Course Description

PAPER NAME : Design & Analysis of Algorithm

PAPER CODE : PCC-CS 404 & PCC-CS 494

Course Description

Course Title/Code: <u>Design and Analysis of Algorithm/PCC-CS404 & PCC-CS494</u>

Semester:- <u>2nd</u> Year:- <u>2nd</u> Group:- <u>A</u>

Name of the Faculty: **Sucharita Das**

E-mail : sucharita.das007@gmail.com

i) Course Objective:

Students will be able to apply different programming design paradigm to develop new algorithms and also analyze the efficiency of its algorithm.

ii) Course Outcomes:

After completion of this course the students are expected to be able to demonstrate following knowledge, skills and attitudes.

a) The Students will be able to:

Course Outcomes		Targets
PCC-CS404.1	Express a logic sentence in terms of predicates, quantifiers, and logical connectives . (**BT-Level 1**)	60% marks
PCC-CS404.2	Derive the solution for a given problem using deductive logic and prove the solution based on logical inference. (**BT-Level 2**)	60% marks
PCC-CS404.3	Classify its algebraic structure for a given a mathematical problem. (**BT-Level 3**)	60% marks
PCC-CS404.4	Evaluate Boolean functions and simplify expressions using the properties of Boolean algebra. (**BT-Level 4**)	60% marks
PCC-CS4	Develop the given problem as graph networks and solve with techniques of graph theory. (**BT-Level 3**)	

b)

Once the student has successfully complete this course, he/she must be able to answer the following questions or perform/demonstrate the following:

SN	QUESTION	BT-LEVEL
1.	**What** do you understand by an algorithm?	1
2.	**What** are the analytic issues of an algorithm?	1
3.	**Write** an algorithm to find the **maximum number among three numbers** and also calculate the running time complexity**.**	1
4.	**Write** an algorithm to calculate the **sum of two matrices** and also calculate the running time complexity.	1
5.	**Define Cook's** theorem. Prove that **3-SAT** is NP- Complete.	2
6.	**Find** out the Recurrence relation of recursive **Tower of Hanoi** problem and solve it for the input size n.	3
7.	**Solve** the following recurrence using iteration method. 1. $T(n)=2T(n/2) + O(n)$	3
8.	**Solve** the following recurrence using master method. 1. $T(n)=2T(n/2) + O(n)$ 2. $T(n)=4T(n/2) + O(n)$ 3. $T(n)=T(n/2) + O(n)$	3
9.	**Show** that the following equation is correct: $33n^2 + 4n = \Omega(n^2)$	3
10.	**Solve** $T(n) = aT(n/b) + O(n^k)$ where $a > 1$ and $b \geq 1$.	3
11.	**Find** out the running time complexity of the **Quick sort** algorithm in Best, Worst and Average cases.	4
12.	**Find** out the running time complexity of the **N-Queen** problem.	4
13.	**Implement** adjacent matrix and adjacent list of a given graph and also conclude which representation is better**.**	5
14.	**Implement** graph traversal techniques like **BFS** and **DFS**.	5
15.	**Implement Binary Search** with the help of **Divide & Conquer** strategy.	6
16.	**Implement** shortest path using **Dijkstra's** algorithm with the help of dynamic programming strategy**.**	6

Design & Analysis of Algorithm syllabus [in Chapters]

Code: CS501

Contact: 3L + 1T

CHAPTER-1

Complexity Analysis: [4L]

Time and Space Complexity, Different Asymptotic notations – their mathematical significance

CHAPTER-2

Heap Sort and its complexity [2L]

CHAPTER-3

Divide and Conquer: [3L]

Basic method, use, following case studies with proper analysis.

1) Binary Search.
2) Merge Sort.
3) Quick Sort and their complexity.

CHAPTER-4

Dynamic Programming: [4L]

Basic method, use, following case studies with proper analysis.

1) Matrix Chain Multiplication.
2) All pair shortest paths
   a. Floyd-Warshall Algorithm.
3) Single source shortest path.
   a. Dijkstra's Algorithm.
   b. Bellmanford Algorithm.

CHAPTER-5

Backtracking: [2L]

Basic method, use, following case studies with proper analysis.

1) n queens problem.
2) Graph coloring problem.

CHAPTER-6

Greedy Method: [4L]

Basic method, use, ,following case studies with proper analysis.

