



Unit - I

Chapter 1 : Basics of Algorithms and Mathematics

1- 1 to 1-17

1.1	Introduction to Algorithm.....	1-1
1.1.1	What is an Algorithm ?	1-1
1.1.2	Properties of Algorithm	1-1
1.1.3	How to Write an Algorithm ?	1-2
1.1.4	Specification of Algorithm	1-3
1.2	Mathematics for Algorithmic Sets	1-4
1.2.1	What is Set ?	1-4
1.2.2	Set Representation Methods	1-4
1.2.2 (A)	Listing Method	1-4
1.2.2 (B)	Describing Properties	1-4
1.2.2 (C)	Recursive Definition.....	1-5
1.2.3	Types of Set	1-5
1.2.3 (A)	Subset.....	1-5
1.2.3 (B)	Superset.....	1-5
1.2.3 (C)	Proper Subset	1-5
1.2.3 (D)	Proper Superset	1-5
1.2.3 (E)	Empty Set	1-5
1.2.3 (F)	Power Set.....	1-5
1.2.3 (G)	Equal Set.....	1-6
1.2.4	Operations on Set	1-6
1.2.4 (A)	Union.....	1-6
1.2.4 (B)	Intersection.....	1-6
1.2.4 (C)	Difference.....	1-6
1.2.4 (D)	Symmetric Difference	1-6
1.2.4 (E)	Complement.....	1-6
1.2.4 (F)	Cartesian Product.....	1-6
1.2.4 (G)	Cardinality	1-6
1.3	Functions.....	1-7
1.3.1	Types of Functions	1-7
1.3.1 (A)	One to One Function	1-7
1.3.1 (B)	Many to One Function	1-7
1.3.1 (C)	Onto Function.....	1-7
1.3.1 (D)	Into Function	1-8
1.3.2	Floor, Ceiling and Round Properties of Function	1-8
1.4	Relations	1-8
1.4.1	Types of Relations.....	1-8
1.4.1 (A)	Reflexive Relation.....	1-8
1.4.1 (B)	Transitive Relation.....	1-9
1.4.1 (C)	Symmetric Relation	1-9
1.4.1 (D)	Asymmetric Relation	1-9
1.4.1 (E)	Equivalence Relation.....	1-9

1.4.1 (F)	Antisymmetric Relation.....	1-9
1.4.1 (G)	Partial Order Relation	1-10
1.5	Vectors.....	1-10
1.6	Matrices	1-10
1.6.1	Types of Matrix.....	1-10
1.6.1 (A)	Zero Matrix	1-10
1.6.1 (B)	Square Matrix	1-10
1.6.1 (C)	Identity Matrix	1-10
1.6.1 (D)	Diagonal Matrix	1-10
1.6.1 (E)	Upper Triangular Matrix.....	1-10
1.6.1 (F)	Lower Triangular Matrix.....	1-11
1.6.1 (G)	Permutation Matrix	1-11
1.6.1 (H)	Symmetric Matrix.....	1-11
1.6.1 (I)	Skew Symmetric Matrix.....	1-11
1.6.1 (J)	Sparse Matrix	1-11
1.6.1 (K)	Singular Matrix	1-11
1.6.1 (L)	Non-Singular Matrix	1-11
1.6.2	Matrix Operations	1-11
1.6.2 (A)	Addition	1-11
1.6.2 (B)	Subtraction	1-11
1.6.2 (C)	Scalar Multiplication	1-12
1.6.2 (D)	Matrix Multiplication	1-12
1.6.2 (E)	Determinant.....	1-12
1.6.2 (F)	Transpose	1-12
1.6.2 (G)	Inverse	1-12
1.6.2 (H)	Minor	1-13
1.6.3	Matrix Properties	1-13
1.6.3 (A)	Associative	1-13
1.6.3 (B)	Commutative	1-13
1.6.3 (C)	Distributive	1-13
1.7	Linear Inequalities	1-13
1.7.1	What is Linear Inequality?	1-13
1.7.2	Properties of Linear Inequality	1-13
1.8	Linear Equation	1-13
1.8.1	What is Linear Equation?	1-13
1.8.2	Cremer's Rule	1-14
1.8.3	Gauss Elimination Method.....	1-14
1.8.4	Gauss Jordan Method	1-15
1.9	Solved Problems	1-16

