FMTH0301/Rev.5.3

Course Plan

Semester: III Year: 2025-2026

Course Title: Design and Analysis of Algorithms	Course Code: 24ECAC203
Total Contact Hours: 60	Duration of ESA: 03 hrs
ISA Marks: 50	ESA Marks: 50
Lesson Plan Authors: Mr . Pradeep Shet	Date: 09 Sept 2025
Checked By: Dr. Narayan D.G	Date: 13 Sept 2025

Prerequisites:

Student must have knowledge of Problem Solving, Data Structures and Programming skills.

Course Outcomes (COs):
At the end of the course the student should be able to:

1. Theorize and reflect on the properties, operations and applications of design and analysis of algorithms using application case studies.
2. Compare various array queries, sorting, searching, and graph algorithms and perform efficiency analysis.
3. Analyze the role of data structures in structuring and manipulating data for problem-solving and evaluate on their time and space efficiency.
4. Apply algorithmic problem-solving techniques by understanding the theoretical foundations in solving a problem scenario

Course Articulation Matrix: Mapping of Course Outcomes (COs) with Program Outcomes (POs)

Course Title: Design and Analysis of Algorithms	Semester: III
Course Code: 24ECAC203	Year: 2025-26

Course Outcomes (COs) / Program Outcomes (POs)	1	2	3	4	5	6	7	8	9	10	11	12	13
1. Theorize and reflect on the properties, operations and applications of Design and Analysis of Algorithms using application case studies.	H	M	M										
2. Compare various array queries, sorting, searching, and graph algorithms and perform efficiency analysis.	H	M	M										
3. Analyze the role of data structures in structuring and manipulating data for problem-solving and evaluate on their time and space efficiency.	H	M	M										
4. Apply algorithmic problem-solving techniques by understanding the theoretical foundations in solving a problem scenario.	H	M	M										

Degree of compliance **L**: Low **M**: Medium **H**: High

Competency addressed in the Course and corresponding Performance Indicators

Competency	Performance Indicators	Course Specific Performance Indicators (CSPI)
1.1 Demonstrate competence in mathematics	1.1.3 Apply probability and statistics to model and analyze data for informed decision making	1.1.3.1 Apply probability and statistics to model and analyze data structure and algorithmic problems for informed decision making.
	1.1.3 Apply concepts of mathematical structures to solve complex systems in engineering	1.1.3.1 Use summations, series and set up recurrence relations and solve it for analyzing algorithmic efficiency for iterative and recursive problems.
		1.1.3.2 Apply tree and graph principles for structured and un-structured data management.
		1.1.3.3 Apply mathematical structures to solve sorting, searching and graph problems
1.3 Demonstrate competence in engineering fundamentals	1.3.2 Apply the principles of computer science including algorithms, data structures and programming to develop a solution for complex engineering problems	1.3.2.1 Apply object oriented programming concepts and principles to solve data structure and algorithm design problems
		1.3.2.2 Apply principles of computer science algorithms, data structures and programming to develop solution for algorithmic real-world scenarios
2.1 Demonstrate an ability to identify and define complex engineering problems by understanding real-world contexts and constraints.	2.1.2 Identify critical systems/sub systems, variables, and constraints that characterize the problem environment by analyzing practical contexts and real-world limitations.	2.1.2.1 Identify critical systems/sub systems that characterize the problem environment by analyzing data structures, algorithm design techniques and real-world limitations..
2.2 Demonstrate an ability to apply principles of mathematics, natural sciences, and engineering to formulate and analyze complex problems, incorporating literature review to support sustainable development.	2.2.1 Decompose complex problems into interconnected sub-problems.	2.2.1.1 Decompose algorithmic problems into interconnected sub-problems.
3.2 Demonstrate an ability to generate a diverse set of alternative design solutions.	3.2.3 Identify suitable criteria for the evaluation of alternate design solutions	3.2.3.1 Identify space and time efficiency in evaluating design techniques

Course Content

Course Code: 24ECAC203	Course Title: Design and Analysis of Algorithms	
L-T-P : 4-0-0	Credits: 04	Theory: 4hrs/week
ISA Marks: 50	ESA Marks: 50	Total Marks: 100
Teaching Hrs: 60		Exam Duration: 03 hrs

