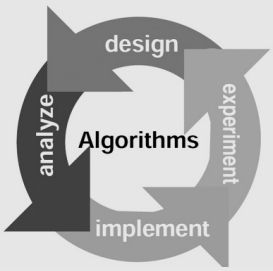
22CS206 DESIGN AND ANALYSIS OF ALGORITHMS



Source: https://www.facebook.com/Design-and-Analysis-of-Algorithms-1553902878155564/

Hours Per Week :

L	T	P	C
2	2	2	4

PREREQUISITE KNOWLEDGE: Programming for problem solving, Discrete Mathematical Structures, Data Structures.

COURSE DESCRIPTION AND OBJECTIVES:

This course offers the basic knowledge required to analyze the asymptotic performance of algorithms. In addition, this course provides the knowledge required to solve different problems using suitable design strategies such as the greedy method, divide and conquer, dynamic programming, backtracking and branch & bound. This course helps to understand the impact of the choice of data structures and algorithm design strategies on the performance. This course also provides the understanding of advanced graph applications and throws light on tractable and intractable problems.

MODULE-1

UNIT-1 **6L+6T+6P=18 Hours**

INTRODUCTION

Algorithm, Pseudo-code for expressing algorithms, Performance analysis – space and time complexity; Asymptotic notation - big oh notation, Omega notation, Theta notation and little oh notation; Analysis of recursive algorithms through recurrence relations- substitution method, Recursion tree method, Masters Theorem.

Disjoint sets: Disjoint set operations, Union and find algorithms.

UNIT-2 **10L+10T+10P=30 Hours**

DIVIDE & CONQUER AND GREEDY METHOD

Divide and Conquer: General method, Applications - Binary search, Quick sort, Merge sort and Strassen's matrix multiplication.

Greedy Method: Applications - job sequencing with deadlines, Knapsack problem, Minimum cost spanning trees.

PRACTICES:

- Sort a given set of elements using the following methods and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n inputs. The elements can be read from a file or can be generated using the random number generator.
  a. Quick sort b. Merge sort
- Search for a given set of elements using the following methods and determine the time required to search the given element. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus no. of elements. The elements can be read from a file or can be generated using the random number generator.
  a. Linear Search b. Binary Search
- Implement the following using divide and conquer approach.
  To multiply two given square matrices.
  To multiply two given square matrices using Strassen's matrix multiplication.
- Design the Algorithm to solve Job sequencing with deadlines problem and Analyze its time complexity. Implement the above algorithm using Greedy method.

SKILLS:

- ✓ *Analyze the given algorithm concerning space and time complexities and compare it with other algorithms.*
- ✓ *Develop algorithms for solving problems using divide and conquer, greedy, dynamic programming, backtracking and branch & bound techniques.*
- ✓ *Application of existing design strategies to solve real-world problems.*

- Design the Algorithm to solve fractional Knapsack problem using Greedy method. Analyze the time complexity and implement the above algorithm.
- Design the Algorithm to find minimum spanning tree and its cost for an undirected graph. Analyze the time complexity and implement the above algorithm.

MODULE-2

UNIT-1 **10L+10T+10P=30 Hours**

DYNAMIC PROGRAMMING AND BACKTRACKING

Dynamic Programming: General method, Applications - optimal binary search trees, Matrix chain multiplication, 0/1knapsackproblem, All pairs shortest path problem, Travelling sales person problem.

Backtracking: General method, Applications - N-Queen problem, Sum of subsets problem, Graph colouring and Hamiltonian cycles.

UNIT-2 **6L+6T+6P=18 Hours**

BRANCH & BOUND AND P, NP, NP - HARD AND NP-COMPLETE

Branch and Bound: General method, Applications- Travelling sales person problem, 0/1 knapsack problem using LC branch and bound solution and FIFO branch and bound solution.

P, NP, NP - HARD and NP-Complete: Basic Concepts - Non-Deterministic Algorithms - The Classes NP-Hard and NP Complete- NP Hard Problems- Clique Decision Problem-Cook's Theorem.

PRACTICES:

- Design the Algorithm to find all pairs shortest path problem by using dynamic programming approach. Analyze its time complexity and implement the above algorithm.
- Design the Algorithm to find optimal binary search tree and its cost by using dynamic programming approach. Analyze its time complexity and implement the above algorithm.
- Design the Algorithm to find optimal order of matrix chain multiplication and its cost using dynamic programming approach. Analyze its time complexity and implement the above algorithm.
- Design the Algorithm to find optimal route for travelling sales person problem and its cost by using dynamic Programming approach. Analyze its time complexity and implement the above algorithm.
- Design the Algorithm to solve N-queens problem by using backtracking approach and Analyze its time complexity. Implement the above algorithm.
- Design the Algorithm to solve sum of subsets problem using backtracking approach and Analyze its time complexity. Implement the above algorithm.
- Design the Algorithm to solve 0/1 Knapsack problem using Branch and Bound method. Analyze the time complexity and Implement the above algorithm.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Analyze the efficiency of a given algorithm using time and space complexity theory. Understanding algorithmic design strategy like divide and conquer approach.	Analyze	1	1, 2, 12
2	Apply greedy algorithm Strategy for suit able problems and argue the correctness of such algorithms with respect to the global optimization.	Apply	1	1, 2,3, 5, 12
3	Apply the dynamic programming paradigm and identify the kind of problem best suited to solve using dynamic programming.	Apply	2	1, 2, 3, 5, 12
4	Compare and contrast the design principles of branch and bound with backtracking strategy.	Apply	2	1, 2,3,5, 12
5	Investigate computational complexity of different class of problems.	Analyze	2	1, 2,4,12

TEXT BOOKS:

1. Ellis Horowitz, SatrajSahni and Rajasekharan, “Fundamentals of Computer Algorithms”, 2nd Edition, Galgotia publications, 2006.
2. Thomas H. Coremen, Charles E. Leiserson and Ronald L. Rivest, “Introduction to Algorithm”, 2nd Edition, MIT press Ltd., 2014.

REFERENCE BOOKS:

1. Anony Levitin, “Introduction to Design and Analysis of Algorithms”, 3rd Edition, Pearson Education, 2016.
2. Donald E. Knuth, “The Art of Computer Programming”, 2nd Edition, Addison Wesley Publishing Company, 1998.
3. Ronald L. Graham, Donald E. Knuth and Oren Patashnik, “Concrete Mathematics”, 2nd Edition, Addison wesley Publishing Company,1998.
4. Dasgupta, Papadimitriou and Vazirani,“Algorithms”, 1st Edition, McGraw-Hill publishers, 2008.
5. Weiss, “Data Structures and Algorithm Analysis”, 1st Edition, Addison-Wesley Publishing Company, 2016.