Unit - II

Chapter 2 : Analysis of Algorithms 2-1 to 2-24

2.1	Efficiency of Algorithm.....	2-1
2.1.1	Theoretical Approach	2-1
2.1.2	Empirical Approach	2-2



2.1.3	Hybrid Approach.....	2-2	3.4.1	Sorting Mechanism.....	3-6
2.2	Order of Growth.....	2-2	3.4.2	Algorithm.....	3-6
2.3	Average, Best and Worst Case Analysis.....	2-4	3.4.3	Complexity Analysis.....	3-6
2.3.1	Best Case.....	2-4	3.4.4	Example.....	3-7
2.3.2	Worst Case.....	2-4	3.5	Shell Sort.....	3-8
2.3.3	Average Case.....	2-5	3.5.1	Sorting Mechanism.....	3-8
2.4	Amortized Analysis.....	2-5	3.5.2	Algorithm for Shell.....	3-8
2.4.1	What is Amortized Analysis?.....	2-5	3.5.3	Complexity Analysis.....	3-8
2.4.2	Methods of Amortized Analysis.....	2-5	3.5.4	Example.....	3-8
2.4.2 (A)	Aggregate Method.....	2-5	3.6	Heap Sort.....	3-9
2.4.2 (B)	Accounting Method.....	2-6	3.6.1	Representation of Tree.....	3-9
2.4.2 (C)	Potential Method.....	2-7	3.6.2	Types of Heap.....	3-10
2.5	Asymptotic Notations.....	2-8	3.6.3	Heap Creation.....	3-11
2.5.1	What is Asymptotic Notations?.....	2-8	3.6.3 (A)	Top-Down Approach.....	3-11
2.5.2	Types of Asymptotic Notation.....	2-9	3.6.3 (B)	Bottom-Up Approach.....	3-13
2.5.2 (A)	Big Oh Notation (O).....	2-9	3.6.4	Working Mechanism.....	3-15
2.5.2 (B)	Big Omega Notation (Ω).....	2-12	3.6.5	Algorithm.....	3-15
2.5.2 (C)	Big Theta Notation (Θ).....	2-13	3.6.6	Complexity Analysis.....	3-15
2.5.2 (D)	Little Oh Notation (o).....	2-15	3.6.7	Examples.....	3-16
2.5.2 (E)	Little Omega Notation (ω).....	2-15	3.7	Sorting in Linear Time.....	3-25
2.5.3	Properties of Asymptotic Notation.....	2-15	3.7.1	Counting Sort.....	3-25
2.5.4	Examples.....	2-16	3.7.1 (A)	Sorting Mechanism.....	3-25
2.6	Analyzing Control Statement.....	2-17	3.7.1 (B)	Algorithm.....	3-26
2.6.1	Sequential Execution.....	2-17	3.7.1 (C)	Complexity Analysis.....	3-26
2.6.2	If-else.....	2-17	3.7.1 (D)	Example.....	3-26
2.6.3	For Loop.....	2-18	3.7.2	Radix Sort.....	3-28
2.6.4	While Loop.....	2-18	3.7.2 (A)	Sorting Mechanism.....	3-28
2.6.5	Recursion.....	2-19	3.7.2 (B)	Algorithm.....	3-28
2.6.6	Examples.....	2-20	3.7.2 (C)	Complexity Analysis.....	3-28
2.7	Loop Invariant and the.....		3.7.2 (D)	Example.....	3-29
	Correctness of the Algorithm.....	2-23	3.7.3	Bucket Sort.....	3-29
			3.7.3 (A)	Working Mechanism.....	3-29
			3.7.3 (B)	Algorithm.....	3-29
			3.7.3 (C)	Complexity Analysis.....	3-29
			3.7.3 (D)	Examples.....	3-29
Chapter 3 : Sorting Techniques			3-1 to 3-30		
3.1	Introduction to Sorting.....	3-1	<div style="border: 1px solid black; padding: 10px; display: inline-block;">Unit - III</div>		
3.1.1	What is Sorting?.....	3-1			
3.1.2	Properties of Sorting Algorithms.....	3-1			
3.2	Insertion Sort.....	3-1			
3.2.1	Sorting Mechanism.....	3-1			
3.2.2	Algorithm.....	3-2			
3.2.3	Complexity Analysis.....	3-2			
3.2.4	Example.....	3-3			
3.3	Bubble Sort.....	3-4			
3.3.1	Sorting Mechanism.....	3-4			
3.3.2	Algorithm.....	3-4			
3.3.3	Complexity Analysis.....	3-5			
3.3.4	Example.....	3-5			
3.4	Selection Sort.....	3-6			
Chapter 4 : Divide and Conquer			4-1 to 4-46		
4.1	Introduction.....	4-1			
4.1.1	General Strategy.....	4-1			
4.1.2	Applications.....	4-2			
4.1.3	Control Abstraction.....	4-2			
4.1.4	Efficiency Analysis.....	4-2			
4.2	Recurrence and Different Methods.....				
	to Solve Recurrence.....	4-3			