Clusters	Hrs
Cluster 1: Foundations Design Philosophy and Intuitions, Space and Time Complexities, Order of an Algorithm, Problem Patterns: Recursion, Iteration, Backtracking	06
Cluster 2: Computing Principles and Tools Pruning, Edge Relaxation, Sets, Traversals, Prefix and Suffix, Union-Find, Hashing and Other Principles	04
Cluster 3: Structured Data Management Graphs and Trees, Tries, AVL Trees, 2-3 Trees, Red-Black Trees, DFS, BFS, Heap, Array Query, Sparse Table, Segment Trees, Fenwick Trees, Skip Lists	15
Cluster 4: Sorting and Searching Sorting and Searching Devices	10
Cluster 5: Graph Algorithms Shortest Path and Spanning Trees	15
Cluster 6: Problem Assortments Problem types, Undecidability, Limitations of Algorithm Power	10

Text Books

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithms, Fourth Edition, The MIT Press, 2022.
2. Anany V. Levitin, Introduction to the Design and Analysis of Algorithms. Addison-Wesley Longman Publishing Co, 2012.

References

1. Hemant Jain, Problem Solving Using Data and Algorithms Using C, Taran Technologies Private Limited, 2016.

2. HackerRank / CodeChef / SPOJ

Evaluation Scheme
ISA Scheme

Assessment	Weightage in Marks
ISA 1	15
ISA 2	15
Self-Learning Component	20
ESA	50
Total	100

Self-Learning Component:

Sl. No.	Activity	Objective	Hours	Assessment	Weightage (Marks)	Deliverables/ Evidences
1	Liminality Case Studies 16 case studies hosted at: https://algorithmsportfolio.wordpress.com/	To connect algorithm design techniques to natural phenomenon inspired by social order	30	Hand-written submissions of case study	50%	File submissions made by students
2	Case Study Assignments	To realize the real time applications of the course concepts	30	Class test on case study reflections	50%	Case study assignment and test sheets

Course Unitization for ISA

Topics / Chapters	Teaching Hours	No. of Questions in ISA 1	No. of Questions in ISA 2	No. of Questions in ESA
Cluster 1: Foundations	6	0.5	-	0.5
Cluster 2: Computing Principles and Tools	4	0.5	-	0.5
Cluster 3: Structured Data Management	10	2	-	2
Cluster 4: Sorting and Searching	8	-	1.5	1.5
Cluster 5: Graph Algorithms	12	-	2.5	2.5
Cluster 6: Problem Assortments	10	-	-	2

Note

1. Each Question carries 20 marks and will consist of 3 sub questions of 4, 6, and 10 marks each
2. There will be mixing of questions from Cluster 1, 2 and 3 for ISA1 and 4 and 5 for ISA2
3. Answer any 2 full questions out of 3 for each ISA and Cluster 6 is equivalent to UNIT III, evaluated in ESA (Question 7 and 8)

Date: 13 Sept. 2025 **HOD CSE**

Course Assessment Plan

Course Title: Design and Analysis of Algorithms					Code: 24ECAC203
Course outcomes (COs)	Weightage in assessment	Assessment Methods			
		ISA-1	**ISA-2**	**Self-Study**	**ESA**
1. Theorize and reflect on the properties, operations and applications of Design and Analysis of Algorithms using application case studies.	25%	✓	✓	✓	✓
2. Analyze, interpret and compare various array query, sorting, searching, and graph algorithms and perform efficiency analysis.	25%	✓	✓	✓	✓
3. Explain causes and analyze the role of data structures in structuring and manipulating data for problem-solving and evaluate on their time and space efficiency.	25%	✓	✓	✓	✓
4. Apply algorithmic problem-solving techniques by understanding the theoretical foundations in solving a problem scenario.	25%	✓	✓	✓	✓
Weightage		20%	20%	10%	50%

Rubrics for self-learning component:

