FR. Conceicao Rodrigues College Of Engineering

Department of Computer Engineering

S.E. (Computer) (semester III)

(2019-2020)

Subject: Analysis of Algorithms

Subject Code: CSC 402

Course Outcomes and Assessment Plan

Syllabus:

Course Objectives:

- To provide mathematical approach for Analysis of Algorithms
- To solve problems using various strategies
- To analyze strategies for solving problems not solvable in polynomial time.

Course Outcomes:

At the end of the course student will be able to

1. Analyze the running time and space complexity of algorithms.
2. Describe, apply and analyze the complexity of divide and conquer strategy.
3. Describe, apply and analyze the complexity of greedy strategy.
4. Describe, apply and analyze the complexity of dynamic programming strategy.
5. Explain and apply backtracking, branch and bound and string matching techniques to deal with some hard problems.
6. Describe the classes P, NP, and NP-Complete and be able to prove that a certain problem is NP-Complete.

Module 1 Introduction to analysis of algorithm - 12 HRS

Performance analysis, space and time complexity, Growth of function – Big –Oh, Omega, Theta notation, Mathematical background for algorithm analysis, Analysis of selection sort, insertion sort. **Recurrences:** -The substitution method, Recursion tree method, Master method

Divide and Conquer Approach: General method, Analysis of Merge sort, Analysis of Quick sort, Analysis of Binary search, Finding minimum and maximum algorithm and analysis, Stassen's matrix multiplication

Module 2: Dynamic Programming Approach: 08 HRS

General Method, Multistage graphs, single source shortest path, all pair shortest path, Assembly-line scheduling, 0/1 knapsack, Travelling salesman problem, Longest common subsequence

Module 3: Greedy Method Approach: 06 HRS

General Method ,Single source shortest path, Knapsack problem, Job sequencing with deadlines Minimum cost spanning trees-Kruskal and prim's algorithm ,Optimal storage on tapes

Module 4: Backtracking and Branch-and-bound: 08 HRS

General Method, 8 queen problem(N-queen problem) ,Sum of subsets, Graph coloring ,15 puzzle problem, Travelling salesman problem.

Module 5 :String Matching Algorithms: 06 HRS

The naïve string matching Algorithms, The Rabin Karp algorithm, String matching with finite automata, The knuth-Morris-Pratt algorithm

Module 6 : Non-deterministic polynomial algorithms: 08 HRS

Polynomial time, Polynomial time verification NP Completeness and reducibility NP Completeness proofs Vertex Cover Problems Clique Problems

Text Books:

1. T.H.coreman , C.E. Leiserson,R.L. Rivest, and C. Stein, "Introduction to algorithms", 2nd edition , PHI publication 2005.
2. 2. Ellis horowitz , Sartaj Sahni , S. Rajsekaran. "Fundamentals of computer algorithms" University Press

Reference Books:

1. Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, "Algorithms", Tata McGraw- Hill Edition.
2. S. K. Basu, "Design Methods and Analysis of Algorithm", PHI.
3. John Kleinberg, Eva Tardos, "Algorithm Design", Pearson.
4. Michael T. Goodrich, Roberto Tamassia, "Algorithm Design", Wiley Publication.

Course Outcomes:

Upon completion of this course students will be able to:

CSC 402.1 : Apply the methods for analyzing the complexity of the algorithms. **(Apply)**

CSC 402.2 : Analyze different techniques of algorithm design.(greedy,dynamic,divide and conquer, backtracking, branch and bound). **(Analyze)**

CSC 402.3 : Analyze different String matching techniques. **(Analyze)**

CSC 402.4 : Implement algorithms using different designing techniques. **(Apply)**

Mapping of CO and PO/PSO

Relationship of course outcomes with program outcomes: Indicate 1 (low importance), 2 (Moderate Importance) or 3 (High Importance) in respective mapping cell.

	PO1 (Engg Know)	PO2 (Ana)	PO3 (De sign)	PO4 (Inve stiga)	PO5 (tools)	PO6 (engg Soci)	PO7 (Env)	PO8 (Eth)	PO9 (Ind Team)	PO10 (comm.)	PO11 (PM)	PO12 (life Long)
CSC402.1	3	2										
CSC402.2	3	3										
CSC402.3	3	3										
CSC402.4	3	3	3						1			
Course To PO	3	3	1						1			

CO	PSO1	PSO2
CSC402.1	3	2
CSC402.2	3	2
CSC402.3	3	2
CSC402.4	3	2
Course to PSO	3	2

Justification

PO1: CSC 402.1, CSC 402.2, CSC 402.3 and CSC402.4 maps to PO1 as engineering graduates apply the knowledge of mathematics and computer programming knowledge for providing solution to complex engineering problem.