4.2.1	Unfolding Methods.....	4-3
4.2.1 (A)	Forward Substitution.....	4-4
4.2.1 (B)	Backward Substitution.....	4-4
4.2.2	Homogeneous Equations	4-5
4.2.3	Inhomogeneous Equation.....	4-7
4.2.4	Master's Theorem	4-11
4.2.5	Change of Variable.....	4-13
4.2.6	Examples	4-13
4.3	Multiplying Large Integers Problem	4-15
4.3.1	Grade School Multiplication.....	4-15
4.3.2	Divide and Conquer Approach.....	4-15
4.3.2 (A)	Working Mechanism.....	4-17
4.3.2 (B)	Algorithm.....	4-17
4.3.2 (C)	Complexity Analysis	4-17
4.3.2 (D)	Examples	4-18
4.4	Searching.....	4-19
4.4.1	Linear Search.....	4-19
4.4.1 (A)	Working Mechanism.....	4-19
4.4.1 (B)	Algorithm.....	4-19
4.4.1 (C)	Complexity Analysis	4-20
4.4.1 (D)	Examples	4-20
4.4.2	Binary Search.....	4-21
4.4.2(A)	Working Mechanism.....	4-21
4.4.2 (B)	Algorithm.....	4-21
4.4.2 (C)	Complexity Analysis	4-22
4.4.2 (D)	Examples	4-23
4.5	Merge Sort	4-23
4.5.1	Working Mechanism.....	4-23
4.5.2	Algorithm.....	4-24
4.5.3	Complexity Analysis	4-25
4.5.4	Properties of Merge Sort	4-25
4.5.5	Examples	4-26
4.6	Quick Sort	4-30
4.6.1	Working Mechanism.....	4-30
4.6.2	Algorithm.....	4-31
4.6.3	Complexity Analysis	4-32
4.6.4	Examples	4-34
4.7	Merge Sort vs. QuickSort	4-40
4.8	Max-Min Problem	4-40
4.8.1	Working Mechanism.....	4-40
4.8.2	Algorithm.....	4-41
4.8.3	Complexity Analysis	4-41
4.9	Matrix Multiplication.....	4-42
4.9.1	Traditional Approach	4-42
4.9.1 (A)	Working Mechanism.....	4-42
4.9.1 (B)	Algorithm.....	4-42
4.9.1 (C)	Complexity Analysis	4-42

4.9.2	Strassen's Method.....	4-42
4.9.2 (A)	Working Mechanism.....	4-42
4.9.2 (B)	Algorithm.....	4-43
4.9.2 (C)	Complexity Analysis	4-43
4.9.2 (D)	Examples	4-43
4.10	Exponential	4-44
4.10.1	Working Mechanism.....	4-44
4.10.2	Algorithm.....	4-44
4.10.3	Complexity Analysis	4-45
4.10.4	Example	4-45

Unit - V

Chapter 5 : Greedy Algorithms 5-1 to 5-33

5.1	Introduction	5-1
5.1.1	General Strategy	5-1
5.1.2	Pillar of Greedy Algorithms.....	5-1
5.1.3	Control Abstraction.....	5-2
5.1.4	Divide and Conquer vs. Greedy Algorithms	5-2
5.2	Elements of Greedy Strategy	5-3
5.3	Problem Solving using Greedy Algorithm	5-3
5.4	Activity Selection Problem.....	5-3
5.4.1	Working Mechanism.....	5-3
5.4.2	Algorithm.....	5-4
5.4.3	Complexity Analysis	5-4
5.4.4	Examples	5-4
5.5	Knapsack Problem	5-5
5.5.1	Binary Knapsack Problem	5-5
5.5.1 (A)	Working Mechanism.....	5-5
5.5.1 (B)	Algorithm.....	5-6
5.5.1 (C)	Complexity Analysis	5-6
5.5.1 (D)	Examples	5-6
5.5.2	Fractional Knapsack Problem.....	5-8
5.5.2 (A)	Working Mechanism.....	5-8
5.5.2 (B)	Algorithm.....	5-9
5.5.2 (C)	Complexity Analysis	5-9
5.5.2 (D)	Examples	5-10
5.6	Job Scheduling Problem.....	5-12
5.6.1	Working Mechanism.....	5-12
5.6.2	Algorithm.....	5-12
5.6.3	Complexity Analysis	5-12
5.6.4	Examples	5-13
5.7	Huffman Code	5-14
5.7.1	Working Mechanism.....	5-14
5.7.2	Algorithm.....	5-15
5.7.3	Complexity Analysis	5-15