Criteria	Excellent(5)	Very Good(4)	Good(3)	Average(2)	Poor(1)	CO	PI	Level
1. Identification of Data Structures and Algorithms	Accurately identifies and uses appropriate data structures and algorithms, demonstrating deep understanding.	Identifies and uses suitable data structures and algorithms with good understanding.	Identifies and uses data structures and algorithms with basic understanding, some errors possible.	Identifies and uses data structures and algorithms with limited accuracy or understanding.	Incorrectly identifies or uses data structures and algorithms, demonstrating minimal understanding.	2	1.3.2	L3
2.Using Principles and Generalizing	Applies principles effectively, generalizes solutions to a wide range of problems with clear logic and relevance.	Applies principles well and generalizes solutions to similar problems with good logic and explanation.	Applies principles with basic generalization, but may not cover all relevant scenarios thoroughly.	Applies principles minimally with limited generalization and logical coverage.	Fails to apply principles effectively or generalize solutions, with unclear or irrelevant logic.	1	2.1.2	L3
3.Efficiency Analysis	Provides a comprehensive and accurate analysis of time and space efficiency, demonstrating deep insight.	Provides a clear and accurate analysis of time and space efficiency with good understanding.	Provides a basic analysis of efficiency with some understanding, but may lack depth.	Provides minimal analysis of efficiency with limited detail and understanding.	Provides unclear or incorrect analysis of efficiency, with little understanding.	3	1.1.3	L3
4.Transfer of Knowledge	Demonstrates exceptional ability to transfer and apply knowledge to new problems or contexts, showing insight and adaptability.	Effectively transfers and applies knowledge to new problems with good understanding and adaptability.	Transfers and applies knowledge with basic effectiveness, showing some adaptability.	Transfers and applies knowledge with limited effectiveness and minimal adaptability.	Fails to transfer or apply knowledge effectively to new problems, showing minimal adaptability.	4	1.3.2	L3

Note:

- Criteria 2 and 4 are applicable for liminality case studies
- AI 4 are applicable for case study assignments

Chapter wise Plan

Course Code and Title: 24ECAC203 and Design and Analysis of Algorithms	
Chapter Number and Title: **1 Foundations**	Planned Hours: 6hrs

Learning Outcomes:-

At the end of the topic the student should be able to:

Topic Learning Outcomes	COs	BL	CA Code
Understand the Design Philosophy and Intuitions	CO1	L2	1.3
Analyze the Space and Time Complexities	CO2	L3	1.1
Explain the Order of an algorithm	CO1	L3	1.1
Realize the Problem Patterns: Recursion, Iteration, Backtracking	CO3	L3	1.3

Lesson Schedule Class No. - Portion covered per hour
1. Design Philosophy and Intuitions
2. Space and Time Complexities, Order of an algorithm
3. Efficiency Analysis Problems
4. Efficiency Analysis Problems Continued
5. Efficiency Analysis Problems Continued
6. Problem patterns: Recursion, Iteration, Backtracking

Review Questions

Sl.No. - Questions	TLOs	BL	PI Code
1. Differentiate between iteration and recursion problem solving approaches.	TLO4	L2	1.3.2

2. Consider the following algorithm for computing the sum of the first n cubes:
$S(n) = 1^3 + 2^3 + \ldots + n^3$.
 	
ALGORITHM $S(n)$
//Input: A positive integer n
//Output: The sum of the first n cubes
if $n = 1$ return 1
else return $S(n-1) + n * n * n$
a. Set up and solve a recurrence relation for the number of times the algorithm's basic operation is executed.
b. How does this algorithm compare with the straightforward non-recursive algorithm for computing this sum?
	TLO2	L3	1.1.3

3. Write a note on Asymptotic notations used to compare and rank the order of growth of algorithms.	TLO2	L3	1.3.2
4. An iterative algorithm was found to have the given below Basic	TLO4	L3	1.1.3

Operation count. What is the order of growth? $\sum_{i=1}^{n} \sum_{j=1}^{n} ij$			
5. Discuss how the backtracking algorithmic technique solves optimization problems? Are there any constraints while applying this technique?	TLO4	L2	1.3.2

Course Code and Title: 24ECAC203 and Design and Analysis of Algorithms	
Chapter Number and Title: **2. Computing Principles and Tools**	Planned Hours: 4 hrs

Learning Outcomes:-

At the end of the topic the student should be able to:

Topic Learning Outcomes	COs	BL	CA Code
Illustrate the Computing Principles	CO1	L2	1.3
Perform the Efficiency Analysis	CO2	L3	1.3
Realize Hashing as a Computing Tool	CO3	L3	1.3

Lesson Schedule Class No. - Portion covered per hour / per Class
1. Computing Principles
2. Principles Continued
3. Efficiency Analysis Problems
4. Hashing

Review Questions

Sl.No. - Questions	TLOs	BL	PI Code
1.Consider the following array of strings: ["apple", "applesauce", "app", "application"] What is the longest common prefix among these strings? A) "app" B) "appl" C) "apple" D) "apples"	TLO1	L3	1.3.2
2.Given a string "abracadabra", what is the longest common suffix of the substrings "cadabra" and "dabra"? A) "cadabra" B) "abra" C) "dabra" D) "a"	TLO1	L3	1.3.2
3.Consider a set of 6 elements: {0, 1, 2, 3, 4, 5}. Initially, each element is in a separate set. The following union operations are performed: 1. Union(0, 1) 2. Union(1, 2)	TLO1	L3	1.3.2

3. Union(3, 4) 4. Union(4, 5) 5. Union(1, 3) What is the final set of elements after all these operations? A) {0, 1, 2, 3, 4, 5} B) {0, 1, 2, 3, 4} C) {0, 1, 2} D) {0, 3, 4, 5}			
4.Consider a set of 7 elements: {0, 1, 2, 3, 4, 5, 6}. Initially, each element is in a separate set. The following union operations are performed: 1. Union(0, 1) 2. Union(1, 2) 3. Union(2, 3) 4. Union(4, 5) 5. Union(5, 6) 6. Union(3, 6) What is the final parent of the element 3? A) 0 B) 1 C) 2 D) 6	TLO2	L3	1.3.2
5. What is Pruning? List the real-world applications of pruning technique.	TLO2	L2	1.3.2
6.The keys 12, 18, 13, 2, 3, 23, 5 and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function h(k) = k mod 10 and linear probing. What is the resultant hash table?	TLO3	L3	1.3.2
7.A hash function h defined h(key)=key mod 7, with linear probing, is used to insert the keys 44, 45, 79, 55, 91, 18, 63 into a table indexed from 0 to 6. What will be the location of key 18 and also find the number of probes required to insert the key 18.	TLO3	L3	1.3.2

Course Code and Title: 24ECAC203 and Design and Analysis of Algorithms	
Chapter Number and Title: **3. Structured Data Management**	Planned Hours: 15hrs

Learning Outcomes:-

At the end of the topic the student should be able to:

Topic Learning Outcomes	COs	BL	CA Code
Analyze the Graphs and Trees, Properties	CO2	L3	1.1
Analyze the Graph and Tree Traversals	CO2	L3	1.1
Explain and Implement the Tree Variants	CO3	L3	1.1
Describe the need and properties of Array Query Algorithms	CO4	L3	2.1
Explain Other Advanced DS as Tools	CO3	L3	3.2

Lesson Schedule Class No. - Portion covered per hour
1.Graphs, Trees
2.Trees and Tree Traversals
3-4. BST
5. AVL Trees
6. 2-3 Trees
7. Tries
8. Segment Trees
9. Skip Lists
10. Heap
11. Array Query, Sparse Tables
12. Red Black Trees
13. Fenwick Trees
14. DFS
15. BFS

Review Questions

Sl.No. - Questions	TLOs	BL	PI Code
1. What is a self-balanced tree?	TLO1	L2	1.3.2
2. Explain the properties of a binary tree	TLO2	L2	1.3.2
3. How does computer represent the graphs and binary trees?	TLO3	L2	1.3.2
4. Construct an AVL tree for the list 5,6,8,3,2,4,7 by successive insertions. Explain four rotation types used in the construction of AVL tree.	TLO3	L3	1.3.2
5. Explain the logic behind the fenwick tree construction	TLO3	L2	2.1.2
6. What is the design technique behind segment tree?	TLO4	L2	1.3.2
7.Write the applications of Red-Black Trees	TLO3	L2	1.3.2
8. Comment on the time efficiency of Heap	TLO3	L3	1.1.3
9. Differentiate between DFS and BFS	TLO2	L2	1.3.2
10.Give an algorithm for constructing a binary search tree. While constructing the tree take care that duplicate values eliminated. Trace the algorithm on 8, 13, 10, 12, 6, 9, 5, 2	TLO2	L3	3.2.3

