



Sri Indu

College of Engineering & Technology

UGC Autonomous Institution

Recognized under 2(f) & 12(B) of UGC Act 1956,

NAAC, Approved by AICTE &

Permanently Affiliated to JNTUH



NAAC
NATIONAL ASSESSMENT AND
ACCREDITATION COUNCIL



HANDOUT

COMPUTER NETWORKS

III Year I Semester

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (AIML)

ACADEMIC YEAR 2025-2026



SRIINDUCOLLEGE OFENGINEERING&TECHNOLOGY

(AnAutonomousInstitutionunderUGC,NewDelhi)

(PermanentlyAffiliatedtoJNTUH,ApprovedbyAICTE,New Delhi and AccreditedbyNBA,
NAAC)SherigudaVillage,Ibrahimpatnam Mandal,RangaReddy Dist.–501 510

DEPARTMENTOFCOMPUTERSCIENCE&ENGINEERING(AIML)

HANDOUT-INDEX

S. No	Contents
1	Vision,Mission,PEOs,POs,PSOs &Cos
2	InstitutionAcademicCalendar
3	DepartmentAcademicCalendar
4	Subjectwise
i)	SyllabusCopy
ii)	Lesson Plan
iii)	QuestionBank
iv)	Mid-1&Mid-2Questions(ModelPaper)
v)	EndExaminationQuestions(ModelPaper)



SRIINDUCOLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC, New Delhi)

(Permanently Affiliated to JNTUH, Approved by AICTE, New

Delhi and Accredited by NBA, NAAC) Sheriguda Village, Ibrahimpatnam Mandal, Ranga Reddy Dist. - 501510

VISION OF THE INSTITUTE

To be a premier institution in engineering & technology and management for competency, values and social consciousness.

MISSION OF THE INSTITUTE

IM₁: Provide high quality academic programs, training activities and research facilities.

IM₂: Promote continuous industry-institute interaction aimed at promoting employability, entrepreneurship, leadership and research aptitude among stakeholders.

IM₃: Contribute to the economic and technological development of the region, state and nation.

VISION OF THE DEPARTMENT

To develop competent engineers in the domain of Artificial Intelligence and Machine Learning, Machine Learning for noteworthy contributions to the society.

MISSION OF THE DEPARTMENT

The Department has following Missions:

DM1: To strengthen academic collaborations for better exposure.

DM2: Promote professional environment to imbibe ethical values and entrepreneurial skills

DM3: Encourage research & development by emphasizing on innovation

DM4: Facilitate Industry-Institute collaboration for mutual benefit

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Generate contributions towards advancements in Artificial Intelligence and Machine Learning

PEO2: Promote design, research and implementation of products through strong communication skills, leadership and entrepreneurial skills.

PEO3: Apply basic principles and practices of AIML to successfully complete software related projects to meet customer business objectives and/or productively engage in research. [Typetext]

Program Outcomes (POs)

PO1	Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	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.
PO3	Design/Development of Solutions: Design solutions for complex engineering problems and design systems, components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	Conduct investigation 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.
PO5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO8	Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO9	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO10	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO11	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program specific Outcomes (PSOs)

PSO1	Diagnose the systems reliability in risk and threat environments.
PSO2	Design and enforce computing based solutions according to the cyber security requirements.
PSO3	Identify professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.

COURSE OUTCOMES(CO's)

Academic Year: 2025-26

Class: III YEAR-ISEM.

Course Name: DAA(R22CSE3111)

At the end of the course, the student will be able to

Course Outcomes(COs)	
C224.1	Gain the knowledge of the basic analysis of the performance of algorithms
C224.2	Calculate time complexity and space complexity
C224.3	Ability to analyze the performance of algorithms
C224.4	Ability to choose appropriate data structures
C224.5	Ability to choose algorithm design methods
C224.6	Ability to understand how the algorithm design methods impact the performance of programs

COURSE ARTICULATION MATRIX

CO	PO														
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C224.1	3	1	2	3	-	-	-	-	-	-	3	-	2	3	2
C224.2	3	3	2	1	-	-	-	-	-	-	2	-	3	3	2
C224.3	2	3	2	2	-	-	-	-	-	-	2	-	3	3	2
C224.4	3	3	2	2	-	-	-	-	-	-	1	-	3	2	1
C224.5	3	3	2	2	-	-	-	-	-	-	1	-	3	3	1
C224.6	2	2	2	2							2		3	3	1
C224	2.6	2.5	2.0	2.0	-	-	-	-	-	-	1.8	-	2.8	2.8	1.5

3: High 2. Medium 1. Low

ACADEMIC CALENDAR



SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY
(An Autonomous Institution under UGC, New Delhi)
Recognized under 2(i) and 12(B) of UGC Act 1956
NBA & NAAC Accredited. Approved by AICTE and Permanently affiliated to JNTUH
Sheriguda (V), Ibrahimpatnam, R.R.Dist, Hyderabad - 501 510

D4

BR-22

Lr.No.SICET/AUTO/DAE/III B.Tech Academic Calendar/29/2025

ANNEXURE -II

B.TECH III-YEAR I-SEM & II-SEM ACADEMIC CALENDAR **(FOR ACADEMIC YEAR : 2025-26)**

Academic Calendar for B.Tech – IIIrd Year Students (2023 - 24 Batch), BR-22 Regulation.

I – Semester

S.No.	EVENT	PERIOD	DURATION
1.	Commencement of class work.	14.07.2025 (Monday)	
2.	1 st Spell of Instructions for covering First Two and a half Units.	14.07.2025 – 13.09.2025	9 Weeks
3.	CRT Classes	15.09.2025 – 07.10.2025	2 Weeks 2 Days
4.	I Mid Term Examinations.	08.10.2025 – 10.10.2025	3 Days
5.	Submission of I Mid Term Examination Marks.	16.10.2025	
6.	2 nd Spell of Instructions for Remaining Two and a half Units.	13.10.2025 – 13.12.2025	9 Weeks
7.	II Mid Term Examinations.	15.12.2025 – 17.12.2025	3 Days
8.	Preparation Holidays and Practical Examinations	18.12.2025 – 27.12.2025	1 Week
9.	Submission of II Mid Term Examination Marks.	23.12.2025	
10.	I Semester End Examinations.	29.12.2025 – 10.01.2026	2 Weeks
Commencement of Class-Work for III B.Tech - II Semester 19.01.2026			

Note: No of Working/instructional days : 90

II – Semester

S.No.	EVENT	PERIOD	DURATION
1.	Commencement of class work.	19.01.2026 (Monday)	
2.	1 st Spell of Instructions for covering First Two and a half Units.	19.01.2026 – 14.03.2026	8 Weeks
3.	I Mid Term Examinations.	16.03.2026 – 18.03.2026	3 Days
4.	Submission of I Mid Term Examination Marks.	24.03.2026	
5.	2 nd Spell of Instructions for Remaining Two and a half Units.	20.03.2026 – 09.05.2026	7 Weeks
6.	Continuation of remaining 2 nd Spell of Instructions	25.05.2026 – 10.06.2026	2 Weeks
7.	II Mid Term Examinations.	11.06.2026 – 13.06.2026	3 Days
8.	Preparation Holidays and Project Evaluation.	15.06.2026 – 20.06.2026	1 Week
9.	Submission of II Mid Term Examination Marks.	18.06.2026	
10.	II Semester End Examinations	22.06.2026 – 04.07.2026	2 Weeks

Note: No of Working/instructional days : 93

ACE

CE

DIRECTOR

PRINCIPAL

Controller of Examination

Sri Indu College of Engineering & Technology
(An Autonomous Institution Under UGC)
Sheriguda (V), Ibrahimpatnam, R.R. Dist-501510

DIRECTOR

Sri Indu College of Engineering & Technology
(An Autonomous Institution Under UGC)
Sheriguda (V), Ibrahimpatnam, R.R. Dist-501510.

PRINCIPAL

Sri Indu College of Engineering & Technology
(An Autonomous Institution Under UGC)
Sheriguda (V), Ibrahimpatnam, R.R. Dist-501510.