5.7.4	Examples	5-15
5.8	Minimum Spanning Trees.....	5-16
5.8.1	Basics of Graph.....	5-16
5.8.2	Kruskal's Algorithm.....	5-18
5.8.2 (A)	Working Mechanism.....	5-18
5.8.2 (B)	Algorithm.....	5-18
5.8.2 (C)	Time Complexity.....	5-19
5.8.2 (D)	Examples	5-19
5.8.3	Prim's Algorithm	5-21
5.8.3 (A)	Working Mechanism.....	5-21
5.8.3 (B)	Algorithm.....	5-22
5.8.3 (C)	Complexity Analysis	5-22
5.8.3 (D)	Examples	5-23
5.8.4	Prim's vs. Kruskal's Algorithm	5-26
5.9	Graphs	5-26
5.9.1	Shortest Path Problems	5-26
5.9.1 (A)	Working Mechanism.....	5-26
5.9.1 (B)	Algorithm.....	5-27
5.9.1 (C)	Complexity Analysis	5-27
5.9.1 (D)	Examples	5-28

Unit - IV

Chapter 6 : Dynamic Programming 6-1 to 6-61

6.1	Introduction	6-1
6.1.1	General Strategy	6-1
6.1.2	Control Abstraction.....	6-1
6.1.3	Characteristics of Dynamic Programming.....	6-2
6.1.4	Applications of Dynamic Programming.....	6-2
6.2	Principle of Optimality.....	6-2
6.3	Elements of Dynamic Programming	6-3
6.4	Comparison with Other Methods	6-3
6.4.1	Divide and Conquer Vs Dynamic Programming.....	6-3
6.4.2	Dynamic Programming Vs Greedy Approach	6-4
6.5	Calculating the Binomial Coefficient	6-4
6.5.1	Binomial Coefficient using Divide and Conquer	6-4
6.5.2	Binomial Coefficient using Dynamic Programming	6-5
6.6	Making Change Problem	6-6
6.6.1	Working Mechanism.....	6-6
6.6.2	Algorithm.....	6-6
6.6.3	Complexity Analysis	6-7
6.6.4	Examples	6-7
6.7	Assembly Line-Scheduling	6-11
6.7.1	Working Mechanism.....	6-11
6.7.2	Algorithm.....	6-12
6.7.3	Complexity Analysis	6-12

6.7.4	Examples	6-13
6.8	Knapsack Problem	6-16
6.8.1	First Approach.....	6-17
6.8.1 (A)	Working Mechanism.....	6-17
6.8.1 (B)	Algorithm.....	6-17
6.8.1 (C)	Complexity Analysis	6-19
6.8.1 (D)	Examples	6-19
6.8.2	Second Approach.....	6-29
6.8.2 (A)	Working Mechanism.....	6-29
6.8.2 (B)	Algorithm.....	6-30
6.8.2 (C)	Examples	6-30
6.9	All Pair Shortest Path	6-32
6.9.1	Working Mechanism.....	6-32
6.9.2	Algorithms	6-34
6.9.3	Complexity Analysis	6-34
6.9.4	Examples	6-34
6.10	Matrix chain multiplication	6-36
6.10.1	Working Mechanism.....	6-36
6.10.2	Algorithm.....	6-38
6.10.3	Complexity Analysis	6-39
6.10.4	Examples	6-39
6.11	Longest Common Subsequence	6-49
6.11.1	Working Principle	6-49
6.11.2	Algorithm.....	6-49
6.11.3	Complexity Analysis	6-50
6.11.4	Examples	6-50