Course Code and Title: 24ECAC203 Design and Analysis of Algorithms	
Chapter Number and Title: **4.Sorting and Searching**	Planned Hours: 10hrs

Learning Outcomes:-

At the end of the topic the student should be able to:

Topic Learning Outcomes	COs	BL	CA Code
1. Describe the different sorting algorithms and perform efficiency analysis	CO2	L3	2.1
2. Describe the different searching algorithms and perform efficiency analysis	CO2	L3	2.1

Lesson Schedule Class No. - Portion covered per hour
1.Sorting,Bubble Sort
2.Selection Sort, Insertion Sort
3.Merge Sort
4.Quick Sort
5.Heap Sort
6. Brute Force String Search, Boyer-Moore Algorithm
7. KMP Algorithm
8. Rabin-Karp Algorithm
9-10. Case Studies

Review Questions

Sl.No. - Questions	TLOs	BL	PI Code
1.Which of the following sort algorithms, has execution time that is least dependent on initial ordering of the input? Justify? A. Selection sort B. Merge sort	TLO1	L3	1.3.2
2.Given two sorted list of size m and n respectively. The number of comparisons needed in the worst case by the merge sort algorithm will be?	TLO1	L3	1.3.2
3.The number of swapping needed to sort the numbers 8 , 22, 7, 9, 31, 5, 13 in ascending order using bubble sort is ?	TLO1	L3	1.3.2
4.Given a list of words followed by two words, the task is to find the minimum distance between the given two words in the list of words. Examples: Input: S = { “the”, “quick”, “brown”, “fox”, “quick”}, word1 = “the”, word2 = “fox” Output: 3 Explanation: Minimum distance between the words “the” and “fox” is 3	TLO1	L2	1.3.2
5. What is the efficiency of Rabin-Karp Algorithm?	TLO2	L2	1.1.3

6. Explain the three different tables constructed in string search algorithms.	TLO2	L3	1.3.2

‘

Course Code and Title: 24ECAC203 Design and Analysis of Algorithms	
Chapter Number and Title: **5. Graph Algorithms**	Planned Hours: 15 hrs

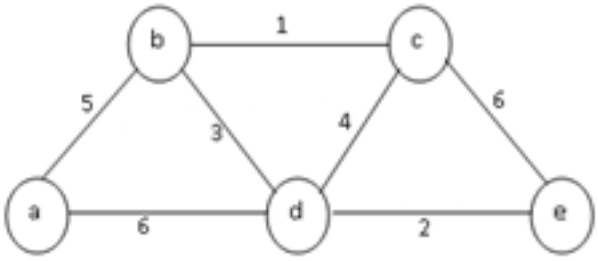
Learning Outcomes:-

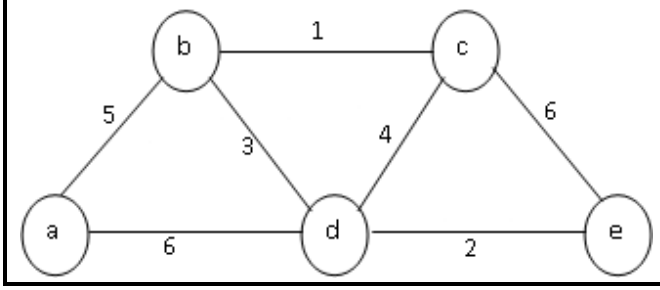
At the end of the topic the student should be able to:

Topic Learning Outcomes	COs	BL	CA Code
Define and describe the shortest path algorithms	CO2	L3	3.2
Describe the minimum spanning tree algorithms	CO2	L3	3.2

Lesson Schedule Class No. - Portion covered per hour
1. Floyd and Warshall’s Algorithm
2-3. Shortest Path algorithms - Dijkstra’s
4-5. Shortest Path algorithms – Bellman-Ford
6. Minimum Spanning Tree Algorithms - Prims
7. Minimum Spanning Tree Algorithms – Kruskal
8-15. Graph Case Studies