Sign:

Dr. M.V.S.S. Giridhar

Prof. of CEA; JNTUH Nominee

Sign:

Dr. T. Venu Gopal

Prof. of CSE; JNTUH Nominee

Sign:

Dr. D. Ramesh

Prof. of CSE; JNTUH Nominee

(Copy to DAP&E and Copy to all the Dept. Heads)

DEPARTMENT CALENDAR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING(AI&ML) DEPARTMENT CALENDAR – 2025-2026(SEMESTER-I)

SUNDAY				SEPTEMBER				DECEMBER	
MONDAY	JULY		1	IV-I MID I EXAMS				1	IV-I END EXAMS
TUESDAY	DEPARTMENT MEETING FOR IV-I & III-I 2025-2026		2	IV-I MID I EXAMS	OCTOBER			2	IV-I END EXAMS
WEDNESDAY			3	IV-I MID I EXAMS	III-I CRT CLASSESS			3	IV-I END EXAMS
THURSDAY		AUGUST	4	IV-I SPELL 2 CLASSES	GANDHI JAYANTI & DASHARA			4	IV-I END EXAMS 1
FRIDAY		II-I SPELL I CLASSES COMMENCEMENT	5		III-I CRT CLASSESS	NOVEMBER		5	IV-I END EXAMS 2
SATURDAY		DEPARTMENT MEETING	6		III-I CRT CLASSESS	WOKSHOP UNDER AI LUB		6	IV-I END EXAMS 3
SUNDAY			7					7	
MONDAY	IV-I SPELL I CLASSES COMMENCEMENT		8		III-I CRT CLASSESS			8	IV-I END EXAMS 5
TUESDAY			9		III-I CRT CLASSESS			9	IV-I END EXAMS 6
WEDNESDAY			10	IV-I MID I EXAMS SUBMISSION	III-I MID I EXAMS & II-I MID I EXAMS			10	IV-I END EXAMS 7
THURSDAY			11		III-I MID I EXAMS & II-I MID I EXAMS			11	IV-I END EXAMS 8
FRIDAY			12	DEPARTMENT MEETING REVIEW ON IV-I SLOW LEARNERS	III-I MID I EXAMS & II-I MID I EXAMS			12	IV-I END EXAMS 9
SATURDAY	AI CLUB MEETING	WOKSHOP UNDER AI LUB	13	AI CLUB MEETING				13	IV-I END EXAMS 10
SUNDAY			14					14	
MONDAY	III-I SPELL I CLASSES COMMENCEMENT		15	III-I CRT CLASSESS	II-I & III-I SPELL 2 CLASSESS			15	II-I & III-I MID II EXAMS 12
TUESDAY			16	III-I CRT CLASSESS				16	II-I & III-I MID II EXAMS 13
WEDNESDAY			17	III-I CRT CLASSESS				17	II-I & III-I MID II EXAMS 14
THURSDAY			18	III-I CRT CLASSESS	II-I & III-I MID I EMARKS SUBMISSION			18	II-I & III-I PREPARATION & PRACTICAL EXAMS 15
FRIDAY		INDEPENDENCE DAY	19	III-I CRT CLASSESS	DEPARTMENT MEETING REVIEW ON III-I & II-I SLOW LEARNERS			19	II-I & III-I PREPARATION & PRACTICAL EXAMS 16
SATURDAY			20	III-I CRT CLASSESS	AI CLUB MEETING			20	II-I & III-I PREPARATION & PRACTICAL EXAMS 17
SUNDAY			21	BATHUKAMMA				21	
MONDAY	BONALU HOLIDAY		22	III-I CRT CLASSESS	DIWALI			22	II-I & III-I PREPARATION & PRACTICAL EXAMS 19
TUESDAY			23	III-I CRT CLASSESS				23	II-I & III-I MID II MARKS SUBMISSION III-I PREPARATION & PRACTICAL EXAMS 20

DESIGN AND ANALYSIS OF ALGORITHM

SYLLABUS

BR22 – B.TECH. – CSE(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY (An Autonomous Institution under UGC, New Delhi)

B.Tech. – III Year – I Semester

L	T	P	C
3	1	0	4

(R22CSE3111) DESIGN AND ANALYSIS OF ALGORITHM

Course Objectives:

- Introduces the notations for analysis of the performance of algorithms.
- Introduces the data structure disjoint sets.
- Describes major algorithmic techniques (divide-and-conquer, backtracking, dynamic programming, greedy, branch and bound methods) and mention problems for which each technique is appropriate;
- Describes how to evaluate and compare different algorithms using worst-, average-, and best- case analysis.
- Explains the difference between tractable and intractable problems, and introduces the problems that are P, NP and NP complete.

Course Outcomes:

- Ability to analyze the performance of algorithms
- Ability to choose appropriate data structures and algorithm design methods for a specified application
- Ability to understand how the choice of data structures and the algorithm design methods impact the performance of programs

UNIT - I

Introduction: Algorithm, Performance Analysis-Space complexity, Time complexity, Asymptotic Notations- Big oh notation, Omega notation, Theta notation and Little oh notation.
Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Strassen's matrix multiplication.

UNIT - II

Disjoint Sets: Disjoint set operations, union and find algorithms
Backtracking: General method, applications, n-queen's problem, sum of subsets problem, graph coloring

UNIT - III

Dynamic Programming: General method, applications- Optimal binary search trees, 0/1 knapsack problem, All pairs shortest path problem, Traveling sales person problem, Reliability design.

UNIT - IV

Greedy method: General method, applications-Job sequencing with deadlines, knapsack problem, Minimum cost spanning trees, Single source shortest path problem.

UNIT - V

Branch and Bound: General method, applications - Travelling sales person problem, 0/1 knapsack problem - LC Branch and Bound solution, FIFO Branch and Bound solution.

NP-Hard and NP-Complete problems: Basic concepts, non deterministic algorithms, NP - Hard and NP-Complete classes, Cook's theorem.

TEXT BOOK:

1. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahn and Rajasekharan, University Press.

REFERENCE BOOKS:

1. Design and Analysis of algorithms, Aho, Ullman and Hopcroft, Pearson Education.
2. Introduction to Algorithms, second edition, T. H. Cormen, C.E. Leiserson, R. L. Rivest, and C. Stein, PHI Pvt. Ltd / Pearson Education.
3. Algorithm Design: Foundations, Analysis and Internet Examples, M.T. Goodrich and R. Tamassia, John Wiley and sons.

	SRI INDU COLLEGE OF ENGG & TECH LESSONPLAN (Regulation :R22) Department of AIML			Prepared on March-25-2026	
	Sub.Code&Title		R22CSE3112DAA		
	AcademicYear:2025-26		Year/Sem./Section	III-I	
	FacultyName&Designation		K ANJANEYULU , ASST.PROFESSOR		

Unit/Item No.	Topic(s)	Book Reference	Page(s)		Teaching Methodology	Proposed No. of Periods	Actual Date of Handled	CO/RBT
			From	To				
	UNIT-I							
I	Physicallayer					13		
1.1	Algorithm,	T1	23	25	Black board	01		CO1/L2
1.2	Performance Analysis-Space complexity	T1	39	40	Black board	02		CO1/L1
1.3	Performance Analysis-Space complexity	T1	41	50	Black board	01		CO1/L2
1.5	Time complexity	T1	51	63	Black board	01		CO1/L2
1.6	Asymptotic Notations- Big oh notation	T1	63	67	Black board	01		CO1/L1
1.7	Omega notation	T1	67	70	Black board	01		CO1/L2
1.8	Theta notation and Little oh notation.	T1	76	80	Black board	01		CO1/L1
1.9	Divide and conquer: General method, applications-Binary search	T1	115	119	Presentation	01		CO1/L3
1.10	Quick sort, Merge sort	T1	116	119	Presentation	02		CO1/L4
1.11	Strassen’s matrix multiplication	T1	125	129	Black board	02		CO1/L1
	Review	Signature of theHOD/Coordinator						
Unit/Ite mNo.	Topic(s)	Book Reference	Page(s)		Teaching Methodology	Proposed No. of Periods	Actual Date of Handled	CO/RBT
UNIT-II								
II	Backtracking					10		
2.1	Disjoint Sets: Disjoint set operations	T1	214	216	Black board	01		CO/L1
2.2	Disjoint Sets: Disjoint set operations	T1	217	219	Black board	01		CO2/L2
2.3	union and find algorithms	T1	193	195	Blackboard	01		CO2/L1
2.4	Backtracking: General method	T1	381	382	Blackboard	01		CO2/L1
2.5	Backtracking: General method	T1	379	381	Blackboard	02		CO2/L1

	applications							
2.6	n-queen’s problem	T1	494	502	Black board	01		CO2/L2
2.7	sum of subsets problem	T1	494	495	Blackboard	01		CO2/L3
2.8	sum of subsets problem	T1	495	501	Blackboard	01		CO2/L3
2.9	graph coloring	T1	495	501	Blackboard	01		CO2/L3
	Review	Signature of theHOD/Coordinator						
UNIT-III								
III						14		
3.1	Dynamic Programming: General method	T1			Blackboard	01		CO3/L3
3.2	Dynamic Programming: General method	T1	512	514	Presentation	01		CO3/L3
3.3	applications- Optimal binary search trees	T1	516	522	Black board	01		CO3/L1
3.4	applications- Optimal binary search trees	T1	523	526	Presentation	01		CO3/L1
3.5	0/1 knapsack problem	T1	528	530	Black board	01		CO3/L3
3.6	0/1 knapsack problem				Black board	01		CO3/L1
3.7	All pairs shortest path problem	T1	532	536	Black board	01		CO3/L2
3.8	All pairs shortest path problem	T1	539	541	Presentation	01		CO3/L3
3.9	Traveling sales person problem	T1	542	545	Presentation	01		CO3/L1
3.10	Traveling sales person problem	T1	567	572	Black board	01		CO3/L2
3.11	Reliability design	T1	578	582	Black board	02		CO3/L2
3.12	Reliability design	T1	590	592	Black board	01		CO3/L4
	Review	Signature of theHOD/Coordinator						
UNIT-IV								
IV						10		
4.1	Greedy method: General method	T1	737	741	Presentation	01		CO4/L2
4.2	applications-Job sequencing with deadlines	T1	737	741	Presentation	01		CO4/L2
4.3	knapsack problem	T1	743	790	Black board	01		CO4/L5
4.4	Minimum cost spanning trees	T1	743	790	Black board	01		CO4/L1
4.5	Single source shortest path problem	T1	743	790	Black board	01		CO4/L2