Unit - VI

Chapter 7 : Exploring Graphs 7-1 to 7-16

7.1	Introduction	7-1
7.1.1	What is a Graph?	7-1
7.1.2	Undirected Graph.....	7-1
7.1.3	Directed Graph.....	7-1
7.1.4	Properties of Graph	7-2
7.1.4 (A)	Weighted Graph	7-2
7.1.4 (B)	Sub Graph.....	7-2
7.1.4 (C)	Sparse Graph.....	7-2
7.1.4 (D)	Dense Graph.....	7-2
7.1.4 (E)	Tree.....	7-2
7.1.4 (F)	Spanning Tree.....	7-3
7.1.4 (G)	Minimum Spanning Tree	7-3
7.1.4 (H)	Multi Graph.....	7-3
7.1.4 (I)	Complete Graph	7-3
7.1.4 (J)	Connected Graph	7-3
7.2	Representation of Graph	7-4



7.2.1	Adjacency Matrix Representation	7-4
7.2.2	Adjacency List Representation	7-5
7.3	Traversing Graphs	7-5
7.3.1	Depth First Search	7-5
7.3.2	Breadth-First Search	7-10
7.3.3	Differentiate DFS and BFS	7-12
7.4	Topological Sort	7-12
7.5	Connected Components	7-13
7.5.1	Strongly Connected Components	7-13
7.5.2	Graph Components	7-14
7.5.2 (A)	Articulation Point	7-14
7.5.2 (B)	Bi-Connected Component	7-16

Unit - VII

Chapter 8 : Backtracking 8-1 to 8-12

8.1	Introduction	8-1
8.1.1	General Method.....	8-1
8.1.2	Terminology	8-2
8.1.3	Control Abstraction.....	8-3
8.1.3 (A)	Recursive Backtracking Method	8-3
8.1.3 (B)	Iterative Backtracking Method	8-4
8.1.4	Applications of Backtracking.....	8-4
8.2	The 8-Queen Problem	8-4
8.3	Knapsack Problem	8-9

Chapter 9 : Branch and Bound 9-1 to 9-30

9.1	Introduction	9-1
9.1.1	General Method.....	9-1
9.1.2	Backtracking vs. Branch and Bound.....	9-2
9.1.3	Applications of Branch and Bound.....	9-3
9.2	Control Abstraction.....	9-3
9.2.1	LC Search	9-3
9.2.2	Control Abstraction for Least Cost Search.....	9-4
9.2.3	Bounding	9-5
9.2.4	Control Abstraction for FIFO Branch and Bound.....	9-5
9.3	Knapsack Problem	9-6
9.3.1	Knapsack Using LCBB	9-8
9.3.2	Knapsack using FIFO	9-12
9.4	Travelling Salesman Problem.....	9-16

9.4.1	LCBB using Static State Space Tree	9-16
9.4.2	LCBB using Dynamic State-Space Tree.....	9-24
9.5	Minimax Principle	9-29

Unit - VIII

Chapter 10 : String Matching 10-1 to 10-10

10.1	Introduction	10-1
10.2	Naive String Matching Algorithm	10-1
10.3	String Matching with Finite Automata	10-3
10.4	Rabin-Karp Algorithm	10-5
10.5	Knuth-Morris-Pratt Algorithm	10-7

Unit - IX

Chapter 11 : Introduction to NP Completeness

11-1 to 11-14

11.1	Basic Definitions.....	11-1
11.2	P Problems	11-2
11.3	NP Problems	11-2
11.4	P Problems vs. NP Problems	11-3
11.5	Polynomial Reduction.....	11-3
11.6	NP-Complete.....	11-4
11.6.1	NP-Complete Problem	11-4
11.6.2	Clique Problem.....	11-4
11.6.3	Vertex Cover Problem	11-5
11.6.4	Travelling Salesman Problem.....	11-7
11.6.5	Hamiltonian Problem	11-7
11.7	NP-Hard Problems	11-8
11.8	NP-Complete vs. NP-Hard	11-8
11.9	Approximation Algorithms.....	11-8
11.9.1	Introduction	11-8
11.9.2	Performance Ratios.....	11-9
11.9.3	Vertex Cover Problem	11-10
11.9.4	Travelling Salesman Problem.....	11-11
11.10	Randomized Algorithms	11-12
11.10.1	Introduction	11-12
11.10.2	Randomized Quick Sort.....	11-13
11.11	Class of Problems beyond NP – P SPACE	11-13