Course Outcomes:

Upon completion of this course students will be able to:

CSC 402.1 : Apply the methods for analyzing the complexity of the algorithms. (Apply)

CSC 402.2 : Analyze different techniques of algorithm design.(greedy,dynamic,divide and conquer, backtracking, branch and bound). (Analyze)

CSC 402.3 : Analyze different String matching techniques. (Analyze)

CSC 402.4 : Implement algorithms using different designing techniques. (Apply)

Mapping of CO and PO/PSO

Relationship of course outcomes with program outcomes: Indicate 1 (low importance), 2 (Moderate Importance) or 3 (High Importance) in respective mapping cell.

	PO1 (Engg Know)	PO2 (Ana)	PO3 (De sign)	PO4 (inve stiga)	PO5 (tools)	PO6 (engg Soci)	PO7 (Env)	PO8 (Eth)	PO9 (ind Team)	PO10 (comm.)	PO11 (PM)	PO12 (life Long)
CSC402.1	3	2										
CSC402.2	3	3										
CSC402.3	3	3										
CSC402.4	3	3	3						1			
Course To PO	3	3	1						1			

CO	PSO1	PSO2
CSC402.1	3	2
CSC402.2	3	2
CSC402.3	3	2
CSC402.4	3	2
Course to PSO	3	2

Justification

PO1: CSC 402.1, CSC 402.2, CSC 402.3 and CSC402.4 maps to PO1 as engineering graduates apply the knowledge of mathematics and computer programming knowledge for providing solution to complex engineering problem.

PO2: CSC 402.1, CSC 402.2, CSC 402.3 ,CSC 402.4 maps to PO1 as engineering graduates identify and formulate a solution to a problem by analyzing efficiency of different algorithms using their time and space complexities ,selecting a design technique (greedy,dynamic,backtracking) as per the requirement of solution .

PO3: CSC 402.4 maps to PO3 because engineering graduates design a programmed solution to a problem using any high level programming language such as C,C++.

PO9: CSC402.4 maps to PO9 as students worked in a team for developing solution to real world problem by applying proper strategy

PSO1: CSC 402.1 to CSC402.4 maps to PSO1 because the graduates will be able to apply knowledge learnt in the subject to provide solution to real world problems.

PSO2: CSC 402.1 to CSC 402.4 maps to PSO2 as the students design and implement a programmed solution for a real world problem.

Assessment Tools:

Course Outcome	Assessment Tool Direct (weightage: 80%)	Assessment Tool Indirect (weightage= 20%)
CO1: Apply the methods for analyzing the complexity of the algorithms. (PO1)	Test 1 (20%) Postlab Assignment (10%) Assignment 1(20%) Quiz (10%) University Exam (30%) Gate questions(10%)	Course Exit Survey
CO2: Analyze different techniques of algorithm design.(greedy, dynamic, divide and conquer, backtracking, branch and bound). .	Test1+Test2 (20%) Postlab assignment(10%) Assignment 1(20%) Quiz (10%) University Exam(30%) Gate questions(10%)	
CO3: Analyze different String matching techniques.	Test 2(20%) Assignment 2(20%) Post lab assignment(20%) University Exam(30%) Gate questions(10%)	

CO4: Implement algorithms using different design strategies. (PO4)	Lab Work(50%) University Exam(20%) Assignment 2 marks(10%) Real world problem (20%)	

CO Assessment Tools:

CSC402.1: **Direct Methods(80%):** Unit Test 1 + PostLab + Assignment 1+Quiz+UniExam+Gate_Quest

CO1dm = 0.2T +0.1PLab+0.2Assignment+0.1Quiz + 0.3Uniexam+0.1Gate_Quest

InDirect Methods(20%): Course exit survey

CO1idm

CSC402.1 = 0.8*CO1dm + 0.2* CO1idm

CPC501.2:Direct Methods (80%):

Unit Test1&2+PostLab+Assignment+Quiz+UniExam+Gate_Quest

CO2dm = 0.2T +0.1PLab+ 0.2Assig+ 0.1Quiz+0.3Uniexam+0.1Gate_Quest

InDirect Methods(20%): Course exit survey

CO2idm

CSC402.2 = 0.8*CO2dm + 0.2* CO2idm

CPC501.3: **Direct Methods (80%):** Unit Test 2+PostLab+Assignment+Quiz+UniExam

CO3dm = 0.20T +0.1PLab+ 0.2Assig+ 0.3Uniexam+0.1Gate_Quest+0.1Quiz

InDirect Methods(20%): Course exit survey

CO3idm

CSC402.3 = 0.8*CO3dm + 0.2* CO3idm

CPC501.4: **Direct Methods (80%):** Lab assignments+Uniexam+Assig2+Real_world_problem