	Review	Signature of the HOD/Coordinator	
UNIT-V			
V			11

5.1	Branch and Bound: General method, applications	T1	737	741	Seminar	01		CO5/L2
5.2	Travelling sales person problem, 0/1 knapsack problem	T1	737	741	Black board	01		CO5/L2
5.3	LC Branch and Bound solution, FIFO Branch and Bound solution	T1	743	790	Black board	01		CO5/L5
5.4	NP-Hard and NP-Complete problems	T1	743	790	Black board	02		CO5/L1
5.5	non deterministic algorithms	T1	743	790	Seminar	01		CO5/L2
5.8	NP - Hard and NP-Complete classes	T1	820	832	Black board	01		CO5/L3
5.9	Cook's theorem	T1	876	891	Black board	02		CO5/L1
	Review	Signature of the HOD/Coordinator						

	SRI INDU COLLEGE OF ENGG & TECHNOLOGY			Prepared on Revised: Pages: 12
	QUESTION BANK (Regulation :R22) Department of CSE(AIML)			
	Sub. Code & Title	R22CSE3111/DAA		
	Academic Year: 2025-26	Year/Sem./Section	III B.TECH, I-SEM	
Faculty Name & Designation	Mr. K ANJANEYULU, Assistant Professor			

UNIT – I

OBJECTIVE QUESTIONS MCQS

SL.NO		BT Level	Course Outcome
1C - 1	Which of the following best describes space complexity? A) The amount of time taken by an algorithm to run B) The amount of memory required by an algorithm to run C) The number of steps an algorithm takes to complete D) The size of the input to the algorithm	I	CO1
1C - 2	If an algorithm requires a fixed amount of memory regardless of the input size, its space complexity is: A) $O(1)$ B) $O(n)$ C) $O(\log n)$ D) $O(n^2)$	III	CO1
1C - 3	Which of the following is the Big-O notation for linear time complexity? A) $O(1)$ B) $O(\log n)$ C) $O(n)$ D) $O(n^2)$	I	CO1
1C - 4	Omega notation (Ω) provides a lower bound for the time complexity of an algorithm. Which case does it describe? A) Best-case scenario B) Worst-case scenario C) Average-case scenario D) Amortized-case scenario	I	CO1
1C - 5	What is the time complexity of the binary search algorithm? A) $O(n)$ B) $O(n \log n)$ C) $O(\log n)$ D) $O(1)$	I	CO1
1C - 6	In binary search, the array must be: A) Unsorted B) Sorted in descending order C) Sorted in ascending order D) Partially sorted	I	CO1

1C - 7	What is the worst-case time complexity of Quick Sort? A) $O(n)$ B) $O(n \log n)$ C) $O(n^2)$ D) $O(\log n)$	I	CO1
1C - 8	Merge Sort uses which of the following approaches? A) Greedy algorithm B) Divide and conquer C) Dynamic programming D) Backtracking	II	CO1
1C - 9	What is the time complexity of Merge Sort in the worstcase? A) $O(n)$ B) $O(n \log n)$ C) $O(n^2)$ D) $O(\log n)$	I	CO1
1C - 10	Strassen's algorithm is used for: A) Searching in an array B) Sorting a list of numbers C) Matrix multiplication D) Finding shortest paths in a graph	II	CO1
1C - 11	What is the time complexity of Strassen's matrix multiplication algorithm? A) $O(n^2)$ B) $O(n^3)$ C) $O(n^{2.81})$ D) $O(n \log n)$	I	CO1
1C - 12	In binary search, the array must be: A) Unsorted B) Sorted in descending order C) Sorted in ascending order D) Partially sorted	I	CO1
1C - 13	Which of the following is not a step in the divide and conquer approach? A) Divide the problem into subproblems B) Conquer the subproblems by solving them independently C) Combine the solutions of the subproblems D) Iterate over the entire problem without dividing	I	CO1
1C - 14	Theta notation (Θ) provides: A) An upper bound on time complexity B) A lower bound on time complexity C) Both an upper and a lower bound on time complexity	I	CO1

	D) Neither an upper nor a lower bound on time complexity		
1C - 15	An algorithm has a time complexity of $O(n^2)$. What happens to the running time if the input size is doubled? A) It remains the same B) It doubles C) It quadruples D) It increases logarithmically	II	CO1

FILL IN THE BLANKS

1F – 1	The _____ complexity of an algorithm refers to the amount of memory required by the algorithm to run to completion	I	CO1
1F – 2	An algorithm with space complexity $O(1)$ requires a _____ amount of memory regardless of the input size	II	CO1
1F – 3	The _____ complexity of an algorithm describes the amount of time taken by the algorithm as a function of the input size	I	CO1
1F – 4	If an algorithm's running time is $O(n^2)$, doubling the input size will cause the running time to _____.	III	CO1
1F – 5	Big-O notation is used to describe the _____-case time complexity of an algorithm	I	CO1
1F - 6	An algorithm with time complexity $O(n)$ is said to have _____ time complexity	I	CO1
1F – 7	Omega (Ω) notation provides a _____ bound for the time complexity of an algorithm.	I	CO1
1F - 8	If an algorithm has time complexity $\Omega(n)$, it means the algorithm will take at least _____ steps to complete	I	CO1
1F - 9	Omega (Ω) notation provides a _____ bound for the time complexity of an algorithm	I	CO1
1F – 10	For an algorithm with time complexity $\Theta(n \log n)$, the running time is bounded both above and below by _____.	II	CO1
1F – 11	The first step in the divide and conquer approach is to _____ the problem into smaller subproblems	I	CO1
1F – 12	A key step in the Quick Sort algorithm is finding the _____ element and partitioning the array.	I	CO1
1F – 13	The time complexity of binary search is _____.	I	CO1
1F – 14	Merge Sort follows the _____ and conquer approach	I	CO1
1F – 15	Strassen's algorithm is used for _____ multiplication.	V	CO1

MATCH THE FOLLOWING

1M – 1	1. Space Complexity 2. Time Complexity 3. $O(1)$ 4. $O(n^2)$ 5. $O(\log n)$ a. Constant time complexity b. The amount of memory required by an algorithm c. The amount of time taken by an algorithm as a function of the input size d. Linearithmic time complexity	III	CO1
--------	--	------------	------------

	e. Quadratic time complexity		
1M – 2	1.Big-O Notation 2.Omega (Ω) Notation 3.Theta (Θ) Notation 4.Little-o (o) Notation a. Describes an upper bound on time complexity b. Describes a lower bound on time complexity c. Describes both upper and lower bounds on time complexity d. Describes a strict upper bound on time complexity	III	CO1
1M – 3	1.Binary Search 2.Quick Sort 3.Merge Sort 4.Strassen's Matrix Multiplication a. Finding an element in a sorted array b. Sorting an array with pivot selection and partitioning c. Matrix multiplication with fewer multiplications than traditional methods d. Sorting an array by dividing, sorting, and merging	I	CO1
1M – 4	1.Divide 2.Conquer 3.Combine 4.T(P) a. Solve the subproblems independently b. Merge the solutions of the subproblems c. Split the problem into smaller subproblems d. Time complexity	I	CO1
1M - 5	1.linear time a. $O(\log n)$ 2. Constant tme b. $O(n)$ 3. logarithmic time c. $O(1)$ 4. time Complexity d. $T(P)$	II	CO1

5 – MARKS QUESTIONS

1D - 1	What do you mean by performance analysis of an algorithm? Explain (understanding)	I	CO1
1D - 2	Define algorithm. Explain the characteristics of the algorithm (remembering)	IV	CO1
1D - 3	Find Big-oh notation and Little-oh notation for $f(n) = 7n^3 + 50n^2 + 200$. (evaluating)	VI	CO1
1D - 4	What is Performance analysis ? Define different time complexity notations. (remembering)	II	CO1
1D - 5	Given $f(n) = 20n^3 - 3$, then prove that $f(n) = O(n^3)$.? (evaluating)	III	CO1
1D - 6		IV	CO1

	Explain Binary search algorithm with an example? (understanding)		
1D - 7	Develop general method for Divide and Conquer approach? (applying)	II	CO1
1D - 8	Explain Merge sort algorithm using divide and conquer method.(understanding)	V	CO1
1D - 9	Explain Strassen's Matrix multiplication with example.(understanding)	II	CO1
1D - 10	Explain how quick sort sorts the following sequences of keys in ascending order. 22,55,33,11,99,77,55,66,54,21,32(creating)	I	CO1
1D - 11	Define Divide and Conquer Method(remembering)	III	CO1
1D - 12	What is binary search ?(remembering)	I	CO1
1D - 13	What is sorting?(remembering)	IV	CO1
1D - 14	Analyze time complexity of quick sort?(analyzing)	V	CO1
1D - 15	Compute time complexity of Merge sort.(analyzing)	III	CO1

Course Outcome		BT Level	
CO2	Which of the following data structures is used to implement the Disjoint Set operations? A) Linked List B) Binary Tree C) Hash Table D) Union-Find Structure	I	
2C - 2	What is the primary purpose of the Union-Find algorithm? A) To sort a list of elements B) To find the shortest path in a graph C) To determine which subset a particular element is in D) To balance a binary search tree	I	CO2
2C - 3	What is the time complexity of the Find operation in the Union-Find algorithm with path compression? A) $O(1)$ B) $O(\log n)$ C) $O(n)$ D) $O(\alpha(n))$, where $\alpha(n)$ is the inverse Ackermann function	I	CO2
2C - 4	In the Union-Find algorithm, what is the purpose of the Union by Rank technique? A) To keep track of the number of elements in each set B) To ensure that the tree remains balanced C) To perform path compression efficiently D) To find the root of the tree quickly	II	CO2
2C - 5	Which of the following operations is not a part of the basic operations of the Union-Find structure? A) Find B) Union C) Insert D) MakeSet	I	CO2
2C - 6	Which of the following problems can be solved using the backtracking technique? ○ A) Sorting a list of numbers ○ B) Solving a system of linear equations ○ C) N-Queen's problem ○ D) Calculating the greatest common divisor	II	CO2
2C - 7	Which backtracking technique is used to eliminate the need to explore all possible solutions?	I	CO2

