Design and Analysis of Algorithm(Theory)

Course Code	18CS44/18IS44	**Credits**	03
Course type	PC	**CIE Marks**	50 marks
Hours/week: L-T-P	3 – 0 – 0	**SEE Marks**	50 marks
Total Hours:	Lecture = 40 Hrs; Tutorial = 00 Hrs Total = 40 Hrs	**SEE Duration**	3 Hours for 100 marks

Unit – I **8 Hours**

Introduction: Fundamentals of Algorithmic Problem Solving, Analysis Framework, Asymptotic Notations and basic efficiency classes, Mathematical Analysis of Non-Recursive and Recursive Algorithms, Brute Force Approaches: Introduction, Selection Sort, linear search.

Self learning topics: Short Tutorial on Recurrence Relations, Bubble Sort(1Hr)

Unit – II **8 Hours**

Algorithm Design Technique-I: Divide and Conquer, Decrease-and-Conquer Transform and Conquer, the General approach and illustration.

Applications of Divide and Conquer technique: Binary Search, Merge Sort, Quick Sort and their performance comparison. Counting Leaf-nodes, Tiling-Game Implementation.

Applications of Decrease and Conquer technique: Insertion Sort, Depth First Search and Breadth First Search. Maze-Game implementation.

Applications of Transform and Conquer: Heaps and Heap Sort, Horner's Rule. Clustering.

Self learning topics: Multiplication of Large Integers and Binary Exponentiation. (2 Hrs)

Unit – III **8 Hours**

Algorithm Design Technique-II: The General Greedy Technique, Illustration with examples**.**

Applications of Greedy method: Kruskal's Algorithm – Minimum-Cost Spanning Trees: Prim's Algorithm, Single Source Shortest Path - Dijkstra's Algorithm, Huffman Trees – Encoding of Data.

Unit – IV **8 Hours**

Algorithm Design Technique-III: Dynamic Programming Definition and Concept Illustration. The General Method,

Applications of Dynamic programming: Warshall's Algorithm – Transitive Closure, Floyd's Algorithm for the All-Pairs Shortest Paths, Knapsack using General Weights and 0/1 Knapsack. Longest Common Difference – Used in implementation of Diff command and polynomial interpolation.

Self learning topics: Computing nCr, the dynamic approach (1 Hr)

Unit – V **8Hours**

Algorithm Design Technique-IV: Backtracking, Branch-and-Bound, String Matching, basics and illustrations.

Applications of backtracking: N - Queens's problem, Hamiltonian Circuit Problem, Sum of Subset – Problem and its use in public key cryptosystem. Graph coloring problem.

Applications of branch and bound: JobAssignment Problem, Knapsack Problem, Traveling Salesperson Problem. Best First Search used in AI.
Applications string matching: Input Enhancement in String Matching, Horspool's method, Rabin-Karp Algorithm. Used in Text processing toolkits like nltk.
Self learning topics: Naïve String Matching Algorithm. (1Hr)

Text Books:

1. Anany Levitin, Introduction to The Design & Analysis of Algorithms, Pearson Education 1st edition and onwards.
2. Ellis Horowitz, SartajSahni, SanguthevarRajasekaran, Fundamentals of Computer Algorithms Universities Press, 1st edition and onwards.

Reference Books:

1. Kenneth Berman, Jerome Paul, Algorithms, Cengage Learning.
2. Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, introduction to Algorithms PHI, 2nd edition and above.
3. R.C.T. Lee, S.S. Tseng, R.C. Chang &Y.T.Tsai: Introduction to the Design and analysis of Algorithms A Strategic Approach, TataMcGraw Hill.
4. NarasimhaKarumanchi, Data structures and Algorithms Made Easy, Career Monk Publications, 1st edition and above.

E Resources:
https://onlinecourses.nptel.ac.in/

Course Outcome (COs)

At the end of the course, the student will be able to	**Bloom's Level**
1. **Formulate and Solve** recurrence equation and compute time complexity of recursive and iterative algorithms	L3
2. **Explain** divide ,decrease ,transform and conquer strategy as applied to sorting and analyze the algorithm complexity	L2
3. **Apply** Dynamic Programming, Greedy approach, to solve a variety of problems.	L3
4. **Design and analyze** String search algorithms and Compare their time complexities.	L4
5. **Apply** branch and bound and backtracking approaches to solve a variety of practical problems	L3

Program Outcome of this course (POs)	**PO No.**
1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	**PO1**

2.	**Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	**PO2**
3.	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	**PO4**

Algorithms Laboratory

Course Code	18CSL47/18ISL47	**Credits**	1.5
Course type	LAB	**CIE Marks**	25 marks
Hours/week: L-T-P	0 – 0 – 3	**SEE Marks**	25 marks
Total Hours:	36	**SEE Duration**	3 Hours for 50 marks

Course learning objectives

1. Illustrate the importance of algorithms in a variety of applications.
2. Illustrate the use of recursive/iterative sorting algorithms in different scenarios.
3. Demonstrate time complexity of various algorithms using various design techniques.
4. Demonstrate efficient algorithms by drawing comparisons.
5. Illustrate the use of algorithms for graph search problems.

