

# DATA STRUCTURES

---

A Pseudocode  
Approach  
with C++



Suranaree University of Technology



31051000657402

Richard F. Gilberg • Behrouz A. Forouzan

# Contents

## 1 Introduction 1

### 1-1 Pseudocode 2

- Algorithm Header 2
- Purpose, Conditions, and Return 3
- Statement Numbers 4
- Variables 4
- Algorithm Analysis 5
- Statement Constructs 5
- Pseudocode Example 6

### 1-2 The Abstract Data Type 7

- Atomic and Composite Data 8
- Data Structure 8
- Abstract Data Type 9

### 1-3 A Model for an Abstract Data Type 11

- ADT Operations 11
- ADT Data Structure 12
- ADT Class Templates 13

### 1-4 Algorithm Efficiency 14

- Linear Loops 15
- Logarithmic Loops 16
- Nested Loops 17
- Big-O Notation 19
- Standard Measures of Efficiency 20
- Big-O Analysis Examples 21

### 1-5 Summary 24

### 1-6 Practice Sets 25

- Exercises 25
- Problems 27
- Projects 27

## 2 Searching 29

### 2-1 List Searches 30

- Sequential Search 30

Variations on Sequential Searches 33

Binary Search 37

Binary Search Algorithm 40

Analyzing Search Algorithms 41

### 2-2 C++ Search Algorithms 43

Sequential Search in C++ 43

Binary Search in C++ 44

Search Example 46

### 2-3 Hashed List Searches 49

Basic Concepts 50

Hashing Methods 52

Hashing Algorithm 57

### 2-4 Collision Resolution 59

Open Addressing 61

Linked List Resolution 65

Bucket Hashing 66

Combination Approaches 67

Hash List Example 68

### 2-5 Summary 72

### 2-6 Practice Sets 74

Exercises 74

Problems 75

Projects 75

## 3 Linked Lists 77

### 3-1 Linear List Concepts 78

Insertion 79

Deletion 80

Retrieval 80

Traversal 80

### 3-2 Linked List Concepts 81

Nodes 81

Linked List Data Structure 82

Pointers to Linked Lists 84

**3-3 Linked List Algorithms 84**

- Create List 84
- Insert Node 85
- Delete Node 90
- Search List 93
- Unordered List Search 96
- Retrieve Node 97
- Empty List 97
- Full List 98
- List Count 98
- Traverse List 99
- Destroy List 101

**3-4 Processing a Linked List 102**

- Add Node 104
- Remove Node 105
- Print List 106
- Testing Insert and Delete Logic 107

**3-5 List Applications 108**

- Append Lists 108
- Array of Lists 110

**3-6 Complex Linked List Structures 113**

- Circularly Linked Lists 113
- Doubly Linked Lists 113
- Multilinked Lists 119
- Multilinked List Insert 121
- Multilinked List Delete 122

**3-7 Building a Linked List—C++ Implementation 122**

- Data Structure 122
- Application Functions 123

**3-8 List Abstract Data Type—Linked List Implementation 129**

- List ADT Declaration 131

**3-9 Summary 143****3-10 Practice Sets 144**

- Exercises 144
- Problems 147
- Projects 148

**4 Stacks 156****4-1 Basic Stack Operations 157**

- Push 157
- Pop 157
- Stack Top 158

**4-2 Stack Linked List Implementation 160**

- Data Structure 160
- Stack Algorithms 161

**4-3 Stack Applications 168**

- Reversing Data 169
- Reverse a List 169
- Convert Decimal to Binary 170
- Parsing 171
- Postponement 173
- Backtracking 181

**4-4 Eight Queens Problem—C++ Implementation 188**

- Main Line Logic 189
- Get Board Size 190

**4-5 Stack Abstract Data Type Implementation 195**

- Data Structure 195
- Stack ADT Implementation 196

**4-6 Stack ADT—Array Implementation 202**

- Array Data Structure 203
- Create Stack Array 204
- Push Stack Array 205
- Pop Stack Array 206
- Stack Top Array 207
- Empty Stack Array 208
- Full Stack Array 208
- Stack Count Array 208
- Destroy Stack Array 209