	○ A) Branch and Bound ○ B) Greedy Method ○ C) Dynamic Programming ○ D) Divide and Conquer		
2C - 8	In graph coloring, what is the objective? ○ A) To find the shortest path between two nodes ○ B) To determine if a graph is bipartite ○ C) To color the vertices of a graph such that no two adjacent vertices have the same color ○ D) To find the minimum spanning tree	I	CO2
2C - 9	Which of the following is an example of a constraint satisfaction problem that can be solved using backtracking? ○ A) Matrix multiplication ○ B) Graph traversal ○ C) Sum of subsets problem ○ D) Fibonacci sequence calculation	I	CO2
2C - 10	In the N-Queen's problem, what is the goal? ○ A) To place N queens on a chessboard such that no two queens attack each other ○ B) To find the shortest path for a knight to visit all squares ○ C) To place N rooks on a chessboard such that no two rooks attack each other ○ D) To solve a Sudoku puzzle	III	CO2
2C - 11	What is the worst-case time complexity of the naive Union operation in a Disjoint Set data structure? ○ A) $O(1)$ ○ B) $O(\log n)$ ○ C) $O(n)$ ○ D) $O(n^2)$	I	CO2
2C - 12	Which of the following sequences of operations on Disjoint Sets is most efficient when combined with path compression? ○ A) Multiple find operations followed by a single union operation ○ B) Multiple union operations followed by a single find operation ○ C) Alternating find and union operations	I	CO2

	○ D) None of the above		
2C - 13	In the context of Disjoint Sets, what does the term "rank" refer to? ○ A) The size of the set ○ B) The depth of the tree representing the set ○ C) The number of elements in the set ○ D) The identifier of the set	II	CO2
2C - 14	Which of the following is true about the Union-Find data structure with both path compression and union by rank? A) The amortized time complexity of each operation is $O(\log n)$ B) The amortized time complexity of each operation is $O(n)$ C) The amortized time complexity of each operation is $O(1)$ D) The amortized time complexity of each operation is $O(\alpha(n))$	II	CO2
2C - 15	In the Union-Find algorithm, what does path compression achieve? ○ A) It reduces the height of the trees ○ B) It merges two sets into one ○ C) It finds the representative of a set ○ D) It keeps track of the size of each set	II	CO2
FILL IN THE BLANKS			
2F - 1	The _____ operation in the Union-Find algorithm is used to merge two subsets into a single subset.	I	CO2
2F - 2	The technique used to keep the tree flat in the Union-Find algorithm is known as _____.	II	CO2
2F - 3	The _____ operation is used to determine the root or representative of the subset to which a particular element belongs.	I	CO2
2F - 4	In the Union-Find algorithm, Union by _____ is used to keep the tree balanced.	I	CO2
2F - 5	The Union-Find data structure is also known as the _____ data structure.	I	CO2
2F - 6	The Union-Find algorithm is commonly used in _____ algorithms to detect cycles.	II	CO2
2F - 7	The amortized time complexity of the Find operation with path compression is _____.	I	CO2
2F - 8	In the N-Queen's problem, the objective is to place _____ queens on an $n \times n$ chessboard such that no two queens threaten each other.	I	CO2
2F - 9	The backtracking technique involves constructing a solution _____ by _____.	I	CO2
2F - 10	A problem that can be solved using backtracking typically allows for _____ construction of a solution.	I	CO2
2F - 11	The sum of subsets problem aims to find a subset of numbers that adds up to a given _____.	I	CO2
2F - 12	In graph coloring, the objective is to color the vertices of a graph such that no two _____ vertices have the same color.	III	CO2
2F - 13	The minimum number of colors needed to color a graph is called the	I	CO2

	_____ number		
2F - 14	The backtracking technique can be applied to solve the _____ Path Problem in graph theory.	III	CO2
2F - 15	In the context of backtracking, a _____ search is often used to explore all possible solutions.	I	CO2
MATCH THE FOLLOWING			
2M - 1	1. Union 2. Find 3. Path Compression 4. Union by Rank	A. Used to determine the root or representative of a set B. Merges two subsets into a single subset C. Ensures that the tree representing the set remains flat D. Keeps the tree flat and reduces the time complexity of the find operation	II CO2
2M - 2	1. MakeSet 2. Disjoint Set 3. Inverse Ackermann Function 4. n queen problem	A. Initializes each element in its own set B. Data structure that supports union and find operations C. Used to express the amortized time complexity of the find operation D. Back tracking	I CO2
2M - 3	1. Sum of Subsets 2. Graph Coloring 3. Chromatic Number 4. Hamiltonian Path Problem	A. the minimum no. of colors required . B. back tracking C. Used to explore all possible solutions and backtrack if a solution violates constraints D. Objective is to color the vertices of a graph such that adjacent vertices have the same color	III CO2
2M - 4	1. Union-Find Structure 2. Connected Components 3. Union Operation 4. Find Operation	A. The process of combining two sets into one B. Determines if two elements are in the same subset C. Subsets in which any two vertices are connected by paths D. A collection of trees representing disjoint sets	IV CO2
2M - 5	1. Constraint Satisfaction 2. Branch and Bound 3. Incremental Approach 4. Graph Coloring Problem	A. . Problems where a solution must meet a number of constraints B. Problem of coloring a graph such that no two adjacent vertices share the same color C. Strategy that builds solutions piece by piece, ensuring each piece is valid before proceeding	III CO2

D .Search strategy that systematically explores all branches, using bounds to prune branches

5 – MARKS QUESTIONS

2D - 1	Explain the usefulness of the following fundamental operations on sets (a) FIND (b) UNION (understanding)	I	CO2
2D - 2	Determine weighted union and collapsing find algorithm With an Example.(evaluating)	III	CO2
2D - 3	Write Algorithm for graph coloring problem	V	CO2
2D - 4	What is general method of back tracking? Explain N queen problem	V	CO2
2D - 5	Explain Sum of subset problem with example .(understanding)	IV	CO2
2D - 6	Two sets S1 and S2 are given as below S1= {1,2,4,6} and S2={7,8}(creating) (a) Draw Disjoint sets S1 and S2 using Trees. (b) Draw Disjoint sets S3 such that S3=S1US2 Draw Disjoint sets S4 such that S4=S2US1	II	CO2
2D - 7	Write and explain the find algorithm using collapse rule with an example.(understanding)	I	CO2
2D - 8	Write and explain the UNION algorithm using weighted rule with an example.(understanding)	I	CO2
2D - 9	Explain Disjoint set operations (remembering)	II	CO2
2D - 10	Write pseudo code for union and find Algorithm .(understanding)	V	CO2
2D - 11	Explain General method of back tracking	I	CO2
2D - 12	Write algorithm to solve 8 queen's problem	VI	CO2
2D - 13	Give example for sum of subsets problem	VI	CO2
2D - 14	Define chromatic number? Write algorithm for graph coloring problem using backtracking	I	CO2
2D - 15	Explain Union and find algorithm	V	CO2

OBJECTIVE QUESTIONS
MCQS UNIT -3

SL.NO		BT Level	Course Outcome
3C - 1	What is the primary approach used in dynamic programming to solve problems? A) Greedy Method B) Divide and Conquer C) Breaking problems into simpler subproblems and storing solutions D) Brute Force	I	CO3
3C - 2	Which of the following is NOT a characteristic of problems suitable for dynamic programming? A) Overlapping Subproblems B) Optimal Substructure C) Independent Subproblems D) Memoization	II	CO3
3C - 3	In the context of dynamic programming, what is memoization? A) A technique to divide the problem into independent subproblems B) A method to store the results of expensive function calls and reuse them C) A strategy to sort elements in a list D) A process to merge sorted arrays	I	CO3
3C - 4	Which algorithm is commonly used to solve the all pairs shortest path problem? A) Bellman-Ford Algorithm B) Dijkstra's Algorithm C) Floyd-Warshall Algorithm D) Prim's Algorithm	I	CO3
3C - 5	In the 0/1 knapsack problem, what does the "0/1" signify? A) Each item can be taken any number of times B) Each item can be either taken completely or not taken at all C) Each item has a binary weight D) Each item has a binary value	II	CO3
3C - 6	What is the main goal of the traveling salesperson problem (TSP)? A) To visit each city exactly once and return to the starting city with the minimum distance traveled B) To find the shortest path between two cities C) To color the cities so that no two adjacent cities have the same color D) To maximize the number of cities visited	I	CO3
3C - 7	Which of the following is a key objective in reliability design using dynamic programming? A) Minimizing the total cost B) Maximizing the reliability of a system C) Finding the shortest path in a network D) Sorting elements in ascending order	I	CO3
3C - 8	In an optimal binary search tree, what is the primary criterion for determining the tree structure? A) Minimizing the height of the tree B) Minimizing the expected search cost C) Maximizing the number of nodes	II	CO3