Pre-requisites :

- Basic computer science concepts such as procedures, decision statements, and loops.
- Basic data structures such as lists, dictionaries, and hash tables.

List of experiments(Programming language C / Java)

1. Implement Merge Sort algorithm to sort a given set of elements 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.
2. Implement Quick Sort algorithm 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.
3. Implement Insertion Sort algorithm 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.
4. Implement Heap Sort algorithm 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.
5. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.
6. Find the Minimum Cost Spanning Tree of a given undirected graph using Prim’s algorithm.

7. Implement All-Pairs Shortest Paths Problem using Floyd's algorithm.
8. Implement 0/1 Knapsack problem using Dynamic Programming.
9. 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}. A suitable message is to be displayed if the given problem instance doesn't have a solution.
10. Implement N Queen's problem using Back Tracking.

Text Books:

1. Anany Levitin, Introduction to The Design & Analysis of Algorithms, Pearson Education, 1st edition and onwards.
2. Java, The Complete Reference, Herbert Schildt.

Reference Books:

1. Kenneth Berman, Jerome Paul, Algorithms, Cengage Learning.
2. Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, introduction to Algorithms PHI, 2nd edition and onwards.

E Resources:
https://onlinecourses.nptel.ac.in/

Course Outcome (COs)

At the end of the course, the student will be able to	**Bloom's Level**
1. **Identify** and implement an appropriate algorithm design technique for a given problem.	**L1**
2. **Implement** and **Compute** time required for recursive and iterative algorithms.	**L3**
3. **Design** algorithms for specific applications using appropriate techniques.	**L6**
4. **Design** graph search and sorting algorithms.	**L6**

Program Outcome of this course (POs)	**PO No.**
1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	**PO1**
2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	**PO2**
3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	**PO3**
4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	**PO4**

Assessment methods

1. Regular Journal Evaluation & Attendance Monitoring.
2. Lab Internal Assessment.

Scheme of Continuous Internal Evaluation (CIE):

Components	Attendance/conduct of lab	Journal	Lab project	Total Marks
Maximum marks :25	10	10	5	25
Submission and certification of journal is compulsory to qualify for SEE Minimum marks required to qualify for SEE : 10 out of 25 marks				

Semester End Examination (SEE):

1.	It will be conducted for 50 marks having 3 hours/2 hours duration. It will be reduced to 25 marks for the calculation of SGPA and CGPA.		
2.	Only one experiment to be conducted. In case, there are two parts then one experiment from each part.		
3.	Initial write up:Algorithm/Flowchart/Tracing	10 marks	50 marks
	Conduct of experiment(s), result and conclusion	20 marks	
	One marks question	10 marks	
	Viva-voce	10 marks	
4.	Viva voce is conducted for individual student and not in group		
5.	Minimum passing marks to be scored in SEE: 20 out of 50 marks		

Course delivery methods	Assessment methods
1. Lecture & Board	1. Assignments
2. Power-point Presentation	2. Quizzes
3. Online Videos / Learning	3. Internal Assessment Tests
4. NPTEL / Edusat	4. Course Seminar
5. Class Room Exercises	5. Course Project (Mini project)
	6. Case Studies

Scheme of Continuous Internal Evaluation (CIE):

Components	Addition of two IA tests	Average of two assignments	Quiz/Seminar/Course Project	Total Marks
Maximum marks :50	15+15 = 30	10	10	50
Writing two IA tests is compulsory. Minimum marks required to qualify for SEE : 20 out of 50 marks				

Semester End Examination (SEE):

1. It will be conducted for 3 hours duration and 100 marks. It will be reduced to 50 marks for the calculation of SGPA and CGPA.
2. Minimum passing marks required to be scored in SEE: 40 out of 100 marks
3. Question paper will have 10 questions carrying 20 marks each. Students have to answer FIVE full questions selecting atleast one full question from each unit.