

图书基本信息

书名：<<数据结构与程序设计(C语言描述)第2版--英文>>

13位ISBN编号：9787302029434

10位ISBN编号：7302029431

出版时间：1998-07

出版时间：清华大学出版社

作者：克鲁泽

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

### 内容简介

本书强调问题的描述和程序的分析、设计、测试、验证以及程序正确性，将深思熟虑的开发的思路融于具体的程序设计之中。

书中介绍了程序设计原理和软件工程知识以及如何将这些原理和知识运用于程序（算法）设计，使用大量实例介绍了几种主要数据结构：栈、表、树、图及主要算法如递归、查找、排序、检索等，在介绍过程中注重运用程序设计的先进思想和软件工程的解决方法。

书中给出的实例很有代表性，能覆盖大量程序设计原理。

数据抽象是贯穿全书的思想。

本书还包括了如倾斜树、红黑树等很多新的内容。

最后，以一个完整的实例详细讨论了用递归、树、栈等手段解决具体问题。

附录中介绍了一些常用数学算法及如何消除递归，并对C语言作了简单小结。

本书适合于计算机专业本科生作为学习数据结构的教材或辅助教材。

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 Ceding 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
- Contents
- 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  $n = 10$

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 Insertin.

7.2.3 Linked Version

7.2.4 Analysis

7.3 SelectionSort

7.3.1 TheAlgorithm

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 TheMainIdeas

7.6.2 An Example

7.7 Mergesort for Linked Lists

7.7.1 TheFunctions

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 Heapsort

7.9.3 Analysis of Heapsort

7.9.4 PriorityQueues

7.10 Review: Comparison of Methods

Pointers and Pitfalls

Review Questions

References for Further Study

CHAPTER 8

Tables and Information Retrieval

8.1 Introduction:

Breaking the  $\lg n$  Barrier

8.2 Rectangular Arrays

8.3 Tables of Various Shapes

8.3.1 Triangular Tables

8.3.2 Jagged Tables

8.3.3 Inverted Tables

8.4 Tables: A New Abstract Data Type

8.5 Application: Radix Sort

8.5.1 The Idea

8.5.2 Implementation

8.5.3 Analysis

8.6 Hashing

8.6.1 Sparse Tables

8.6.2 Choosing a Hash Function

8.6.3 Collision Resolution with

Open Addressing

8.6.4 Collision Resolution by Chaining

8.7 Analysis of Hashing

8.8 Conclusions:

Comparison of Methods

8.9 Application:

The Life Game Revisited

8.9.1 Choice of Algorithm

8.9.2 Specification of Data Structures

8.9.3 The Main Program

8.9.4 Functions

Pointers and Pitfalls

Review Questions

References for Further Study

## CHAPTER 9

### Binary Trees

#### 9.1 Introduction to Binary Trees

##### 9.1.1 Definitions

##### 9.1.2 Traversal of Binary Trees

##### 9.1.3 Linked Implementation of Binary Trees

#### 9.2 Binary Search Trees

##### 9.2.1 Ordered Lists and Implementations

##### 9.2.2 Tree Search

##### 9.2.3 Insertion into a Binary Search Tree

##### 9.2.4 Tree Sort

##### 9.2.5 Deletion from a Binary Search Tree

#### 9.3 Building a Binary Search Tree

##### 9.3.1 Getting Started

##### 9.3.2 Declarations and the Main Function

##### 9.3.3 Inserting a Node

##### 9.3.4 Finishing the Task

##### 9.3.5 Evaluation

##### 9.3.6 Random Search Trees and Optimality

#### 9.4 Height Balance: AVL Trees

##### 9.4.1 Definition

##### 9.4.2 Insertion of a Node

##### 9.4.3 Deletion of a Node

##### 9.4.4 The Height of an AVL Tree

#### 9.5 Splay Trees:

##### A Self-Adjusting Data Structure

##### 9.5.1 Introduction

##### 9.5.2 Splaying Steps

##### 9.5.3 Splaying Algorithm

##### 9.5.4 Amortized Algorithm Analysis: Introduction

##### 9.5.5 Amortized Analysis of Splaying Pointers and Pitfalls

##### Review Questions

##### References for Further Study

## CHAPTER 10

### Multiway Trees

#### 10.1 Orchards, Trees, and Binary Trees

##### 10.1.1 On the Classification of Trees

##### 10.1.2 Ordered Trees

##### 10.1.3 Forests and Orchards

##### 10.1.4 The Formal Correspondence

##### 10.1.5 Rotations

##### 10.1.6 Summary

#### 10.2 Lexicographic Search Trees: Tries



- 10.2.1 Tries
- 10.2.2 Searching for a Key
- 10.2.3 CAlgorithm
- 10.2.4 Insertion into a Trie
- 10.2.5 Deletion from a Trie
- 10.2.6 AssessmentofTries
- 10.3 External Searching: B-Trees
  - 10.3.1 AccessTime
  - 10.3.2 Multiway Search Trees
  - 10.3.3 Balanced Multiway Trees
  - 10.3.4 Insertion into a B-tree
  - 10.3.5 CAlgorithms:  
Searching and Insertion
  - 10.3.6 Deletion from a B-tree
