III B. Tech. – I Semester

(20BT53103) ALGORITHM ANALYSIS

(Skill-Oriented Course)

(Common to CSE(AI),CSE(DS) and CSE(AI&ML)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	2	-	-	2

PRE-REQUISITES: A Course on "Data Structures"

COURSE DESCRIPTION: Algorithms and asymptotic notations; Algorithm performance analysis; Recurrences; Disjoint sets; Divide and Conquer; Dynamic programming; Greedy algorithms; Back tracking; Branch and bound; NP-hard and NP-complete problems.

COURSE OUTCOMES: *After successful completion of this course, the students will be able to:*

CO1. Analyze the complexity of algorithms by applying the knowledge of asymptotic notations and recurrence methods.

CO2. Analyze the given problem and identify appropriate algorithm design technique for problem solving.

CO3. Perceive and apply different algorithm design paradigms to find solutions for computing problems.

CO4. Apply the knowledge of NP-hard and NP-Complete complexity classes to classify decision problems.

DETAILED SYLLABUS:

UNIT- I: INTRODUCTION TO ALGORITHMS (6 Periods)

Algorithm, Algorithm pseudocode conventions, Performance analysis - Space complexity, Time complexity, Asymptotic notations; Recurrences - Substitution method, Recursion-tree method, Master method.

UNIT- II: DISJOINT SETS, DIVIDE AND CONQUER (6 Periods)

Disjoint Sets: Operations, Union and Find algorithms.

Divide and Conquer: General method, Binary search, Finding maximum and minimum, Strassen's matrix multiplication.

UNIT- III: DYNAMIC PROGRAMMING (6 Periods)

General method, Matrix-chain multiplication, 0/1 Knapsack problem, All-pairs shortest path, Traveling salesperson problem.

UNIT- IV: GREEDY METHOD, BACKTRACKING (6 Periods)

Greedy Method: General method, Knapsack problem, Job sequencing with deadlines, Single source shortest paths.

Backtracking: General method, Sum of subsets, Graph coloring.

UNIT- V: BRANCH AND BOUND, NP-HARD AND NP-COMPLETE PROBLEMS

(6 Periods)

Branch and Bound: LC search, LC branch and bound, FIFO branch and bound, 0/1 knapsack problem using LCBB.

NP Hard and NP-Complete Problems: Nondeterministic algorithms, NP-hard and NP-complete classes.

Total Periods: 30

Topics for self-study are provided in the lesson plan

TEXT BOOKS:

1. Ellis Horowitz, Sartaj Sahni, and SanguthevarRajasekaran*, Fundamentals of Computer Algorithms,* 2nd Edition, Universities Press, 2008.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, *Introduction to Algorithms*,3rd Edition, MIT Press, 2009.

REFERENCE BOOKS:

1. Michael T. Goodrich and Roberto Tamassia*, Algorithm Design and Applications,* Wiley, 2014.
2. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, *The Design and Analysis of Computer Algorithms*, Pearson, 2006.

ADDITIONAL LEARNING RESOURCES:

- https://nptel.ac.in/courses/106/106/106106131/

CO-PO-PSO Mapping Table:

Course Outcome	Program Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	-	-	-	-	-	-	-	-	-	-	3	-	-	-
CO2	3	3	-	-	-	-	-	-	-	-	-	-	3	-	-	-
CO3	2	3	3	-	-	-	-	-	-	-	-	-	3	-	-	-
CO4	3	3	-	-	-	-	-	-	-	-	-	-	3	-	-	-

Correlation Level: 3-High 2-Medium 1-Low

III B. Tech. – I Semester

(20BT513131) ADVANCED PROGRAMMING LAB

(Common to CSE(AI),CSE(DS) and CSE(AI&ML)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	-	-	3	1.5

PRE-REQUISITES: A Course on "Algorithm Analysis"

COURSE DESCRIPTION: Problem solving using Linear search; Divide and conquer – Problem solving using Binary search, Quick Sort, Merge Sort; Dynamic Programming – 0/1 Knapsack, All-pairs shortest path, Problem-solving; Greedy method; Greedy method – Knapsack problem, Single source shortest paths, Problem-solving; Backtracking method - Sum of subsets.

COURSE OUTCOMES: *After successful completion of this course, the students will be able to:*

CO1. Analyze the complexity of algorithms by applying the knowledge of asymptotic notations and recurrence methods.

CO2. Analyze the given problem and identify appropriate algorithm design technique for problem solving.