Review Questions

Sl.No. - Questions	TLOs	BL	PI Code
1. What is a spanning tree? Design a greedy algorithm to compute a spanning tree for a given graph using Prim’s algorithm. Comment on the efficiency of the same.	TLO1	L3	1.3.2
2. Define single source shortest paths problem. Design a greedy algorithm for single source shortest paths problem. Write its time complexity. 	TLO1	L3	1.3.2
3. Apply Floyd’s algorithm to solve all pair shortest path for the diagraph with the weight matrix given below. Draw the digraph for the weight matrix.	TLO1	L3	1.3.2

R/C | 1 | 2 | 3 1 | 0 | 4 | 11 2 | 6 | 0 | 2 3 | 3 | ∞ | 0			
4.Apply Kruskal's algorithm to find a minimum spanning tree of the following graph. And also give the Kruskal's algorithm. How is the result different from Prim's algorithm? 	TLO2	L3	3.2.3

Course Code and Title: 24ECAC203 Design and Analysis of Algorithms	
Chapter Number and Title: **6. Problem Case Studies**	Planned Hours: 10 hrs

Learning Outcomes:-

At the end of the topic the student should be able to:

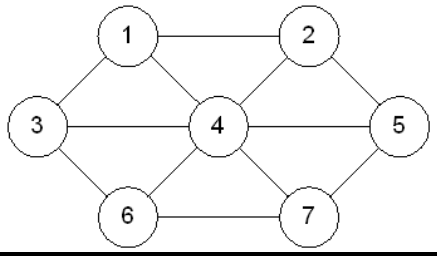
Topic Learning Outcomes	COs	BL	CA Code
Solve the different problem types and explain their limitations	CO4	L3	1.3
Explain undecidability	CO1	L3	1.3
Compare and comprehend N and different NP classes	CO2	L3	3.2

Lesson Schedule Class No. - Portion covered per hour
1. Travelling Salesperson Problem,
2. Knapsack Problem
3. Strassen's Matrix Multiplication
4. Fake Coin Problem
5. Huffman Coding
6. Application Case Studies
7. Undecidability
8. P, NP and classes
9. Case Studies

10. Case Studies – NP type

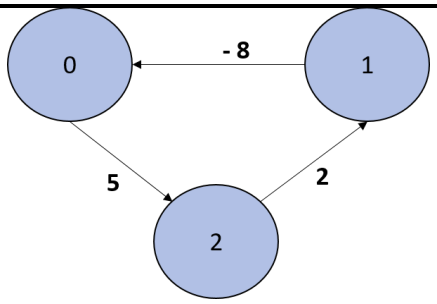
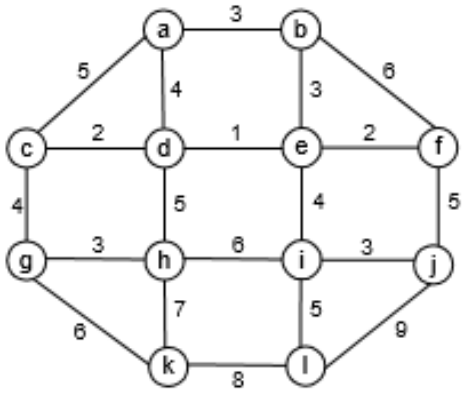
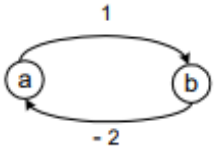
Review Questions

Sl.No. - Questions	TLOs	BL	PI Code
1. What is a Hamiltonian Circuit? How does it map to TSP?	TLO1	L2	1.3.2
2. What are memory functions?	TLO2	L1	1.3.2
3. Solve the following instance using Brute force and Dynamic Programming: Items: 1, 2, 3, 4 Weights: 7, 3, 4, 5 Values: 42, 12, 40, 25 The capacity of Knapsack, W = 10	TLO3	L3	1.3.2
4. Explain the applications of Huffman encoding.	TLO2	L2	1.3.2
5. What does halting problem infer?	TLO4	L2	2.1.2
6.Write a short note on the following: - N - NP - NP-Hard - NP-Complete	TLO 4	L2	1.3.2