- 10.4 Red-Black Trees
  - 10.4.1 Introduction
  - 10.4.2 Definition and Analysis
  - 10.4.3 Insertion
  - 10.4.4 C Insertion
- 10.5 Tree-Structured Programs:  
Look-Ahead in Games
  - 10.5.1 GameTrees
  - 10.5.2 The Minimax Method
  - 10.5.3 Algorithm Development
  - 10.5.4 Refinement
- Pointers and Pitfalls
- Review Questions
- References for Further Study
- CHAPTER 11
- Graphs
  - 11.1 Mathematical Background
    - 11.1.1 Definitions and Examples
    - 11.1.2 Undirected Graphs
    - 11.1.3 Directed Graphs
  - 11.2 Computer Representation
  - 11.3 Graph Traversal
    - 11.3.1 Methods
    - 11.3.2 Depth-First Algorithm
    - 11.3.3 Breadth-First Algorithm
  - 11.4 Topological Sorting
    - 11.4.1 TheProblem
    - 11.4.2 Depth-First Algorithm
    - 11.4.3 Breadth-First Algorithm
  - 11.5 A Greedy Algorithm:  
Shortest Paths
  - 11.6 Graphs as Data Structures
- Pointers and Pitfalls

Review Questions

References for Further Study

## CHAPTER 12

Case Study: The Polish Notation

12.1 The Problem

12.1.1 The Quadratic Formula

12.2 The Idea

12.2.1 Expression Trees

12.2.2 Polish Notation

12.2.3 C Method

12.3 Evaluation of Polish Expressions

12.3.1 Evaluation of an Expression in  
Prefix Form

12.3.2 C Conventions

12.3.3 C Function for Prefix Evaluation

12.3.4 Evaluation of Postfix Expressions

12.3.5 Proof of the Program:

Counting Stack Entries

12.3.6 Recursive Evaluation of  
Postfix Expressions

12.4 Translation from Infix Form to Polish  
Form

12.5 An Interactive Expression

Evaluator

12.5.1 Overall Structure

12.5.2 Representation of the Data

12.5.3 Initialization and Auxiliary Tasks

12.5.4 Translation of the Expression

12.5.5 Evaluating the Expression

12.5.6 Graphing the Expression

References for Further Study

## APPENDIX A

Mathematical Methods

A.1 Sums of Powers of Integers

A.2 Logarithms

A.2.1 Definition of Logarithms

A.2.2 Simple Properties

A.2.3 Choice of Base

A.2.4 Natural Logarithms

A.2.5 Notation

A.2.6 Change of Base

A.2.7 Logarithmic Graphs

A.2.8 Harmonic Numbers

A.3 Permutations, Combinations,  
Factorials

A.3.1 Permutations

A.3.2 Combinations

A.3.3 Factorials

A.4 Fibonacci Numbers

A.5 Catalan Numbers

A.5.1 ThevMain Result

A.5.2 The Proof by One-to-One

Correspondences

A.5.3 History

A.5.4 Numerical Results

References for Further Study

APPENDIX B

Removal of Recursion

B.1 General Methods for Removing  
Recursion

B.1.1 Preliminary Assumptions

B.1.2 GeneralRules

B.1.3 Indirect Recursion

B.1.4 Towers of Hanoi

B.1.5 Further Simplifications

B.2 Recursion Removal by Folding

B.2.1 Program Schemata

B.2.2 Proof of the Transformation

B.2.3 Towers of Hanoi:

The Final Version

B.3 Nonrecursive Quicksort

B.4 Stackless Recursion Removal:  
Mergesort

B.5 Threaded Binary Trees

B.5.1 Introduction

B.5.2 Threads

8.5.3 Inorder and Preorder Traversal

B.5.4 Insertion in a Threaded Tree

B.5.5 Postorder Traversal

References for Further Study

APPENDIX C

An Introduction to C

C.1 Introduction

C.1.1 Overview of a C Program

C.2 CElements

C.2.1 Reserved Words

C.2.2 Constants

C.3 Types and Declarations

C.3.1 Basic Types

C.3.2 Arrays

C.3-3 Enumerations

C.3 4 Structures

C.3.5 Unions

C.3.6 Type Definitions with typedef

C.4 Operators

C.5 Control Flow Statements

C.5.1 K-Else

C.5.2 Switch

C.5.3 Loops

C.5.4 Break and Continue

C.5.5 Goto

C.6 Pointers

C.6.1 Pointer to a Simple Variable

C.6.2 Pointer to an Array

C.6.3 Array of Pointers

C.6.4 Pointer to Structures

C.7 Functions

C.7.1 Arguments to Functions:

CallbyValue

C.7.2 Arguments to FUNctions:

Call by Reference

C.7.3 Function Prototypes and Include  
Files

C.8 Pointers and Functions

C.8.1 Pointer to a FuCnction

C.8.2 Functions that Return a Pointe

C.8.3 Pointer to a Pointer as an

Argument

References for Further Study

INDEX

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>