CO3. Perceive and apply different algorithm design paradigms to implement solutions for computing problems.

CO4. Work independently or in team to solve problems with effective communication.

LIST OF EXERCISES:

1. Given an array of integers A, write a program to find the maximum sum that can be obtained by picking some non-empty subset of the array. If there are many such non-empty subsets, choose the one with the maximum number of elements. Print the maximum sum and the number of elements in the chosen subset (Hint: Use Linear Search technique to solve the problem).

2. A flight company has to schedule a journey of N groups of people from a source city to a destination city. Here, G_1, G_2, . . . G_N represent the number of people in each group. All groups are present at the source city. The flight company has M planes where P_1, P_2, . . . P_M represents the capacity of each plane. Write a program to provide solution so that all groups can be sent from source to the destination with the following conditions:

    i) Each plane can travel from the Source to Destination with only one group at a time such that capacity of a plane is enough to accommodate all people in that group.
    ii) All people belonging to the same group should travel together.
    iii) Every plane can make multiple journeys between source and destination.
    iv) It costs 1 unit of time to travel between source to destination and vice versa.

v) Multiple planes can fly together and also it is not necessary for planes to end their journey at the source

Determine the minimum time required to send all groups from the source to the destination. (Hint: Use Binary Search technique to solve the problem)

3. Given an array A of length N and given an integer X, write a program to find an integer Z such that Z is strictly greater than X and Z is not present in the array A. The objective is to minimize the value of Z. (Hint: Use Quick Sort technique to solve the problem)
4. Sort a given set of N integer elements using merge sort method and compute its time complexity. Run the program for varied values of n >1000 and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator. Demonstrate how the divide and conquer method works along with its time complexity analysis: worst case, average case and best case.
5. There are N students in the class where each of them has two parameters, marks and the bound. There are two arrays M and B denoting the marks and the bound of N students in the class. Initially, the cutoff is zero for passing. Now, each of the students will be evaluated on the basis of his/her marks. If the marks obtained by the student are greater than the cutoff, he or she will pass and the cutoff will be incremented by the bound of that student. Write a program to find the maximum number of students that can pass if the arrangement of students is done properly. (Hint: Use Dynamic Programming technique to solve the problem)
6. Write a program to implement 0/1 Knapsack problem using Dynamic Programming method.
7. Write a program to implement All-Pairs Shortest Paths problem using FloydWarshall algorithm.
8. There are N binary strings (consisting of 0s and 1s) in a table. A pair of strings is good if there exists at least one character in common in both strings. Write a program to find and display the good pairs of strings. (Hint: Use Dynamic Programming technique to solve the problem)
9. Write a program to implement Knapsack problem using greedy method.
10. Write a program to implement Dijkstra's Algorithm for solving Single Source Shortest Path problem using greedy approach.
11. Any number is called beautiful if it consists of 2N digits and the sum of the first N digits is equal to the sum of the last N digits. Write a program task is to find beautiful numbers in the given interval from M to N (including M and N). (Hint: Use greedy approach to solve the problem).
12. Write a program to find a subset of a given set S = {Sl, S2,.....,Sn} of n positive integers whose SUM is equal to a given positive integer d. For example, if S ={1, 2, 5, 6, 8} and d= 9, there are two solutions {1,2,6}and {1,8}. Display a suitable message, if the given problem instance doesn't have a solution. (Hint: Use Backtracking technique to solve the problem)

REFERENCE BOOKS:

1. Ellis Horowitz, SatrajSahni and Rajasekharam, *Fundamentals of Computer Algorithms*, Galgotia Publications Pvt. Ltd, New Delhi, 2nd Edition, 2007.

SOFTWARE/TOOLS USED:

Programming Language: C/Java/Python

Tools Used: Any IDE for C/Java/Python

ADDITIONAL LEARNING RESOURCES:

- NPTEL course on Design and Analysis of Algorithms

  URL: https://nptel.ac.in/courses/106/101/106101060/

CO-PO-PSO Mapping Table:

Course Outcome	Program Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	-	-	-	-	-	-	-	-	-	-	3	-	-	-
CO2	3	3	-	-	-	-	-	-	-	-	-	-	3	-	-	-
CO3	2	3	3	3	3	-	-	-	-	-	-	-	3	-	-	-
CO4	3	3	-	-	-	-	-	-	-	-	-	-	3	-	-	-

Correlation Level: 3-High 2-Medium 1-Low