CO4dm=0.5LabAssignment+0.3UniExam+0.2Assign+0.2Real_world_Problem

InDirect Methods(20%): Course exit survey

CO4idm

CSC402.4 = 0.8*CO4dm + 0.2* CO4id

Rubrics for Lab Experiments:

Sr. No	Performance Indicator	Excellent	Good	Satisfactory	Unsatisfactory
1)	Completeness and correctness [4]	Well commented and formatted, program functions correctly for all input cases. [4M]	Comparatively less use of comments, inconsistent formatting. Program functions correctly for all input cases.[3M]	Inconsistent comments and formatting. Program functions correctly for most of the input cases.[2M]	Improper formatting, no comments. Program functions correctly for very limited cases [1M]
2)	Efficiency [3]	The code could be reused as a whole or each routine could be reused. It is readable and easy to understand [3M]	Most of the code could be reused in other programs. It is fairly readable and easy to understand [2M]	Only Some parts of the code could be reused in other programs. The code is unnecessarily long and repeated. [1M]	The code lacks reusability. It is huge and repeated at many places[0M]
3)	Post Lab Questions [2]	Answers to all questions are correct and explained in depth. [2M]	Answers to most of the questions are correct but not explained in much depth. {2-1.5M]	Answers of few questions are incorrect and lacks sufficient depth [0-1M]	Answers to most of the questions are incorrect and not explained in depth. [0 mark]
4)	Promptness [1]	The laboratory report is submitted on time [1 mark]	The laboratory report is submitted next day. [0.5 marks]	The laboratory report is submitted in next practical session. [0 marks]	

Rubrics for Assignments:

Indicator	Excellent	Good	Below average
Timeline (2)	submitted on time or early (2)	Submitted next day (1)	Submitted in same week (0.5)
Organization (2)	Well organized, neat and clear handwriting, neat diagrams with all labels.(2)	Organized to some extent, diagrams and handwriting is neat with some missing labels(1)	Poorly organized, diagrams incomplete (0.5)
Level of content (3)	All points are covered(3) and answered accurately	Some important points are omitted / addressed minimally (1-2)	Many important points are missing and the answers are not accurate. (1-0)
Knowledge about the topic (3)	All Concepts of a topic are clear and knows the application to real world problems (3)	All Concepts of a topic are mostly clear lacks understanding about the application to real world problems (2-1)	Poor understanding of concepts and application to real world problems.(1-0)

Lesson Plan

Module 1: Introduction to Analysis of Algorithms				
Lecture No.	Date		Topic	Content Delivery Method
	Planned	Actual		
1	6/1/2020	6/1/2020	Introduction to analysis of algorithms: Introduction to subject and fundamentals of algorithms. What is meant by efficient algorithm?	Chalk and board
2	7/1/2020	7/1/2020	Efficiency of algorithms, Time and Space Complexities Fundamentals	Chalk and board
3	8/1/2020	7/1/2020	Growth of Function – Big O, Omega, Theta	Chalk and board
4	10/1/2020	8/1/2020	Calculation of time complexity for code samples	Chalk and board
5	11/1/2020	9/1/2020	Calculation of time complexity for code samples continued	Chalk and board
6	12/1/2020	9/1/2020	Finding space complexity for code samples	Chalk and board
7	13/1/2020	lab explaination	Finding Complexities of Bubble, Insertion & Selection Sort & Linear Search	Chalk and board, Lab performance
8	16/1/2020	13/1/2020	Recurrences: Solving recurrence using Iteration Method	Chalk and board
9	17/1/2020	14/1/2020	Solving recurrence using Recursion Tree	Chalk and board
10	21/1/2020	22/1/2020	Solving recurrence using Master Method	Chalk and board
11	22/1/2020	23/1/2020	**Divide and Conquer Approach:** General Method of Divide & Conquer, Analysis of Binary Search	Chalk and board, simulation
12	23/1/2020	lab expln	Analysis of Merge Sort and quick sort	Chalk and board, Lab performance, animation
13	24/1/2020	24/1/20	Minmax algorithm	Chalk and board, Lab performance
14	28/1/2020	28/1/20	Strassen's matrix multiplication	Chalk and board

Module 3: Greedy Method Approach:

15	29/1/2020	29/1/2020	Greedy General Method, Knapsack Problem	Chalk and board, Lab performance
16	30/1/2020	30/1/2020	Job Sequencing with deadline	Chalk and board
17	31/1/2020	31/1/2020	SSSP (Dijkstra's Algo)	Chalk and board, visualization, Lab performance
18	4/2/2020	adjusted with Math Proof. (6/2/20)	MST- Prims	Chalk and board, visualization, Lab performance
19	5/2/2020	(6/2/20)	MST – Kruskal	Chalk and board, Lab performance, visualization
20	6/2/2020	30/1/2020	Optimal Storage on tapes	Chalk and board

Module 2: Dynamic programming Approach:

21	7/2/2020	7/2/20	General Method, 0/1 Knapsack	Chalk and board, Lab performance
22	11/2/2020		Single Source Shortest Path	Chalk and board, Lab performance, visualization
23	12/2/2020		All pair shortest Path	Chalk and board, Lab performance, visualization
24	13/2/2020		MultiStage Graph	Chalk and board
25	14/2/2020		Travelling Salesman Problem	Chalk and board, visualization
26	25/2/2020		Longest common subsequence	Chalk and board, Lab performance
27	26/2/2020		Assembly line schedulling	Chalk and board, Lab performance

Module 4: **Backtracking and branch and bound**				
28	3/3/2020		General Method of backtracking, n queen problem	Chalk and board , Lab performance
29	4/3/2020		Sum of Subsets	Chalk and board , Lab performance
30	5/3/2020		Graph Coloring	Chalk and board , Lab performance
31	6/3/2020		General Method of branch and bound, 15 puzzle problem	Chalk and board
32	11/3/2020		Travelling Salesman Problem	Chalk and board
Module 5: **String Matching algorithms**				
33	12/3/2020		Naïve String Maching	Chalk and board
34	13/3/2020		Rabin Karp Algo	Chalk and board
35	17/3/2020		KMP Algo	Chalk and board
36	18/3/2020		String matching with Finite Automata	Chalk and board
Module 6: **Non Deterministic Polynomial algorithms**				
37	19/3/2020		Polynomial time ,Polynomial time verification	Chalk and board , handouts
38	20/3/2020		NP completeness and reducibility	Chalk and board , handouts
39	24/3/2020		Vertex cover problems, Clique Problem	Chalk and board, handouts
40	26/3/2020		**Multiplying long integers(divide and Conquer(Content Beyond Syllabus)**	**Chalk and board, handouts**
41	27/3/2020		**Optimal binary search tree(dynamic programming (Content Beyond Syllabus)**	

LAB PLAN

Sr. No.	TITLE	Mapped Co	Planned Week	Actual dates Batch A	Actual dates Batch B	Actual dates Batch C	Actual dates Batch D
1	WAP to implement Modified bubble sort, Insertion sort, Selection sort and derive its complexity.	CO1 and CO4	1 st week	20/1/20	22/1/20	24/1/20	20/1/20
2	WAP to implement Liner search and binary search and derive its time complexity.	CO1 and CO4	1 st week	20/1/20	22/1/20	24/1/20	20/1/20
3	WAP to implement Quick sort, randomized quick sort, merge sort and derive its complexity.	CO1 and CO4	2nd week	3/2/20	29/1/20	31/1/20	3/2/20
4	WAP to implement min max algorithm.	CO2 andCO4	2nd week	3/2/20	29/1/20	31/1/20	3/2/20.
5	WAP to implement fractional knapsack using greedy method.	CO2 and CO4	3rd week	10/2/20	5/2/20	10/2/20	24/2/20
6	WAP to implement Dijkstra's algorithm.	CO2 and CO4	3rd week				
7	WAP to implement Prim's algorithm	CO2 and CO4	4th week				
8	WAP to implement 0/1 knapsack using dynamic programming.	CO2 and CO4	4th week				
9	WAP to implement Floyd Warshall algorithm.	CO2 and CO4	5th week				
10	WAP to implement bellman ford algorithm.	CO2 and CO4	5th week				

11	WAP to implement N queen problem using backtracking approach.	CO2 and CO4	6th week				
12	WAP to implement sum of subset problem using backtracking approach	CO2 and CO4	6th week				
13	WAP to implement graph coloring using backtracking approach.	CO2 and CO4	7th week				
14	WAP to implement Longest common subsequence.	CO2 and CO4	7th week				
15	WAP to implement Knuth Morris Pratt Algorithm	CO3 and CO4	8th week				
16	WAP to implement Assembly Line scheduling	CO2 and CO4	8th week				