1) Knapsack problem.
2) Job sequencing with deadlines.
3) Minimum cost spanning tree
   a. Prim's Algorithm.
   b. Kruskal's Algorithm.

CHAPTER-7

Lower Bound Theory: [1L]

Prove O(nlg(n)) bound for comparison sort .

CHAPTER-8

Disjoint set manipulation: [1L]

Set manipulation algorithm like UNION-FIND, union by rank.

CHAPTER-9

Graph traversal algorithm: [3L]

1) Breadth First Search(BFS)
2) Depth First Search(DFS)
3) Classification of edges - tree, forward, back and cross edges – complexity and comparison

CHAPTER-10

String matching problem: [2L]

Different techniques Naive algorithm, string matching using finite automata, and Knuth, Morris, Pratt (KMP) algorithm with their complexities.

CHAPTER-11

Amortized Analysis: [2L]

Aggregate, Accounting, and Potential Method.

CHAPTER-12

Network Flow: [3L]

Ford Fulkerson algorithm, Max-Flow Min-Cut theorem (Statement and Illustration)

CHAPTER-13

Matrix Manipulation Algorithm: [3L]

Strassen's matrix manipulation algorithm; application of matrix multiplication to solution of simultaneous linear equations using LUP decomposition, Inversion of matrix and Boolean matrix multiplication.

CHAPTER-14

Notion of NP-completeness: [4L]

P class, NP class, NP hard class, NP complete class – their interrelationship, Satisfiability problem, Cook's theorem (Statement only), and Clique decision problem.

CHAPTER-15

Approximation Algorithms:[1L]

Necessity of approximation scheme, performance guarantee, polynomial time approximation schemes, vertex cover problem, travelling salesman problem.

c) Chapter Layout

Chapter No.	Chapter	Lecture Hours	Tutorials	Laboratory hours

Chapter - 1	Complexity Analysis	4 HRS	1	6 HRS
Chapter – 2	Heap Sort and its complexity	2 HRS	1	3 HRS
Chapter – 3	Divide and Conquer	3 HRS	1	3 HRS
Chapter – 4	Dynamic Programming	4 HRS	1	6 HRS
Chapter – 5	Backtracking	2 HRS		3 HRS
Chapter – 6	Greedy Method	4 HRS	1	3 HRS
Chapter – 7	Lower Bound Theory	1 HRS		
Chapter – 8	Disjoint set manipulation	1 HRS	1	
Chapter – 9	Graph traversal algorithm	3 HRS	1	3 HRS
Chapter – 10	String matching problem	2 HRS	1	3 HRS
Chapter – 11	Amortized Analysis	2 HRS		
Chapter – 12	Network Flow:	3 HRS	1	
Chapter – 13	Matrix Manipulation Algorithm	3 HRS	1	
Chapter – 14	Notion of NP-completeness	4 HRS	1	
Chapter - 15	Approximation Algorithms	1 HRS		
Total		39 HRS	11	30 HRS

d) Textbooks:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein , “Introduction to Algorithms”
2. Aho, J. Hopcroft and J. Ullman “The Design and Analysis of Algorithms” D. E. Knuth “The Art of Computer Programming”, Vol. 3
3. Jon Kleiberg and Eva Tardos, "Algorithm Design"

e) Reference Books:

1. K. Mehlhorn , “Data Structures and Algorithms” - Vol. I & Vol. 2.
2. S. Baase “Computer Algorithms”
3. E. Horowitz and Shani “Fundamentals of Computer Algorithms”

f) Evaluation Scheme:

1) THEORY

Evaluation Criteria	Marks
First & Second Internal Exam*	15
Quiz/ Assignments	10
Attendance	5

University Exam	70
Total	100

*Two internal examinations are conducted; based on those two tests, average of them are considered in a scale of 15.