	D) Maximizing the balance factor		
3C - 9	Which dynamic programming algorithm can be used to solve the single-source shortest path problem in a weighted graph with negative weights? A) Dijkstra's Algorithm B) Kruskal's Algorithm C) Floyd-Warshall Algorithm D) Bellman-Ford Algorithm	I	CO4
3C - 10	In dynamic programming, which property allows an optimal solution to be constructed efficiently from optimal solutions to its subproblems? A) Greediness B) Optimal substructure C) Overlapping subproblems D) Exponential complexity	I	CO4
3C - 11	Which of the following problems is NOT typically solved using dynamic programming? A) Fibonacci sequence B) Merge Sort C) Longest Common Subsequence D) Matrix Chain Multiplication	IV	CO4
3C - 12	In the context of dynamic programming, what does the term "overlapping subproblems" mean? A) Subproblems that are completely independent B) Subproblems that share some common subproblems C) Subproblems that do not need to be solved multiple times D) Subproblems that are solved using a divide-and-conquer approach	I	CO4
3C - 13	The 0/1 knapsack problem can be solved using dynamic programming by creating a table that stores the _____. A) Minimum weight of the items B) Maximum value obtainable with a given weight capacity C) Number of items selected D) Average value of the items	II	CO4
3C - 14	In the Floyd-Warshall algorithm, the primary purpose is to find the _____. A) Longest path in a graph B) Minimum spanning tree of a graph C) Shortest paths between all pairs of vertices D) Maximum flow in a network	I	CO4
3C - 15	Which of the following statements about dynamic programming is true? A) It is always more efficient than greedy algorithms B) It solves each subproblem only once and stores the result C) It cannot be applied to optimization problems D) It is the same as recursion	IV	CO4
FILL IN THE BLANKS			
3F - 1	Dynamic programming solves problems by breaking them down into	I	CO3

	simpler _____ and storing their solutions		
3F - 2	The two key properties that a problem must have to be suitable for dynamic programming are optimal substructure and _____ subproblems.	IV	CO3
3F - 3	The _____ technique in dynamic programming involves storing the results of expensive function calls and reusing them	I	CO3
3F - 4	The _____ algorithm is used to solve the all pairs shortest path problem in a weighted graph	IV	CO3
3F - 5	In the 0/1 knapsack problem, each item can be either taken completely or _____ taken at all.	I	CO3
3F - 6	The goal of the traveling salesperson problem (TSP) is to find the shortest possible route that visits each city exactly once and returns to the _____ city	I	CO3
3F - 7	In reliability design using dynamic programming, the objective is to maximize the _____ of the system.	IV	CO3
3F - 8	An optimal binary search tree minimizes the expected _____ cost.	I	CO3
3F - 9	The _____ algorithm can be used to solve the single-source shortest path problem in a weighted graph with negative weights.	II	CO4
3F - 10	The property of dynamic programming that allows an optimal solution to be constructed efficiently from optimal solutions to its subproblems is called _____ substructure.	I	CO4
3F - 11	The table used in the 0/1 knapsack problem stores the _____ value obtainable with a given weight capacity.	I	CO4
3F - 12	The dynamic programming approach to matrix chain multiplication minimizes the total number of _____ multiplications	I	CO4
3F - 13	The _____ problem aims to find the longest subsequence that appears in both given sequences.	I	CO4
3F - 14	The _____ problem involves finding the maximum value of items that can be placed in a knapsack with a given weight capacity.	V	CO4
3F - 15	The Floyd-Warshall algorithm operates on a _____ matrix to find the shortest paths between all pairs of vertices	I	CO4

MATCH THE FOLLOWING

3M – 1	1. Memorization 2. 0/1 Knapsack Problem 3. Optimal Substructure 4. Floyd-Warshall Algorithm	A. A dynamic programming algorithm used to solve the all pairs shortest path problem B. An optimal solution can be constructed from optimal solutions of its subproblems C. Each item can be either taken completely or not taken at all D. Storing the results of expensive function calls and reusing them	I	CO3
3M – 2	1. Optimal Binary search Trees 2. 0/1 knapsack problem 3. All pairs shortest problem	A. Finding the most efficiency way search data B. Maximizing value with a weight constraint C. Determinin the shortest paths between all pairs of nodes		CO3

	4. Travelling sales man Problem	D. Minimizing the total distance traveled to visit all cities once		
3M – 3	1.Principle of optimality 2. spanning tree 3.prims algorithm 4. Dijkstra's Algorithm	A. solves SSSP problem B. Minimum cost spanning tree C.each subsequence must be optimal D. subset of a graph	II	CO3
3M – 4	1. Matrix Chain Multiplication 2.Optimal Binary Search Tree 3.Knapsack Table 4. Reliability Design	A. Problem of finding the most reliable system configuration using components with given reliabilities B. Table used to store maximum values for given weights and items in the knapsack problem C. . Minimizes the expected search cost in a binary search tree D. Minimizes the number of scalar multiplications	I	CO4
3M – 5	1. Optimal Binary Search Tree 2. Overlapping Sub problems 3. Bellman-Ford Algorithm 4. All Pairs Shortest Path	A. Finds the shortest paths between all pairs of vertices in a weighted graph B. Finds the shortest paths from a single source vertex to all other vertices in a graph C. Characteristic of problems where the same sub problems are solved multiple times D. Minimizes the expected search cost in a binary search tree	I	CO4

5 – MARKS QUESTIONS

3D – 1	Explain the general method of Dynamic programming	V	CO3
3D – 2	Implement Optimal binary search trees and dynamic programming	II	CO3
3D – 3	Show the step by step procedure of deriving the minimum cost spanning tree using Kruskal's algorithm.(remembering)	II	CO3
3D – 4	Explain Matrix chain multiplication with an Example.(understanding)	I	CO3
3D – 5	Evaluate 0/1 knapsack problem with dynamic programming	V	CO3
3D – 6	Design optimal binary search tree algorithm with an example.(creating)	II	CO3
3D – 7	Explain All Pairs Shortest Problem and algorithm for solving it	II	CO3
3D – 8	What do you mean by Travelling sales person problem? How can you solve it	IV	CO3
3D - 9	What is reliability? How to design reliability	II	CO4
3D - 10	Write and explain Algorithm for OBST	I	CO4
3D - 11	What is the Significance of 0/1 knapsack problem? how can you solve it	I	CO4
3D - 12	What is APSP problem? Write Algorithm to solve it	II	CO4
3D - 13	What is reliability design problem? How can you solve it	II	CO4

3D - 14	Evaluate TSP problem with an Example	II	CO4
3D – 15	Discuss Advantages and Disadvantages of Dynamic Programming	II	CO4

Unit IV

OBJECTIVE QUESTIONS MCQS			
SL.NO		BT Level	Course Outcome
4C – 1	Which algorithm is commonly used to find the Minimum Cost Spanning Tree of a graph? ○ A) Dijkstra's Algorithm ○ B) Kruskal's Algorithm ○ C) Bellman-Ford Algorithm ○ D) Floyd-Warshall Algorithm	II	CO5
4C – 2	The Fractional Knapsack Problem differs from the 0/1 Knapsack Problem in that: ○ A) Items cannot be divided or taken partially ○ B) Items can be divided into fractions or partial quantities ○ C) It does not use dynamic programming ○ D) It does not involve weights and values	II	CO5
4C – 3	Which of the following algorithms is used to solve the Single Source Shortest Path Problem with non-negative weights? ○ A) Kruskal's Algorithm ○ B) Dijkstra's Algorithm ○ C) Floyd-Warshall Algorithm ○ D) Bellman-Ford Algorithm	I	CO5
4C – 4	In the context of the Greedy Method, which of the following problems can be solved using a greedy algorithm? ○ A) The Traveling Salesperson Problem ○ B) Job Sequencing with Deadlines ○ C) Longest Common Subsequence ○ D) Matrix Chain Multiplication	I	CO5
4C – 5	For which of the following problems is the greedy approach NOT typically used? ○ A) Fractional Knapsack Problem ○ B) Minimum Spanning Tree ○ C) Job Sequencing with Deadlines ○ D) Longest Path in a Graph	I	CO5
4C – 6	Which of the following statements about the Fractional Knapsack Problem is true? ○ A) The problem does not allow fractional items ○ B) The problem involves making choices that are not divisible ○ C) The problem can be solved using a greedy approach by	I	CO5

	taking items with the highest value-to-weight ratio ○ D) The problem is solved using dynamic programming		
4C – 7	Which algorithm is used for finding the minimum spanning tree that grows one edge at a time by adding the smallest edge each time? ○ A) Dijkstra's Algorithm ○ B) Prim's Algorithm ○ C) Bellman-Ford Algorithm ○ D) Floyd-Warshall Algorithm	I	CO5
4C – 8	In the Single Source Shortest Path Problem, which algorithm is suitable for graphs with negative weights? ○ A) Dijkstra's Algorithm ○ B) Kruskal's Algorithm ○ C) Bellman-Ford Algorithm ○ D) Prim's Algorithm	II	CO5
4C – 9	In the Job Sequencing Problem with deadlines, if two jobs have the same deadline, the job with the _____ should be scheduled first. ○ A) Later start time ○ B) Higher profit ○ C) Shorter duration ○ D) Longer duration	I	CO5
4C – 10	Which of the following algorithms is used to solve the Minimum Spanning Tree problem by growing the spanning tree from a starting vertex and adding the smallest weight edge at each step? ○ A) Kruskal's Algorithm ○ B) Prim's Algorithm ○ C) Dijkstra's Algorithm ○ D) Bellman-Ford Algorithm	I	CO5
4C – 11	Which of the following is true about the greedy approach used for solving the Fractional Knapsack Problem? ○ A) It is not applicable if items cannot be divided ○ B) It solves the problem by taking items in order of their weight-to-value ratio ○ C) It solves the problem by taking items in order of their value-to-weight ratio ○ D) It involves using dynamic programming to solve the problem	I	CO5
4C – 12	The time complexity of Kruskal's Algorithm for finding a minimum		CO5