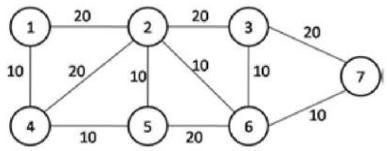
Model Question Paper for ISA-1						
Course Code: 24ECAC203	Course Title: Design and Analysis of Algorithms					
Duration:1hr 15mins	Max. Marks 40					
Note: Answer any TWO FULL questions.						
Q. No	Questions	Marks	CO	BL	PO	PI Code
1 a	Write an algorithm and orders of growth for non-recursive binary search	04	CO3	L3	2	2.1.2
b	Explain how you can use DFS algorithm to find the connected components in the graph.	06	CO1	L3	3	3.2.1
c	Construct an BST, 2-3 Tree and AVL tree for: 1, 2, 3, 4, 5, 6, 7. Compare and contrast between the two.	10	CO3	L4	3	3.2.3
2 a	Answer the following: i) The cities become nodes, and flood effected people need evacuation swiftly. Which method would be optimal: DFS or BFS? ii) The Cambodian character set has 70 alphabets. What's the disadvantage if they opt to trie? iii) The minimum height and number of nodes in a binary tree are connected by log2n. Explain. iv) Perform DFS for the graph below with source as vertex 4: 	04	CO1	L2	1	1.3.2
b	Write an algorithm for chained progressive overflow in hashing.	06	CO4	L3	1	1.1.3
c	Write an algorithm for Towers of Hanoi and perform efficiency analysis.	10	CO4	L4	2	2.1.2
3 a	The keys in the sequence: 12, 18, 13, 2, 3, 23, 5 and 15 are inserted into an initially empty hash table of length 10 using the hash function h(k) = k mod 10 using linear probing/progressive overflow. What is the resultant hash table?	04	CO1	L4	2	2.1.2
b	With appropriate code snippets or pseudo-code,	06	CO3	L3	1	1.1.1

	explain the case of deleting a node with two children from a binary search tree. Draw appropriate diagrams while you present your case.					
c	Consider the given algorithm: ALGORITHM Even(k) Input: k, a positive integer Output: kth even natural number (the first even being 0) Description: Algorithm for finding the kth even natural number if k = 1 return 0; else return Even(k-1) + 2 i) Identify the basic operation and set up a recurrence relation ii) Compute the number of times the basic operation is executed iii) Identify the orders of growth iv) Identify the metric to measure the size of the input v) Trace the algorithm for the input k = 4	10	CO4	L3	2	2.1.2

Model Question Paper for ISA-II						
Course Code: 24ECAC203	Course Title: Design and Analysis of Algorithms					
Duration: 1hr 15mins	Max. Marks : 40					
Note: Answer any TWO FULL questions.						
Q. No	Questions	Marks	CO	BL	PO	PI Code
1 a	Write the recurrence relation for best, worst and average case of Quick sort.	04	CO2	L3	1	1.1.3
b	How is Rabin-Karp algorithm different from BFSS?	06	CO3	L3	1	1.3.2
c	Write a program for heap sort. at?	10	CO3	L3	1	3.2.3
2 a	Apply Bellman-Ford algorithm on given graph with source vertex as 0.	04	CO3	L3	1	1.3.2

						
b	Write Dijkstra's algorithm.	06	CO3	L3	1	1.1.3
c	Generate Minimum Spanning Tree for the Given Graph using Kruskals Algorithm using union-find data structure. 	10	CO3	L3	1	1.3.2
3 a	How many character comparisons will the Boyer-Moore algorithm make in searching for the pattern: 10000 in the binary text of one thousand zeros?	04	CO3	L2	2	2.2.4
b	Apply Floyd's algorithm on the given graph: 	06	CO3	L2	1	1.3.2
c	In insertion sort, we first search for the place using linear search and then insert. Write an algorithm for binary-insertion sort which uses binary search to search for a location and insert. Comment on its efficiency.	10	CO3	L3	2	2.1.2

