
Contents

Preface	ix
Acknowledgments	xv
Chapter 1. Introduction	1
1.1. History of algorithms	3
1.2. Definition, structure and properties of algorithms	4
1.2.1. Definition	4
1.2.2. Structure and properties	4
1.3. Development of an algorithm	5
1.4. Data structures and algorithms	6
1.5. Data structures – definition and classification	7
1.5.1. Abstract data types	7
1.5.2. Classification	9
1.6. Algorithm design techniques	9
1.7. Organization of the book	11
Chapter 2. Analysis of Algorithms	13
2.1. Efficiency of algorithms	13
2.2. Apriori analysis	15
2.3. Asymptotic notations	17
2.4. Time complexity of an algorithm using the O notation	19
2.5. Polynomial time versus exponential time algorithms	20
2.6. Average, best and worst case complexities	21

2.7. Analyzing recursive programs.	23
2.7.1. Recursive procedures	23
2.7.2. Apriori analysis of recursive functions	27
2.8. Illustrative problems	31
Chapter 3. Arrays	45
3.1. Introduction.	45
3.2. Array operations	46
3.3. Number of elements in an array.	46
3.3.1. One-dimensional array.	46
3.3.2. Two-dimensional array	47
3.3.3. Multidimensional array	47
3.4. Representation of arrays in memory	48
3.4.1. One-dimensional array.	49
3.4.2. Two-dimensional arrays	51
3.4.3. Three-dimensional arrays	52
3.4.4. N-dimensional array	53
3.5. Applications	54
3.5.1. Sparse matrix	54
3.5.2. Ordered lists	55
3.5.3. Strings	56
3.5.4. Bit array	58
3.6. Illustrative problems	60
Chapter 4. Stacks	71
4.1. Introduction.	71
4.2. Stack operations	72
4.2.1. Stack implementation	73
4.2.2. Implementation of push and pop operations	74
4.3. Applications	76
4.3.1. Recursive programming	76
4.3.2. Evaluation of expressions	79
4.4. Illustrative problems	83
Chapter 5. Queues	101
5.1. Introduction.	101
5.2. Operations on queues	102
5.2.1. Queue implementation.	102
5.2.2. Implementation of insert and delete operations on a queue	103

5.2.3. Limitations of linear queues.	105
5.3. Circular queues.	106
5.3.1. Operations on a circular queue	106
5.3.2. Implementation of insertion and deletion operations in circular queue	109
5.4. Other types of queues	112
5.4.1. Priority queues	112
5.4.2. Deques	117
5.5. Applications	119
5.5.1. Application of a linear queue	119
5.5.2. Application of priority queues	120
5.6. Illustrative problems	125
Chapter 6. Linked Lists	143
6.1. Introduction.	143
6.1.1. Drawbacks of sequential data structures	143
6.1.2. Merits of linked data structures.	145
6.1.3. Linked lists – structure and implementation	145
6.2. Singly linked lists	147
6.2.1. Representation of a singly linked list	147
6.2.2. Insertion and deletion in a singly linked list	149
6.3. Circularly linked lists	155
6.3.1. Representation.	155
6.3.2. Advantages of circularly linked lists over singly linked lists	155
6.3.3. Disadvantages of circularly linked lists	156
6.3.4. Primitive operations on circularly linked lists	158
6.3.5. Other operations on circularly linked lists	159
6.4. Doubly linked lists.	160
6.4.1. Representation of a doubly linked list.	161
6.4.2. Advantages and disadvantages of a doubly linked list	162
6.4.3. Operations on doubly linked lists	163
6.5. Multiply linked lists	166
6.6. Unrolled linked lists.	171
6.6.1. Retrieval of an element	172
6.6.2. Insert an element	172
6.6.3. Delete an element.	173
6.7. Self-organizing lists	175
6.8. Applications	175
6.8.1. Addition of polynomials.	176
6.8.2. Sparse matrix representation	178
6.9. Illustrative problems	182

Chapter 7. Linked Stacks and Linked Queues	201
7.1. Introduction.	201
7.1.1. Linked stack	202
7.1.2. Linked queues	203
7.2. Operations on linked stacks and linked queues	203
7.2.1. Linked stack operations	203
7.2.2. Linked queue operations.	204
7.2.3. Algorithms for Push/Pop operations on a linked stack	205
7.2.4. Algorithms for insert and delete operations in a linked queue.	206
7.3. Dynamic memory management and linked stacks	209
7.4. Implementation of linked representations	214
7.5. Applications	216
7.5.1. Balancing symbols	216
7.5.2. Polynomial representation.	218
7.6. Illustrative problems	222
References	241
Index	243
Summaries of other volumes	245