

Class	Topic	Unit	MT
1	Introduction	1	MT-1
2	Running Time Analysis (Simple codes - independent of machine speeds, why log n, how to calculate in recursion - fib)	1	
3	Asympmtotic Notation	1	
4	Exaustive Search TSP, Knapsack, Assignment	1	
5	Brute Force, String Matching	2	
6	Closest Pair, Convex Hull	2	
7	Recursion, Factorial, Divide and Conquer	2	
8	Fibonacci, Binary Search	2	
9	Quick Sort, Mergesort	2	
10	Starssens Matrix Multiplication, Master's Theorem	2	
11	Master's Theorem	1	
12	MT-1 Paper Discussion		
13	Dynamic Programmig Recursion, Fibonacci, 01 Knapsack	3	MT-2
14	Principal of Optimality,, Dynamic Programmig Memoization, Fibonacci, 01 Knapsack	3	
15	Dynamic Programmig Tabulation, Fibonacci, 01 Knapsack	3	
16	Coin Change (Minimum Number of Coin, Total Number of Ways)	3	
17	Binomial Coefficient , Floyd's algorithm , Multistage graph	3	
18	Sum of Subsets	3	
19	MCM	3	
20	OBST	3	
21	Greedy, Fractional Knapsack	3	
22	Minium Spanning Tree, Prims Algorithm	3	
23	Union Find, Kruskal's Algorithm	3	
24	Dijkstra's Shortest Path	3	
25	Huffman Encoding	3	MT-3
26	Optimal merge patterns	3	
27	MT-2 Paper Discussion		
28	Backtracking, Permutation	4	
29	NQeen, Sum of Subsets	4	
30	Hamiltonian Cycles	4	
31	Graph Coloring	4	
32	Branch and Bound, 01 Kanpsack	4	
33	Travelling Salesman Problem	4	
34	Assignment problem (Least Cost BB)	4	
35	NP Complete	5	
36	NP Complete	5	
37	NP Complete	5	
38	Approximate Algo	5	
39	Approximate Algo	5	
40	Conclusion		

Personal Time Table:

Time	Mon	Tue	Wed	Thu	Fri	Sat
Slot-1			C3			
Slot-2			C2			
Slot-3	C4					
Slot-4	C1					
LUNCH BREAK						
Slot-5		C4	C1	C2	C2	
Slot-6		C3	C4	C1	C3	

Course Code	Course Title				Course Type	
CS2201	Design and Analysis of Algorithms				PCC	
Prerequisite	Contact Hours per Week			Internal	External	Credits
	L	T	P			
	3	0	0	40	60	3

Course Objectives

- Introduces the notations for analysis of the performance of algorithms.
- Describes major algorithmic techniques (divide-and-conquer, backtracking, dynamic programming, greedy, branch and bound methods) and mention problems for which each technique is appropriate;
- Describes how to evaluate and compare different algorithms using worst-, average-, and best-case analysis.
- Explains the difference between tractable and intractable problems, and introduces the problems that are P, NP and NPcomplete.

Course Outcomes

1. Students will understand the basic concepts of algorithm design and analysis.
2. Students will be able to analyze and evaluate algorithm performance using asymptotic notations.
3. Students will learn to apply various algorithmic techniques such as divide and conquer, dynamic programming, and greedy method.
4. Students will gain proficiency in solving problems using backtracking and branch and bound methods.
5. Students will gain an understanding of NP-hard and NP-complete problems and their implications in the real world.

Detailed Contents

UNIT-I

Introduction:

Notion of an Algorithm – Fundamentals of Algorithmic Problem Solving – Important Problem Types – Fundamentals of the Analysis of Algorithmic Efficiency –Asymptotic Notations and their properties. Analysis Framework – Empirical analysis – Mathematical analysis for Recursive and Non-recursive algorithms – Visualization

UNIT-II

Brute Force – String Matching , Closest-Pair and Convex-Hull Problems.

Exhaustive Search – Traveling Salesman Problem, Knapsack Problem , Assignment problem.

Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Strassen's matrix multiplication,convex hull,closest pair, large integer multiplication.

UNIT-III

Dynamic programming – Principle of optimality,Chain Matrix Multiplication, Computing a Binomial Coefficient , Floyd's algorithm , Multistage graph , Optimal Binary Search Trees ,

Knapsack Problem and Memory functions.

Greedy Technique –Prim’s algorithm and Kruskal’s Algorithm , Fractional Knapsack problem, Optimal Merge pattern – Huffman Trees.

UNIT-IV

Backtracking: General method, applications n-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

Branch and Bound: General method, applications - Traveling Salesperson Problem, 0/1 knapsack problem, Assignment problem - LC Branch and Bound solution, FIFO Branch and Bound solution.

UNIT-V

NP-Hard and NP-Complete problems: NP Hard and NP completeness: Basic concepts, cook’s theorem, NP-hard graph problems and scheduling problem, NP- hard code generation problems, Clique Decision problem, Node covering problem, scheduling problem, NP hard code generation problem.

Approximation Algorithms for NP-Hard Problems – Traveling Salesman problem, Knapsack problem.

Text Books

- Horowitz, E., Sahni, S., and Rajasekaran, S. (2019). Fundamentals of Computer Algorithms. University Press.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). Introduction to Algorithms, Second Edition. Pearson Education.

References

1. V. Levitin, "Introduction to the Design and Analysis of Algorithms," Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
2. A. Aho, J. Ullman, and M. Hopcroft, "Design and Analysis of Algorithms," Pearson Education.
3. M. T. Goodrich and R. Tamassia, "Algorithm Design: Foundations, Analysis and Internet Examples," John Wiley and Sons.