Questions
 References for Further Study
 CHAPTER 6 Searching
 6.1 Searching: Introduction and Notation
 6.2 Sequential Search
 6.3 Coatrooms: A Project
 6.3.1 Introduction and Specification
 6.3.2 Demonstration and Testing Programs
 6.4 Binary Search
 6.4.1 Algorithm Development
 6.4.2 The Forgetful Version
 6.4.3 Recognizing Equality
 6.5 Comparison Trees
 6.5.1 Analysis for $n=106$
 6.5.2 Generalization
 6.5.3 Comparison of Methods
 6.5.4 A General Relationship
 6.6 Lower Bounds
 6.7 Asymptotics
 6.7.1 Introduction
 6.7.2 The Big-O Notation
 6.7.3 Imprecision of the Big-O Notation
 6.7.4 Ordering of Common Functions
 Pointers and Pitfalls
 Review Questions
 References for Further Study
 CHAPTER 7 Sorting
 7.1 Introduction and Notation
 7.2 Insertion Sort
 7.2.1 Ordered Lists
 7.2.2 Sorting by Insertion
 7.2.3 Linked Version
 7.2.4 Analysis
 7.3 Selection Sort
 7.3.1 The Algorithm
 7.3.2 Contiguous Implementation
 7.3.3 Analysis
 7.3.4 Comparisons
 7.4 Shell Sort
 7.5 Lower Bounds
 7.6 Divide-and-Conquer Sorting
 7.6.1 The Main Ideas
 7.6.2 An Example
 7.7 Mergesort for Linked Lists
 7.7.1 The Functions
 7.7.2 Analysis of Mergesort
 7.8 Quicksort for Contiguous Lists
 7.8.1 The Main Function
 7.8.2 Partitioning the List
 7.8.3 Analysis of Quicksort
 7.8.4 Average-Case Analysis of Quicksort
 7.8.5 Comparison with Mergesort
 7.9 Heaps and Heapsort
 7.9.1 Two-Way Trees as Lists
 7.9.2