	spanning tree, using a union-find data structure, is: ○ A) $O(E \log E)$ ○ B) $O(V^2)$ ○ C) $O(E \log V)$ ○ D) $O(V \log V)$	I	
4C – 13	Which greedy algorithm is specifically designed to handle problems with fractional weights and values? ○ A) Kruskal's Algorithm ○ B) Prim's Algorithm ○ C) Fractional Knapsack Algorithm ○ D) Dijkstra's Algorithm	II	CO5
4C – 14	The Single Source Shortest Path Problem using Dijkstra's Algorithm requires that the graph has _____ weights. ○ A) Negative ○ B) Positive ○ C) Zero ○ D) Arbitrary	II	CO5
4C – 15	In the Job Sequencing Problem with deadlines, if two jobs have the same deadline, the job with the _____ should be scheduled first. ○ A) Later start time ○ B) Higher profit ○ C) Shorter duration ○ D) Longer duration	I	CO5
FILL IN THE BLANKS			
4F – 1	The _____ method solves problems by making the locally optimal choice at each step, with the hope of finding a global optimum	I	CO5
4F – 2	In the Job Sequencing Problem with deadlines, the goal is to maximize the number of jobs completed by their _____.	I	CO5
4F – 3	The _____ Knapsack Problem allows items to be divided into fractions or partial quantities, and is solved by taking items with the highest value-to-weight ratio first.	II	CO5
4F – 4	Kruskal's Algorithm is used to find the _____ Spanning Tree of a graph by adding edges in increasing order of weight and avoiding cycles.	II	CO5
4F – 5	In the context of the Single Source Shortest Path Problem, Dijkstra's Algorithm is used for graphs with _____ weights.	I	CO5
4F – 6	In the Job Sequencing Problem, if two jobs have the same deadline, the job with the _____ profit should be scheduled first	II	CO5

4F – 7	Prim's Algorithm is a greedy algorithm used to find the Minimum Spanning Tree by expanding the tree from a starting _____ and adding the smallest edge each time	II	CO5
4F – 8	The _____ Knapsack Problem involves making decisions on a set of items to maximize the total value without exceeding a given weight capacity.	I	CO5
4F – 9	For the Minimum Spanning Tree problem, Kruskal's Algorithm works by adding edges in increasing order of _____ and avoiding cycles.	II	CO5
4F – 10	The greedy approach for solving the Fractional Knapsack Problem involves taking items in order of their _____-to-weight ratio.	V	CO5
4F – 11	The time complexity of Dijkstra's Algorithm, when using a priority queue, is _____.	I	CO5
4F – 12	The greedy method is often applied to problems where a _____ choice at each step leads to a globally optimal solution	I	CO5
4F – 13	In Kruskal's Algorithm, edges are added to the Minimum Spanning Tree based on their _____, ensuring that no cycles are formed.	I	CO5
4F – 14	The _____ Knapsack Problem does not allow items to be divided and is solved using dynamic programming or a greedy approach.	II	CO5
4F – 15	In the Single Source Shortest Path Problem, the Dijkstra's Algorithm finds the shortest path from a single source to all other _____ in the graph	III	CO5
MATCH THE FOLLOWING			
4M – 1	- Greedy Method - Job Sequencing with Deadlines - Fractional Knapsack Problem - Minimum Cost Spanning Tree A.. Maximizes the number of jobs completed before their deadlines B. Solves the knapsack problem where items can be divided into fractions or partial quantities C. Finds the minimum spanning tree by adding edges in increasing order of weight and avoiding cycles D. Solves the problem by making the locally optimal choice at each step	I	CO5
4M – 2	- Dijkstra's Algorithm - Kruskal's Algorithm - Prim's Algorithm 0/1 Knapsack Problem A. Solves the problem of finding the minimum spanning tree by adding the smallest edge each time B. Finds the most valuable subset of items that can fit into a knapsack with a given capacity, without	I	CO5

	dividing items C. Technique used to solve the problem of finding the minimum spanning tree in a weighted graph D. Solves the problem of finding the shortest path from a single source		
4M - 3	1.Kruskal's Algorithm 2.0/1 Knapsack Problem 3. Job Sequencing with Deadlines 4.Prim's Algorithm A. A problem where items cannot be divided and must be taken or left entirely B.A greedy algorithm that adds edges with the minimum weight and avoids cycles C. Solves the job sequencing problem by scheduling jobs with the highest profit first D. Solves the problem of finding a minimum spanning tree by expanding from a starting vertex	II	CO5
4M - 4	1.Feasible solution A. at each stage decision is made 2.Optimal function B. satisfies given constraints 3.Objective function C. maximize the value 4.Greedy method D. optimization measure	II	CO5
4M - 5	1.Knapsack problem A. Bag 2.Spanning tree B. collection of nodes connected by edges 3.Knapsack C. subset of graph 4. graph D. resources allocation	III	CO5
5 – MARKS QUESTIONS			
4D – 1	Explain General Method of Greedy Approach	II	CO5
4D – 2	Explain Job sequencing algorithm with an example.(understanding)	II	CO5
4D – 3	Solve the following problem of Job sequencing with the dead line specified using Greedy strategy.(evaluating) N=4, (p1, p2, p3, p4) = (100, 10, 15, 27) (d1, d2, d3, d4)= (2, 1, 2, and 1).	III	CO5
4D – 4	Find the optimal solution for the given instance of knapsack problem.(evaluating) n=7, m=15, (p1, p2, p3, p4, p5, p6, p7)= (10, 5, 15, 7, 6, 8, 3) (w1, w2, w3, w4, w5, w6, w7)= (2, 3, 5, 7, 1, 4, 1) Find the optimal solution for a. Maximum profit. b. Minimum weight.	I	CO5
4D – 5	Explain kruskal's algorithm with an example .(understanding)	II	CO5
4D – 6	Explain minimum cost spanning tree.(understanding)	III	CO5

4D – 7	Show the step by step procedure of deriving the minimum cost spanning tree using Kruskal's algorithm.(remembering)	IV	C05
4D – 8	Evaluate Single source shortest path problem with Example	I	C05
4D – 9	Describe the following a) Feasible solution b) optimal solution	I	C05
4D – 10	Differentiate Prim's and Kruskal 's Algorithms	I	C05
4D – 11	Evaluate the steps in Kruskal's Algorithm	I	C05
4D – 12	Explain Prim's Algorithm	I	C05
4D – 13	Describe the concept of single source shortest path Problem	V	C05
4D – 14	Describe the following a) Minimum cost spanning tree b) Single source shortest path Problem	V	C05
4D – 15	Write Algorithm for greedy Method	I	C05

UNIT – V

**OBJECTIVE QUESTIONS
MCQS**

SL. NO		BT Level	Course Outcome
5C – 1	The Travelling Salesperson Problem (TSP) can be solved using Branch and Bound by: ○ A) Generating all possible permutations of cities ○ B) Using a heuristic to estimate the shortest possible route and pruning paths that exceed this estimate ○ C) Applying a greedy algorithm to select the shortest edge at each step ○ D) Solving the problem by dynamic programming	II	CO6
5C – 2	FIFO Branch and Bound solution primarily uses which data structure to manage the search tree nodes? ○ A) Stack ○ B) Queue ○ C) Priority queue ○ D) Hash table	I	CO6
5C – 3	The Branch and Bound method is typically used for which type of problems? ○ A) Polynomial-time problems ○ B) NP-Hard problems ○ C) Linear programming problems ○ D) Greedy problems	I	CO6
5C – 4	Which of the following is true about NP-Hard problems? ○ A) They are solvable in polynomial time ○ B) They can be solved by a non-deterministic polynomial-time algorithm ○ C) They are at least as hard as the hardest problems in NP ○ D) They can be solved by dynamic programming in polynomial time	II	CO6
5C - 5	Cook's theorem demonstrates that: ○ A) The Traveling Salesperson Problem is NP-Hard ○ B) The class NP is equivalent to the class P ○ C) Any problem in NP can be reduced to the Boolean satisfiability problem (SAT) in polynomial time ○ D) There are no NP-Hard problems	II	CO6
5C - 6	A problem that is both NP-Hard and NP-Complete: ○ A) Can be solved in polynomial time if and only if $P = NP$	II	CO6

