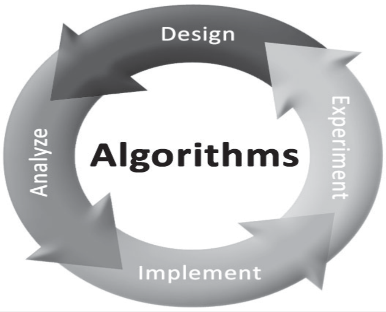
16CS206 DESIGN AND ANALYSIS OF ALGORITHMS



Hours Per Week:

L	T	P	C
3	-	2	4

Total Hours:

L	T	P
45	-	30

WA/RA	SSH/HSH	CS	SA	S	BS
5	40	3	8	5	2

Course Description and Objectives:

This course offers insight into the time and space complexities of various algorithms, design of algorithms using divide and conquer, greedy, dynamic, branch and bound, and back tracking approches. The objective of this course is to design and analyze the algorithms for their time and space complexities and also to understand problems such as 0/1 knapsack, shortest path, minmum spanning tree, matrix multiplication, graph coloring, n-queens and travelling salesman problem.

Course Outcomes:

The student will be able to:

- argue the correctness of algorithms using inductive proofs and invariants.
- analyze worst case running times of algorithms using asymptotic analysis.
- derive and solve recurrences describing the performance of divide and conquer algorithms.
- design the greedy algorithms and analyze them.
- design dynamic programming algorithms and analyze them.
- analyze major graph algorithms.

SKILLS:

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

UNIT - 1 **L-9**

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; Randomized algorithms; Disjoint sets - Disjoint set operations, Union and find algorithms, Connected components and Biconnected components.

UNIT - 2 **L-9**

DIVIDE AND CONQUER: General method, Applications - Binary search, Quick sort, Merge sort, and Stassen's matrix multiplication, Greedy method; Applications - Job sequencing with deadlines, Knapsack problem, Minimum cost spanning trees and Tree vertex splitting problem, Single source shortest path.

UNIT - 3 **L-9**

DYNAMIC PROGRAMMING: General method, Applications - Multi stage graphs, Optimal binary search trees, Matrix chain multiplication, 0/1 knapsack problem, All pairs shortest path problem, Travelling sales person problem.

UNIT - 4 **L-9**

BACKTRACKING: General method, Applications – n-queen problem, Sum of subsets problem, Graph coloring, Hamiltonian cycles.

UNIT - 5 **L-9**

BRANCH AND BOUND & COMPLETE PROBLEMS: General method, Applications - Traveling sales person problem, 0/1 knapsack problem, LC Branch and Bound solution, FIFO Branch and Bound solution, NP - Hard and NP - Basic concepts, Non deterministic algorithms, NP – Hard and NP complete classes, Cook's theorem.

ACTIVITIES:

- o *Analysis of various algorithmic techniques to solve problems.*
- o *Implementing robust set of algorithms to solve new problems efficiently.*
- o *Design and implementation of solutions using dynamic programming methods.*
- o *Design and implementation of various backtracking algorithms to solve the given problem.*
- o *Compare the performance of different problem solving strategies for given applications.*

LABORATORY EXPERIMENTS

Course Outcomes:

The student will be able to:

- analyze a problem & design the solution for the problem.
- derive solution must be optimum, i.e., time complexity & memory usage of the solution must be very low.
- learn and execute various problem solving mechanisms.

LIST OF EXPERIMENTS

Total Hours: 30

Language: C++

1. Implement the following:
   a) Prim's algorithm.
   b) Kruskal's algorithm.
2. Find optimal order of matrix multiplication using dynamic programming method.
3. Consider the problem of eight queens on an (8x8) chessboard. Two queens are said to attack each other if they are on the same row, column, or diagonal. Implements backtracking algorithm to solve the problem i.e. place eight non-attacking queens on the board.

4. Implement the following:
   a. Optimal Binary Search Trees Using Dynamic Programming.
   b. Job Sequencing With Deadlines Using Greedy Approach.
   c. dynamic programming algorithm to solve all pairs shortest path problem.
   d. sum of subsets problem by using backtracking.
   e. 0/1 knapsack problem using Greedy algorithm.
   f. 0/1 knapsack problem using Dynamic programming algorithm.
   g. 0/1 knapsack problem using Branch and Bound.

TEXT BOOK:

1. Ellis Horowitz, Satraj Sahni and Rajasekharam, “Fundamentals of Computer Algorithms”, 2nd edition, Galgotia publications, 2006.

REFERENCE BOOKS:

1. Thomas H.Coremen,Charles E.Leiserson and Ronald L.Rivest, “ Introduction to Algorithm”, 2nd edition, Clifford Stein, 2014.
2. Anony Levitin, “Introduction to Design and Analysis of Algorithms”, 3rd edition, Pearson Education, 2016.
3. Donald E.knuth, “The Art of Computer Programming”, Volume 3, 2nd edition, Addison-wesley Longman Inc,1998.
4. Ronald L.Graham,Donald E.Knuth and Oren Patashnik, “Concrete Mathematics”, 2nd edition, Addison-wesley Publishing Company,1998.
5. Jeffrey J.McConnell, “Analysis of Algorithms: An Active Learning Approach”, 1st edition, Jones and Bartlett Publishers, 2001.