**4-7 Summary 209****4-8 Practice Sets 210**

- Exercises 210
- Problems 211
- Projects 213

**5 Queues 217****5-1 Queue Operations 218**

- Enqueue 218
- Dequeue 218
- Queue Front 219
- Queue Rear 220
- Queue Example 220

**5-2 Queue Linked List Design 220**

- Data Structure 220
- Queue Algorithms 222

Create Queue 224  
 Enqueue 224  
 Dequeue 225  
 Retrieving Queue Data 227  
 Empty Queue 228  
 Full Queue 228  
 Queue Count 228  
 Destroy Queue 229

### 5-3 Queuing Theory 229

### 5-4 Queue Applications 231

Queue Simulation 231  
 Categorizing Data 239

### 5-5 Categorizing Data—C++ Implementation 241

Main Line Logic 241  
 Fill Queues 242  
 Print Queues 244  
 Print One Queue 244

### 5-6 Queue ADT—Linked List Implementation 246

Queue Structure 246  
 Queue ADT Implementation 247

### 5-7 Queue ADT—Array Implementation 253

Array Queues Implementation 254

### 5-8 Summary 261

### 5-9 Practice Sets 262

Exercises 262  
 Problems 265  
 Projects 266

## 6 Recursion 271

### 6-1 Factorial—A Case Study 272

Recursion Defined 272  
 Iterative Solution 273  
 Recursive Solution 273

### 6-2 How Recursion Works 275

### 6-3 Designing Recursive Algorithms 277

The Design Methodology 278  
 Limitations of Recursion 279  
 Design Implementation—Reverse a  
 Linked List 279

### 6-4 Another Case Study—Fibonacci Numbers 281

### 6-5 The Towers of Hanoi 285

Recursive Towers Of Hanoi 286

### 6-6 C++ Implementations of Recursion 290

Fibonacci Numbers 290  
 Prefix to Postfix Conversion 291  
 Towers of Hanoi 297

### 6-7 Summary 299

### 6-8 Practice Sets 300

Exercises 300  
 Problems 302  
 Projects 303

## 7 Introduction to Trees 305

### 7-1 Basic Tree Concepts 306

Terminology 306  
 Tree Representation 308

### 7-2 Binary Trees 310

Properties 312

### 7-3 Binary Tree Traversals 313

Depth-First Traversals 314  
 Breadth-First Traversals 320

### 7-4 Expression Trees 322

Infix Traversal 322  
 Postfix Traversal 324  
 Prefix Traversal 324

### 7-5 General Trees 324

Changing General Tree to Binary  
 Tree 325  
 Insertions into General Trees 326  
 General Tree Deletions 327

### 7-6 Huffman Code 327

### 7-7 Summary 332

### 7-8 Practice Sets 333

Exercises 333  
 Problems 337  
 Projects 337

## 8 Search Trees 338

### 8-1 Binary Search Trees 339

Definition 339

Operations on Binary Search  
Trees 341

Binary Search Tree Search  
Algorithms 342

### **8-2 AVL Trees 353**

AVL Balance Factor 354  
Balancing Trees 355  
AVL Node Structure 360  
AVL Delete Algorithm 367  
Adjusting the Balance Factors 372

### **8-3 AVL Tree Implementation 372**

Data Structure 372  
Program Design 373  
Count Words Summary 377

### **8-4 AVL Abstract Data Type 378**

AVL Tree Data Structures 379  
AVL Tree Functions 380  
AVL Tree Data Processing 392  
AVL Tree Utility Functions 395