	○ B) Can be solved in exponential time but not in polynomial time ○ C) Has a known polynomial-time algorithm for solving it ○ D) Is guaranteed to be solvable in polynomial time		
5C - 7	Which of the following is NOT an NP-Complete problem? A) The Boolean Satisfiability Problem (SAT) B) The Traveling Salesperson Problem (TSP) C) The Knapsack Problem D) The Graph Coloring Problem	II	CO6
5C - 8	Which theorem proves that the Boolean Satisfiability Problem (SAT) is NP-Complete? A) Cook's Theorem B) P=NP Theorem C) Karp's Theorem D) Shannon's Theorem	I	CO6
5C - 9	In the context of NP-Complete problems, which of the following statements is TRUE? ○ A) NP-Complete problems can be solved in polynomial time ○ B) NP-Complete problems are a subset of NP-Hard problems ○ C) NP-Complete problems are only solvable in exponential time ○ D) NP-Complete problems can be solved in polynomial time if and only if P=NP	II	CO6
5C - 10	Which theorem proves that the Boolean Satisfiability Problem (SAT) is NP-Complete? ○ A) Cook's Theorem ○ B) P=NP Theorem ○ C) Karp's Theorem ○ D) Shannon's Theorem	II	CO6
5C - 11	A problem that can be verified in polynomial time but is not known to be solvable in polynomial time is classified as: ○ A) NP-Complete ○ B) NP-Hard ○ C) P ○ D) Co-NP	I	CO6
5C - 12	The class of problems that can be solved in polynomial time is denoted	I	CO6

	by: ○ A) NP ○ B) NP-Complete ○ C) P ○ D) NP-Hard		
5C - 13	The LC Branch and Bound method for solving the 0/1 Knapsack Problem involves: ○ A) Solving the problem using a greedy approach ○ B) Relaxing the integer constraints and solving the resulting fractional knapsack problem ○ C) Generating all possible combinations of items and selecting the optimal one ○ D) Using a heuristic to approximate the solution	I	CO6
5C - 14	In the Branch and Bound method, what is the primary role of the bounding function? ○ A) To create a complete search tree ○ B) To reduce the number of nodes by calculating upper or lower bounds and discarding suboptimal branches ○ C) To generate all possible solutions to the problem ○ D) To sort nodes in a priority queue	II	CO6
5C - 15	In the Branch and Bound method, which of the following is used to prune branches of the search tree that cannot lead to an optimal solution? ○ A) Depth-first search ○ B) Breadth-first search ○ C) Bounding function ○ D) Backtracking	II	CO6
FILL IN THE BLANKS			
5F - 1	The Branch and Bound method is used to solve optimization problems by systematically _____ the search space and using bounds to eliminate suboptimal solutions.	I	CO6
5F - 2	In the Branch and Bound method, the _____ function is used to provide upper or lower bounds to prune branches that cannot lead to an optimal solution.	I	CO6
5F - 3	The Traveling Salesperson Problem (TSP) can be approached using Branch and Bound by estimating the _____ of incomplete tours and pruning those that exceed the current best tour.	II	CO6
5F - 4	The LC Branch and Bound solution for the 0/1 Knapsack Problem involves solving a relaxed version of the problem using _____ programming to determine bounds	I	CO6

5F - 5	FIFO Branch and Bound typically uses a _____ to manage the nodes of the search tree in a first-in, first-out order.	I	CO6
5F - 6	In Branch and Bound, if a node's bound is worse than the current best solution, it is _____ from further consideration.	II	CO6
5F - 7	A common approach in the FIFO Branch and Bound method is to use a _____ to explore nodes level by level.	I	CO6
5F - 8	The Branch and Bound method is particularly useful for solving _____ problems where an exhaustive search is computationally infeasible.	I	CO6
5F - 9	NP-Hard problems are at least as difficult as the hardest problems in NP, and they may or may not be in _____.	I	CO6
5F - 10	A problem is classified as NP-Complete if it can be verified in polynomial time and every problem in _____ can be reduced to it in polynomial time.	I	CO6
5F - 11	If a problem is both NP-Hard and NP-Complete, it means that solving it in polynomial time would imply that _____ = NP.	I	CO6
5F - 12	Non-deterministic algorithms are able to verify solutions in _____ time but do not necessarily solve problems in polynomial time.	I	CO6
5F - 13	The class P consists of problems that can be solved in _____ time.	I	CO6
5F - 14	An NP-Complete problem is a problem for which every problem in NP can be reduced to it in _____ time.	I	CO6
5F - 15	A problem that can be verified in polynomial time but not necessarily solved in polynomial time is classified as _____.	II	CO6

MATCH THE FOLLOWING

5M - 1	1.Branch and Bound 2.FIFO Branch and Bound 3.LC Branch and Bound 4. Traveling Salesperson Problem A. Method used to solve optimization problems B. items can be divided or partially included C. Used to manage nodes in a first-in, first-out order D. Problem of finding the shortest tour	III	CO6
5M - 2	1. 0/1 Knapsack Problem 2. NP-Hard 3. NP-Complete 4. Cook's Theorem A. Problems that are at least as hard as the hardest problems in NP but may not be in NP B. Problems that can be verified in polynomial time and are as hard as any problem in NP C. Proves that the Boolean Satisfiability Problem (SAT) is NP-Complete D. where items can be divided or partially	III	CO6

	included		
5M - 3	1. Non-deterministic Algorithms 2. Class P 3. 0/1 Knapsack Problem 4.TSP Problem A. Problems that can be solved in polynomial time B. Algorithms that can verify solutions in polynomial time but may not solve the problem in polynomial time C. finds minimum cost tour D. A combinatorial optimization problem where items are either included or excluded	II	CO6
5M - 4	1.Branch and Bound 2.FIFO Branch and Bound 3.LC Branch and Bound 4.Traveling Salesperson Problem (TSP) A. Uses a queue to explore nodes level by level B. Method for solving optimization problems using bounds to prune suboptimal solutions C. The problem of finding the minimum cost tour visiting each city once and returning to the start D.Solves the 0/1 Knapsack Problem by relaxing integer constraints and using linear programming	II	CO6
5M - 5	1. NP-Hard 2.NP-Complete 3. Cook's Theorem 4. Non-deterministic Algorithms A. Algorithms that verify solutions in polynomial time but may not solve the problem in polynomial time B. Problems that are at least as difficult as the hardest problems in NP but may not be in NP C. Problems that can be verified in polynomial time and to which any NP problem can be reduced in polynomial time D. Proves that SAT is NP-Complete	II	CO6
5 – MARKS QUESTIONS			
5D - 1	Explain General method of Branch and Bound.(understanding)	II	CO6
5D - 2	Explain the Travelling Salesperson Problem with an example.(creating)	III	CO6
5D - 3	Explain branch and bound method to solve 0/1 knapsack problem	II	CO6
5D - 4	Describe LC Branch and Bound Solution	II	CO6
5D - 5	Distinguish NP-Hard and NP Complete classes	II	CO6

5D - 6	Explain Cook's Theorem with example (understanding)	II	CO6
5D - 7	Distinguish between Deterministic and non-deterministic algorithm.(analyzing)	II	CO6
5D - 8	What is meant by Halting problem explain with an example.(remembering)	II	CO6
5D - 9	Differentiate between NP-Complete and NP-Hard.(evaluating)	II	CO6
5D - 10	Explain maximum Clique.(understanding)	V	CO6
5D - 11	Explain Optimization problem.(understanding)	II	CO6
5D - 12	Briefly explain the concepts of the NP-Hard and NP-Complete?(understanding)	I	CO6
5D - 13	Explain the P, NP, NP-Hard and NP- complete classes? Give relationship between them?(understanding)	II	CO6
5D - 14	State and explain Cook's Theorem.(understanding)	I	CO6
5D - 15	Explain different types of problems?.(understanding)	II	CO6



SRIINDU COLLEGE OF ENGINEERING & TECHNOLOGY

Sheriguda(V), R.R Dist. (An Autonomous Institution Under UGC)

Department of AIML

YEAR/SEMESTER: III YEAR / I SEM/AIML-C

ACADEMIC YEAR: 2025-

2026 SUBJECT CODE: R22CSE3212 SUBJECT NAME: DAA

ASSIGNMENT -I

Batch	Assignment Questions	Course outcome	BT Level	Register
1	What do you mean by performance analysis of an algorithm? Explain	L2	CO3	23D41A66C To 23D41A66D 4
2	Define an algorithm. Explain the characteristics of the algorithm.	L3	CO1	
3	Explain Sum of subset problem with example .	L2	CO2	
4	Two sets S1 and S2 are given as below $S1 = \{1, 2, 4, 6\}$ and $S2 = \{7, 8\}$ (a) Draw Disjoint sets S1 and S2 using Trees. (b) Draw Disjoint sets S3 such that $S3 = S1 \cup S2$ Draw Disjoint sets S4 such that $S4 = S2 \cup S1$	L1	CO2	
5	Explain the general method of Dynamic programming.	L2	CO3	
Batch-2				
1	Explain the general method of Dynamic programming	L2	CO3	23D41A66D 5 To 23D41A66E
2	What is Performance analysis? Define different time complexity notations.	L4	CO1	
3	What is Performance analysis? Define different time complexity notations.	L1	CO2	
4	What is Performance analysis? Define different time complexity notations.	L4	CO2	
5	Implement Optimal binary search trees and dynamic programming	L2	CO3	
Batch-3				
1	Given $f(n) = 20n^3 - 3$, then prove that $f(n) = O(n^3)$?	L1	CO3	23D41A66E To 23D41A66E
2	Explain Binary search algorithm with an example?	L1	CO1	
3	Explain Disjoint set operations	L2	CO2	
4	Write pseudo code for union and find Algorithm.	L2	CO2	
5	Explain the general method of Dynamic programming	L2	CO3	