University Grading System:

Grade	Marks
O	90% and above
E	80 – 89.9%
A	70 – 79.9%
B	60 – 69.9%
C	50 – 59.9%
D	40 – 49.9%
F	Below 40%

2) LABORATORY

Evaluation Criteria	Marks
Internal Exam*	40
University Exam	60
Total	100

* Internal Evaluation will be based on daily lab performance as per the following schedule:

g) Laboratory Evaluation:

Expt. No.	Experiment Name	Schedule	Marks
P1	Experiment on different **Searching** Techniques and also judge the running time complexity. List of Experiments --- **1) Linear Search** 2) **Binary Search**	3 HRS	2 + 2
P2	Experiment on different **Sorting** techniques and also judge the running time complexity. List of Experiments --- **3) Merge Sort** 4) **Quick Sort**	3 HRS	2 + 2
P3	Experiment on different **Sorting** techniques and also judge the running time complexity. List of Experiments --- **5) Heap Sort**	3 HRS	2 + 2

	6) **Counting Sort**		
P4	Experiment on some **recursion** problems also judge the running time complexity as well as plot the graph. List of Experiments --- **7) Calculate x^y** **8) Nth Fibonacci Number** 9) **Tower of Hanoi etc and**	3 HRS	1+1+2
P5	Experiment on **Dynamic Programming** algorithm strategy and also judge the running time complexity. **10) Matrix Chain Multiplication.**	3 HRS	4
P6	Experiment on **Dynamic Programming** algorithm strategy and also judge the running time complexity. 11) **Floyd's Algorithm**	3 HRS	4
P7	Experiment on **Backtracking** algorithm strategy and also judge the running time complexity. List of Experiments --- **12) 4 Queen** 13) **Graph Coloring**	3 HRS	2+2
P8	Experiment on **Minimum Spanning Tree** and also judge the running time complexity. (Any one) List of Experiments --- **14) Prim's Algorithm** 15) **Kruskal's Algorithm**	3 HRS	4
P9	Experiment on **Graph Traversal Techniques** and also judge the running time complexity. List of Experiments --- **16) BFS** 17) **DFS**	3 HRS	2 + 2
P10	Experiment on **String Matching** Algorithm and also judge the running time complexity. **18) KMP**	3 HRS	4

Overall Course Attainment Target

Attainment Level	Inference	Marks
Attainment Level 1	50% of the students have attained more than the target level of that CO	1
Attainment Level 2	60% of the students have attained more than the target level of that CO	2
Attainment Level 3	70% of the students have attained more than the target level of that CO	3

(70% of university and 30% of the internal exam) will be = **Attainment Level 3**

Ta[illegible]een set on the basis of last year's performance / result by the students, student quality this year and difficulty level of the course.

h) A. Weekly Lesson Plan

Week	Lecture	Tutorial	Laboratory	Assignment/Quiz
1	Complexity Analysis.	Tutorial on Linear/Non Linear Data Structures. (TS1)	Review on basic algorithms.	---
2	Heap Sort.	Tutorial on Asymptotic Notations & Recurrences (TS2)	Recursion (P4)	Assignment - I (A1)
3	Binary Search, Merge Sort, Quick Sort.	Tutorial on Heap Sort & Binary Search. (TS3)	Linear & Binary Search (P1)	---
4	Matrix Chain Multiplication. Single Source Shortest Path. (Dijkstra's & Bellman Ford)	Tutorial on Binary Search, Merge Sort & Quick Sort. (TS4)	Sorting (P2)	Assignment - II (A2)
5	All Pair Shortest Path (Floyd's Algorithm). N-Queen, Graph Coloring.	Tutorial on Matrix Chain Multiplication & Shortest Path Problem. (TS5)	Sorting (P3)	Quiz – I (Q1)
6	Knapsack problem. Kruskal's Algorithm. Prim's Algorithm.	Tutorial on Floyd's algorithm. (TS6)	Matrix Chain (P5)	---
7	Job Sequencing with deadline. Lower Bound Theory. Disjoint set manipulation.	Tutorial on Knapsack Problem & Job Sequencing. (TS7)	Floyd's (P6)	Assignment - III (A3)
8	Graph traversal algorithm (BFS & DFS)	Tutorial on MST. (TS8)	N-Queen & Graph Coloring (P7)	Quiz – II (Q2)
9	String Matching.	Tutorial on BFS & DFS. (TS9)	Prim's & Kruskal's (P8)	---
10	Amortized Analysis Approximation Algorithms.	-----	BFS & DFS (P9)	---
11	Network Flow, Ford- Fulkerson algorithm.	-----		---
12	System of Linear Equations Solve by LUP. Strassen's Matrix Multiplication. Matrix Inversion & Boolean Matrix Multiplication.	Tutorial on Network Flow. (TS12)	KMP (P10)	---
13	Notion of NP-completeness.	Tutorial on LUP. (TS10)	---	Quiz – III

		Tutorial on Matrix Inversion. (TS11)		(Q3)