### **8-5 Summary 398**

### **8-6 Practice Sets 399**

Exercises 399  
Problems 403  
Projects 403

## **9 Heaps 406**

### **9-1 Heap Definition 407**

### **9-2 Heap Structure 407**

### **9-3 Basic Heap Algorithms 408**

ReheapUp 409  
ReheapDown 411

### **9-4 Heap Data Structure 412**

### **9-5 Heap Algorithms 414**

ReheapUp 414  
ReheapDown 415  
BuildHeap 416  
InsertHeap 418  
DeleteHeap 419

### **9-6 Heap Applications 421**

Selection Algorithms 421  
Priority Queues 423

### **9-7 A Heap Program 424**

Heap Program Design 425  
Heap Functions 430

### **9-8 Summary 433**

### **9-9 Practice Sets 434**

Exercises 434  
Problems 436  
Projects 436

## **10 Multiway Trees 439**

### **10-1 m-Way Search Trees 440**

### **10-2 B-Trees 442**

B-Tree Insertion 443  
B-Tree Insert Design 444  
B-Tree Insert Node 446  
B-Tree Deletion 454  
Traverse B-Tree 467  
B-Tree Search 470

### **10-3 Simplified B-Trees 471**

2-3 Tree 471  
2-3-4 Tree 472

### **10-4 B-Tree Variations 472**

B\*Tree 473  
B+Tree 474

### **10-5 Lexical Search Tree 474**

Tries 475  
Trie Structure 476

### **10-6 B-Tree Abstract Data Type 478**

Header File 479  
Utility Functions 481  
Insert Algorithms 485  
Delete Algorithms 491

### **10-7 Summary 499**

### **10-8 Practice Sets 499**

Exercises 499  
Problems 500  
Projects 501

## **11 Advanced Sorting Concepts 502**

### **11-1 General Sort Concepts 503**

Sort Order 503  
Sort Stability 504  
Sort Efficiency 504  
Passes 505

### **11-2 Insertion Sorts 505**

Straight Insertion Sort 505  
Shell Sort 508

- Insertion Sort Algorithms 513
- Insertion Sort Implementation 515
- 11-3 Selection Sorts 517**
  - Straight Selection Sort 517
  - Selection Sort Algorithms 522
  - Selection Sort Implementation 524
- 11-4 Exchange Sorts 526**
  - Bubble Sort 526
  - Bubble Sort Algorithm 527
  - Quick Sort 529
  - Exchange Sort Algorithms 536
- 11-5 Summary 538**
  - Exchange Sort Implementation 538
- 11-6 External Sorts 542**
  - Merging Ordered Files 543
  - Merging Unordered Files 544
  - The Sorting Process 546
  - Sort Phase Revisited 551
- 11-7 Summary 553**
- 11-8 Practice Sets 554**
  - Exercises 554
  - Problems 556
  - Projects 556

## 12 Graphs 560

- 12-1 Terminology 561**
- 12-2 Operations 563**
  - Add Vertex 563
  - Delete Vertex 563
  - Add Edge 563
  - Delete Edge 564
  - Find Vertex 564
  - Traverse Graph 564
- 12-3 Graph Storage Structures 568**
  - Adjacency Matrix 568
  - Adjacency List 569
- 12-4 Graph Algorithms 570**
  - Create Graph 571
  - Insert Vertex 572
  - Delete Vertex 573
  - Insert Arc 575

- Delete Arc 577
- Retrieve Vertex 578
- First Arc 579
- Depth-First Traversal 581
- Breadth-First Traversal 583

### 12-5 Networks 585

- Minimum Spanning Tree 585
- Shortest Path Algorithm 591

### 12-6 Abstract Data Type 596

- Insert Vertex 598
- Delete Vertex 599
- Insert Arc 600
- Delete Arc 602
- Depth-First Traversal 604
- Breadth-First Traversal 605

### 12-7 Summary 607

### 12-8 Practice Sets 608

- Exercises 608
- Problems 610
- Projects 611

## Appendixes

- A ASCII Tables 615**
- B Structure Charts 620**
- C Program Standards and Styles 627**
- D Random Numbers 633**
- E Standard C++ Libraries 639**
- F C++ Function Prototypes 641**
- G Classes Related to Input and Output 650**
- H The String Class 655**
- I Pointers to Functions 666**
- J Inheritance 670**
- K C++ Templates 685**
- L Standard Template Library 692**

**Solutions to Selected Exercises 712**

**Glossary 735**

**Index 747**