

CONTENTS

1. FUNDAMENTALS OF ALGORITHMS	1
1.1. ALGORITHM	1
1.1.1. Algorithms as a technology	2
1.1.2. The Constraints on the operations	3
1.1.3. Program	3
1.1.4. Distinct areas of study of algorithms.	3
1.2. ANALYZING ALGORITHMS	4
1.2.1. Structural Programming Model	4
1.2.2. Random Access Machine (RAM) Model	5
1.2.3. Asymptotic Notation	6
1.3. DESIGNING ALGORITHMS	9
1.3.1. The Divide and Conquer approach	9
1.3.2. Analyzing Divide-and-Conquer Algorithms	11
1.4. STANDARD NOTATIONS AND COMMON FUNCTIONS	12
1.5. SUMMATION FORMULAS AND PROPERTIES	16
1.6. RECURRENCES	24
1.6.1. Recursion	24
1.6.2. Two example of recursive program	25
1.6.3. Methods of solving recurrences	25
1.7. Exercises in Asymptotic notation	34
1.8. Exercises	35
1.9. Definitions	39
2. FUNDAMENTALS OF DATA STRUCTURES	47
2.0. INTRODUCTION	47
2.1. ABSTRACT DATA TYPE (ADT)	48
2.1.1. Basic Concepts of Data Representations	48
2.1.2. Standard Primitive Types	49
2.1.3. The Array Structure	49
2.1.4. The Record Structure	50
2.1.5. The Set Structure	51

2.1.6.	Representation of Array, Record and Set Structures	51
2.1.8.	Representation of Sets	52
2.1.9.	The Sequential File Structure	52
2.2.	MANIPULATION OF DATA TYPES	53
2.2.1.	Fundamental Operations on Sets	53
2.2.2.	Hashing	55
2.2.3.	Binary Search	56
2.2.4.	Binary Search Trees	57
2.3.	STACKS	57
2.3.1.	Definition	57
2.3.2.	Templates in C++	58
2.3.3.	Stack Linked List Implementation	60
2.3.4.	Stack Applications	64
2.4.	QUEUES	67
2.4.1.	The Abstract Data Type(ADT)	69
2.4.2.	Array Representation	69
2.4.3.	Programs Array Length	70
2.5.	LISTS	71
2.5.1.	Linked List	73
2.5.2.	Linked List Data Structure	74
2.5.3.	Representing lists in C++	75
2.5.4.	Circular List	78
2.5.5.	Doubly Linked Lists	78
2.6.	TREES	79
2.6.1.	Binary Trees	83
2.7.	GRAPHS	87
2.7.1.	Introduction	87
2.7.2.	Definitions	88
2.7.3.	Operations	89
2.8.	SEARCHING	98
2.9.	SORTING	102
2.10.	MERGE SORT	110
2.11.	HASHING	117
2.11.1.	Definitions	117
2.11.2.	Static hashing and dynamic hashing	118
2.11.3.	Static Hashing	118
2.11.4.	Hash functions	119
2.12.	DYNAMIC HASHING	123
2.13.	HEAPS	125
2.13.1.	Definitions	125
2.13.2.	Heap Structure	129
2.13.3.	Heap Data Structure	130

2.13.4.	Build Heap	131
2.13.5.	Program	136
2.13.6.	Program for Insertion into a min – max heap.	139
2.13.7.	Program Insertion into A Min – Max Heap	139
2.13.8.	Program : Searching for the Correct Max Node for Insertion	140
2.13.9.	Analysis of Min Max Heap :: Insert:	140
2.13.10.	Program: Deleting the Element with Minimum Key	141
3.	SEARCH STRUCTURES	160
3.1.	BINARY SEARCH TREES	160
3.1.1.	Abstract data types	161
3.1.2.	Optimal binary search trees	162
3.2.	AVL TREES	169
3.2.1.	Introduction	169
3.2.2.	Definitions	170
3.2.3.	AVL Balance Factor	171
3.3.	RED-BLACK TREES	175
3.3.1.	Definition and Properties	175
3.4.	B-TREES	178
3.4.1.	Definitions	178
3.5.	SPLAY TREES	181
3.5.1.	Introduction	181
3.6.	TRIES	185
3.6.1.	Definition	185
3.6.2.	Implementation of Digital Search Tree by Tree	186
4.	MULTI MEDIA STRUCTURES	187
4.1.	INTRODUCTION	187
4.2.	K-D TREES	188
4.2.1.	Node Structure	188
4.2.2.	Insertion and Search in 2-d Trees	189
4.2.3.	Deletion in 2-d Tree	192
4.2.4.	Range Queries in 2-d Trees	194
4.3.	POINT QUAD TREES	197
4.3.1.	Insertion/Search in Point Quad Trees	197
4.3.2.	Deletion in Point Quad Trees	201
4.3.3.	Range Queries in Point Quad Trees	203
4.4.	THE MX-QUAD TREE	203
4.4.1.	Insertion/search in MX-Quad Trees	204
4.4.2.	Deletion in MX-Quad Trees	206
4.4.3.	Range Queries in MX-Quad Trees	207
4.5.	R-TREES	207

4.5.1.	Insertion/starch in an R-tree	208
4.5.2.	Deletion in R-trees	209
4.6.	COMPARISON OF DIFFERENT DATA STRUCTURES	209
4.7.	TV – TREES	210
4.7.1.	Organization of a TV – Tree	211
4.7.2.	Insertion into TV – Series	213
4.7.3.	Searching in TV – Trees	218
4.7.4.	Nearest Neighbor Retrievals in TV – Trees	219
5.	APPLICATIONS	221
5.1.	GREEDY ALGORITHMS	221
5.1.1.	A Recursive Greedy Algorithm	221
5.1.2.	An Iterative Greedy Algorithm	222
5.1.3.	Application of Greedy Method	223
5.2.	HUFFMAN CODING	226
5.2.1.	Introduction	226
5.2.2.	Constructing a Huffman Code	229
5.3.	Garbage Collection And Compaction	234
5.4.	Automatic List Management	235
5.4.1.	Reference Count Method	235
5.3.2.	Garbage Collection Method	237
5.3.3.	Algorithms for Garbage Collection	237
5.3.4.	Collection and Compaction	239
5.5.	TOPOLOGICAL SORT	240
5.5.1.	Introduction	240
5.5.2.	Topological-Sort (G)	242
5.6.	NP-COMPLETENESS	243
5.6.1.	Introduction	243
5.6.2.	NP-Completeness and the Classes P and NP	244
5.6.3.	Algorithmic Complexity-Intractability	245
6.	RECURSIVE ALGORITHMS	247
6.1.	INTRODUCTION	247
6.2.	WHEN NOT TO USE RECURSION	249
6.3.	TWO EXAMPLES OF RECURSIVE PROGRAMS	250
6.4.	BACK TRACKING ALGORITHMS	254
7.	SHORT AND LONG ANSWER QUESTIONS	266
7.1.	SHORT ANSWER QUESTIONS	266
7.2.	LONG ANSWER QUESTIONS	283
	BIBLIOGRAPHY	304
	INDEX	305