Heapsort7.9.3 Analysis of Heapsort7.9.4 Priority Queues7.10 Review:Comparison of MethodsPointers and PitfallsReview QuestionsReferences for Further StudyCHAPTER 8 Tables and Information Retrieval8.1 Introduction:Breaking the Ignbarrier8.2 Rectangular Arrays8.3 Tables of Various Shapes8.3.1 Triangular Tables8.3.2 Jagged Tables8.3.3 Inverted Tables8.4 Tables:A New Abstract Data Type8.5 Application:Radix Sort8.5.1 The Idea8.5.2 Implementation8.5.3 Analvsis8.6 Hashing8.6.1 Sparse Tables8.6.2 Choosing a Hash Function8.6.3 Collision Resolution with Open Addressing8.6.4 Conision Resolution by Chaining8.7 Analysis of Hashing8.8 Conclusions:Comparison of Methods8.9 Application:The Life Game Revisited8.9.1 Choice of Algorithm8.9.2 Specification of Data Structures8.9.3 The Main Program8.9.4 FunctionsPOinters and PitfallsReview QuestionsReferences for Further StudyCHAPTER 9 Binary Trees9.1 Introduction to Binary Trees9.1.1 Definitions9.1.2 Traversal of Binary Trees9.1.3 Linked Implementation of Billarv Trees9.2 BinarV Search Trees9.2.1 ordered Lists and Implementations9.2.2 Treesearch9.2.3 Insertion into a BinarV Search Tree9.2.4 Treeseort9.2.5 Deletion from a Binarv SearCh Tree9.3 Building a Binary Search Tree9.3.1 Getting Started9.3.2 Declarations and the Main Function9.3.3 Inserting a Node9.3.4 Finishing the Task9.3.5 Evaluation9.3.6 RandOm Search Trees and Optimalit9.4 Height Balance:AVL Trees9.4.1 Definition9.4.2 Insertion of a Node9.4.3 Deletion of a Node9.4.4 The Height of an AVL Tree9.5 Splav Trees:A Self-Adj using Data Stillcture9.5.1 Introduction9.5.2 Splayillg Steps9.5.3 Splaying Algorithm9.5.4 AmOrtized Algorithm AnalVsis:Introduction9.5.5 Amortized Analysis of SplayingPointers and PitfallsReview QuestionsReferences for Further StudyCHAPTER 10 Multiway Trees10.1 Orchards,Trees,and Binary Trees10.1.1 on the Classification of Species10.1.2 Ordered Trees10.1.3 Forests and Orchards10.1.4 The Formal Correspondence10.1.5 Rotations10.1.6 Summary10.2 Lexicographic Search Trees:Tries10.2.1 Tries10.2.2 Searching for a Key10.2.3 C Algorithm10.2.4 Insertion into a Trie10.2.5 Deletion from a Trie10.2.6 Assessment of Tries10.3 External Searchiring:B-Trees10.3.1 Access Time10.3.2 Multiway Search Trees10.3.3 Balanced Multiway Trees10.3.4 Insertion into a B-tree10.3.5 C Algorithms:Searching and Insertion10.3.6 Deletion from a B-tree10.4 Red-Black Trees10.4.1 Introduction10.4.2 Definition and Analysis10.4.3 Insertion10.4.4 C Insertion10.5 Tree-Structured Programs:Look-Ahead in Games10.5.1 Game Trees10.5.2 The Minimax Method10.5.3 Algorithm Development10.5.4 RefinementPointers and PitfallsReview QuestionsReferences for Further StudyCHAPTER 11 Graphs11.1 Mathematical Background11.1.1 Defimtions and Examples11.1.2 Undirected Graphs11.1.3 Directed Graphs11.2 Computer Representation11.3 Graph Traversal11.3.1 Methods11.3.2 Depth-First Algorithm11.3.3 Breadth-First Algorithm11.4 T0p010gical Sorting11.4.1 The Problem11.4.2 Depth-First Algorithm11.4.3 Br.eadth.First AlgOrithm11.5 A Greedy Algorithm:Shortest Paths11.6 Graphs as Data StructuresPOinterS and PitfallsReview QuestionsReferences for Funher StudyCHAPTER 12 Case Study:The Polish Notation12.1 The PrOblem12.1.1 The Quadratic Fomula12.2 The Idea12.2.1 Expression Trees12.2.2 Polish Notation12.2.3 C Method12.3 Evaluation of Polish Expressions12.3.1 Evaluation of an Expression in Prefix Form12.3.2 C Conventions12.3.3 C Function for Prefix Evaluation12.3.4 Evaluation of Postfix Expressions12.3.5 PrOof of the Program:Counting:Stack Entries12.3.6 Recursive Evaluation of Postfix Expressions12.4 Translation from Infix Form to Polish Form12.5 An Interactive Expression Evaluator12.5.1 Overall Structure12.5.2 Representation of the Data12.5.3 Initialization and Auxiliary Tasks12.5.4 Translation of the Expression12.5.5 Evaluating the Expression12.5.6 Graphing the ExpressionReferences for Further StudyAPPENDIX A Mathematical MethodsA.1 Sums of Powers of IntegersA.2 LogarithmsA.2.1 Definition of LogarithmsA.2.2 Simple PropertiesA.2.3 Choice of BaseA.2.4 Natural LogarithmsA.2.5 NotationA.2.6 Change of BaseA.2.7 Logarithmic GraphsA.2.8 Harmonic NumbersA.3 Permutations,Combinations,FactorialsA.3.1 PermutationsA.3.2 CombinationsA.3.3 FactorialsA.4 Fibonacci NumbersA.5 Catalan NumbersA.5.1 The Main ResultA.5.2 The Proof by one-to-one CorrespondencesA.5.3 HistoryA.5.4 Numerical ResultsReferences for Further StudyAPPENDIX B Removal of RecursionB.1 General Methods for Removing RecursionB.1.1 Preliminary AssumptionsB.1.2 General RulesB.1.3 Indirect RecursionB.1.4 Towers of HanoiB.1.5 Further SimplificationsB.2 Recursion Removal by F0ldingB.2.1 Program SchemataB.2.2 Proof of the TransformatorIB.2.3 Towers of Hanoi:The Final VersionB.3 Nonrecursive QuicksortB.4 Stackless Recursion Removaal:MergesortB.5 Threaded Binary TreesB.5.1 IntroductionB.5.2 ThreadsB.5.3 Inorder and

Preorder TraversalB.5.4 Insertion in a Threaded TreeB.5.5 Postorder TraversalReferences for Further StudyAPPENDIX C An Introduction to CC.1 IntrOductionC.1.1 Overview of a C ProgramC.2 C ElementsC.2.1 Reserved WordsC.2.2 ConstantsC.3 Types and DeclarationsC.3.1 Basic TypesC.3.2 ArravsC.3.3 EnumerationsC.3.4 StructuresC.3.5 UnionsC.3.6 Type Defillitions with typedefC.4 0permorsC.5 Control Flow StatementsC.5.1 If-ElseC.5.2 SwitchC.5.3 LoopsC.5.4 Break and ContinueC.5.5 GotoC.6 PointersC.6.1 Pointer to a Simple VariableC.6.2 Pointer to an ArrayC.6.3 Array of PointersC.6.4 Pointer to StructuresC.7 FunctionsC.7.1 Arguments to Functions:Call by ValueC.7.2 Arguments to Functions:Call by ReferenceC.7.3 Function Prototypes and Include FilesC.8 Pointers and FunctionsC.8.1 Pointer to a FunctionC.8.2 Functios that Ream a PointerC.8.3 Pointer to a Pointer as an ArgumentReferences for Further StudyINDEX

<<数据结构和程序设计>>

编辑推荐

《数据结构和程序设计——应用C语言(第2版)》以C++为描述语言，系统介绍数据结构的有关内容及程序设计方法。

每章都是先引入实例，然后结合实例讲解知识点，每章后都附有指针和陷阱的内容，还配有复习思考题，以检验读者的学习效果和培养读者的程序设计能力。

此外，每章后还有深入学习本章知识点的阅读参考资料，有利于读者加深对本章知识点的理解。

全书既注重原理又重视实践，内容叙述详细，并配有大量的实例和习题。

书中所有算法均在计算机上运行通过，且程序中做了较详细的注解，有利于读者理解算法的实质和编程思想。

本书由克鲁斯等著。

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>