Model Question Paper for End Semester Assessment (ESA)						
Course Code: 24ECAC203	Course Title: Design and Analysis of Algorithms					
Duration: 03 hours	Max. Marks: 100					
Note: Any two full question from Unit-1 and Unit-2. Any one full question on Unit-3						
UNIT I						
Q. No	Questions	Marks	CO	BL	PO	PI Code
1a	Do you think trees would be a better solution for storing the data? Comment	04	CO3	L3	2	2.1.2
1b	Construct an AVL tree by successive insertions for the list 3,6,5,1,2,4	06	CO1	L3	3	3.2.1
1c	There is a situation, where you are supposed to place N number of Queens on a chess board such that no two queens should be taken off game. Write a program to implement the same by backtracking method.	10	CO3	L4	3	3.2.3
2a	Using the hash function 'key mod 7', insert the following sequence of keys in the hash table- 50, 700, 76, 85, 92, 73 and 101. Explain each step through separate chaining technique for collision resolution.	04	CO1	L2	1	1.3.2
2b	Give the decrease and conquer BFS algorithm. Illustrate its working taking an example.	06	CO4	L3	1	1.1.3
2c	A mathematical student keeps solving expression for most of the day. Now he thinks can there be an automatic way to do it. Show him that yes there is a way by writing a program to implement the evaluation of expression.	10	CO4	L4	2	2.1.2
3a	Discuss detailed differences between recursion and iteration.	04	CO1	L4	2	2.1.2
3b	Using the hash function 'key mod 15', insert the following sequence of keys in the hash table- 50, 720, 176, 825, 962, 77 and 101. Explain stepwise for linear probing and quadratic probing.	06	CO3	L3	1	1.1.1
3c	Write a recursive C function to determine: i) depth of a binary tree. ii) count number of leaf nodes. iii) Search an item in a binary tree.	10	CO4	L3	2	2.1.2
UNIT II						
4a	Explain divide-and-conquer technique .Give the control abstraction for the same	04	CO2	L3	1	1.1.3
4b	Demonstrate the working of bubble sort on the data ={'E','X','A','M','P','L','E'} and analyze the same for time efficiency.	06	CO3	L3	1	1.3.2
4c	Illustrate prim's algorithm to find minimum spanning tree for the below given graph	10	CO3	L3	1	3.2.3

						
5a	To implement Dijkstra's shortest path algorithm on unweighted graphs so that it runs in linear time, the data structure to be used is? Justify your answer	04	CO3	L3	1	1.3.2
5b	Suppose we are given an n-element sequence S such that each element in S represents a different vote in an election, where each vote is given as an integer representing the ID of the chosen candidate. Without making any assumptions about who is running or even how many candidates there are, design an algorithm to see who wins the election S represents, assuming the candidate with the most votes wins.	06	CO3	L3	1	1.1.3
5c	Write program for brute force string matching algorithm	10	CO3	L3	1	1.3.2
6a	Define and Explain minimum spanning tree with example.	04	CO3	L2	2	2.2.4
6b	Apply KMP to match given text and pattern Input: txt[] = "BACDGABCDA" pat[] = "ABCD"	06	CO3	L2	1	1.3.2
6c	Analyze which sorting algorithm is better to sort the following data in increasing order in terms of their time and space complexities: 20,12,8,6,3,1,-2.	10	CO3	L3	2	2.1.2
	UNIT III					
7a	For every Huffman code, the code words of the two least frequent characters have same length. Justify true or false with appropriate reasoning.	4	CO4	L2	2	2.2.3
7b	For a dynamic programming solution of a Knapsack problem, how do you compute the items selected for optimal profit? Present your solution using algorithm.	6	CO4	L3	1	1.3.2
7c	Given a two pan fair balance and N identical looking coins out of which only one coin may be defective. How can we trace which coin, if any, is odd one and also determine whether it is lighter or heavier in minimum number of trials in the worst case? Explain the efficiency analysis.	10	CO4	L3	1	1.1.3
8a	What is an undecidable problem? How is it different from a reducible problem?	4	CO1	L2	1	1.3.2
8b	Explain Halting problem.	6	CO2	L2	2	2.1.2
8c	Explain your inferences of the Halting problem. What does it imply in real world?	10	CO4	L3	1	1.3.2