	Batch-4			
1	Develop general method for Divide and Conquer approach?	L2	CO3	23D41A66F3 To 23D41A66F9
2	Explain Merge sort algorithm using divide and conquer method	L6	CO1	
3	Explain General method of back tracking	L1	CO2	
4	Write algorithm to solve 8 queen's problem	L3	CO2	
5	Show the step by step procedure of deriving the minimum cost spanning tree using Kruskal's algorithm.	L2	CO3	
	Batch-5			
1	Explain Strassen's Matrix multiplication with example.	L3	CO3	23D41A66G0 To 23D41A66G4
2	Explain how quick sort sorts the following sequences of keys in ascending order. 22,55,33,11,99,77,55,66,54,21,32	L6	CO1	
3	Write and explain the find algorithm using collapse rule with an example.	L4	CO2	
4	Write pseudo code for union and find Algorithm.	L3	CO2	
5	Explain the general method of Dynamic programming	L2	CO3	
	Batch-6			
1	Explain the general method of Dynamic programming	L3	CO3	23D41A66G5 TO 23D41A66H1
2	What is binary search?	L2	CO1	
3	Give example for sum of subsets problem	L2	CO2	
4	Define chromatic number? Write algorithm for graph coloring problem using backtracking	L2	CO2	
5	Implement Optimal binary search trees and dynamic programming	L2	CO3	
	Batch-7			
1	What is sorting?	L3	CO3	23D41A66H2 To 23D41A66H8
2	Analyze time complexity of quick sort?	L2	CO1	
3	Explain Union and find algorithm	L2	CO2	
4	Define chromatic number? Write algorithm for graph coloring problem using backtracking	L3	CO2	
5	Show the step by step procedure of deriving the minimum cost spanning tree using Kruskal's algorithm.			
	Batch-8			
1	Compute time complexity of Merge sort.	L2	CO3	

2	What is binary search?	L2	CO1	23D41A66H7 To 23D41A66J1
3	Explain the usefulness of the following fundamental operations on sets (a) FIND (b) UNION	L3	CO2	
4	Implement Optimal binary search trees and dynamic programming	L3	CO2	
5	Show the step by step procedure of deriving the minimum cost spanning tree using Kruskal's algorithm.	L2	CO3	
	Batch-9			
1	Develop general method for Divide and Conquer approach?	L2	CO3	23D41A66J2 To 23D41A66J3
2	Explain how quick sort sorts the following sequences of keys in ascending order. 22, 55, 33, 11, 99, 77, 55, 66, 54, 21, 32	L3	CO1	
3	Show the step by step procedure of deriving the minimum cost spanning tree using Kruskal's algorithm.	L2	CO2	
4	Write and explain the find algorithm using collapse rule with an example.	L1	CO2	
5	What is general method of back tracking? Explain N queen problem	L2	CO3	
	Batch-10			
1	Find Big-oh notation and Little-oh notation for $f(n) = 7n^3 + 50n^2 + 200$	L2	CO3	23D41A66J4 To 23D41A66J8
2	Explain Binary search algorithm with an example?	L4	CO1	
3	Explain the usefulness of the following fundamental operations on sets (a) FIND (b) UNION	L1	CO2	
4	Determine weighted union and collapsing find algorithm With an Example.	L4	CO2	
5	Implement Optimal binary search trees and dynamic programming	L2	CO3	
	Batch-11			
1	Explain Strassen's Matrix multiplication with example.	L1	CO3	23D41A66J9 To 23D41A66K0
2	What is Performance analysis? Define different time complexity notations.	L1	CO1	
3	Write Algorithm for graph coloring problem	L2	CO2	
4	What is general method of back tracking? Explain N queen problem	L2	CO2	
5	Show the step by step procedure of deriving the minimum cost spanning tree using Kruskal's algorithm.	L2	CO3	

	Batch-12			
1	Given $f(n) = 20n^3 - 3$, then prove that $f(n) = O(n^3)$.?	L2	CO3	23D41A66K1 To 24D41A6610
2	What is binary search?	L6	CO1	
3	Write and explain the find algorithm using collapse rule with an example.	L1	CO2	
4	What is general method of backtracking? Explain N queen problem	L3	CO2	
5	Explain the general method of Dynamic programming	L2	CO3	



SRIINDU COLLEGE OF ENGINEERING & TECHNOLOGY

Sheriguda(V), R.R Dist. (An Autonomous Institution Under UGC)

Department of AIML

YEAR/SEMESTER: III YEAR / I SEM/AIML-C ACADEMIC YEAR: 2025-2026 SUBJECT
CODE: R22CSE3212 SUBJECT NAME: DAA

ASSIGNMENT -II

Batch No.	Assignment Questions	Course outcome	BT Level	Register Number
1	What do you mean by performance analysis of an algorithm? Explain	L2	CO3	23D41A66C9 To 23D41A66D4
2	Define algorithm. Explain the characteristics of the algorithm.	L3	CO4	
3	Explain Sum of subset problem with example.	L2	CO4	
4	Two sets S_1 and S_2 are given as below $S_1 = \{1, 2, 4, 6\}$ and $S_2 = \{7, 8\}$ (a) Draw Disjoint sets S_1 and S_2 using Trees. (b) Draw Disjoint sets S_3 such that $S_3 = S_1 \cup S_2$ Draw Disjoint sets S_4 such that $S_4 = S_2 \cup S_1$	L1	CO5	
5	Explain the general method of Dynamic programming.	L2	CO6	
	Batch-2			
1	Explain the general method of Dynamic programming	L2	CO3	23D41A66D5 To 23D41A66E0
2	What is Performance analysis? Define different time complexity notations.	L4	CO4 CO4	
3	What is Performance analysis? Define different time complexity notations.	L1	CO5	
4	What is Performance analysis? Define different time complexity	L4		

	notations.		CO6	
5	Implement Optimal binary search trees and dynamic programming	L2	CO3	
	Batch-3			
1	Given $f(n) = 20n^3 - 3$, then prove that $f(n) = O(n^3)$.?	L1	CO3	23D41A66E1 To 23D41A66E6
2	Explain Binary search algorithm with an example?	L1	CO4	
3	Explain Disjoint set operations	L2		
4	Write pseudo code for union and find Algorithm.	L2		
5	Explain the general method of Dynamic programming	L2	CO6	

	Batch-4			
1	Develop general method for Divide and Conquer approach?	L2	CO3	23D41A66F3 To 23D41A66F9
2	Explain Merge sort algorithm using divide and conquer method	L6	CO4	
3	Explain General method of back tracking	L1	CO5	
4	Write algorithm to solve 8 queen's problem	L3	CO6	
5	Show the step by step procedure of deriving the minimum cost spanning tree using Kruskal's algorithm.	L2	CO6	
	Batch-5			
1	Explain Strassen's Matrix multiplication with example.	L3	CO3	23D41A66G0 To 23D41A66G4
2	Explain how quick sort sorts the following sequences of keys in ascending order. 22, 55, 33, 11, 99, 77, 55, 66, 54, 21, 32	L6	CO4	
3	Write and explain the find algorithm using collapse rule with an example.	L4	CO4	
4	Write pseudo code for union and find Algorithm.	L3	CO5	
5	Explain the general method of Dynamic programming	L2	CO6	
	Batch-6			
1	Explain the general method of Dynamic programming	L3	CO3	

2	What is binary search?	L2	CO4 CO4	23D41A66G5 To 23D41A66H1
3	Give example for sum of subsets problem	L2	CO5	
4	Define chromatic number? Write algorithm for graph coloring problem using backtracking	L2	CO6	
5	Implement Optimal binary search trees and dynamic programming	L2		
	Batch-7		CO6	
1	What is sorting?	L3	CO3	23D41A66H2 To 23D41A66H8
2	Analyze time complexity of quick sort?	L2	CO4	
3	Explain Union and find algorithm	L2	CO4	
4	Define chromatic number? Write algorithm for graph coloring problem using backtracking	L3	CO5	
5	Show the step by step procedure of deriving the minimum cost spanning tree using Kruskal's algorithm.		CO6	
	Batch-8			
1	Compute time complexity of Merge sort.	L2	CO3	

2	What is binary search?	L2	CO1	23D41A66H7 To 23D41A66J1
3	Explain the usefulness of the following fundamental operations on sets (a) FIND (b) UNION	L3	CO2	
4	Implement Optimal binary search trees and dynamic programming	L3	CO2	
5	Show the step by step procedure of deriving the minimum cost spanning tree using Kruskal's algorithm.	L2	CO3	
	Batch-9			
1	Develop general method for Divide and Conquer approach?	L2	CO3	23D41A66J2 To 23D41A66J3
2	Explain how quick sort sorts the following sequences of keys in ascending order. 22, 55, 33, 11, 99, 77, 55, 66, 54, 21, 32	L3	CO4	
3	Show the step by step procedure of deriving the minimum cost spanning	L2		

	tree using Kruskal's algorithm.		CO4	
4	Write and explain the find algorithm using collapse rule with an example.	L1	CO5	
5	What is general method of back tracking? Explain N queen problem	L2	CO6	
	Batch-10			
1	Find Big-oh notation and Little-oh notation for $f(n) = 7n^3 + 50n^2 + 200$	L2	CO3	23D41A66J4 To 23D41A66J8
2	Explain Binary search algorithm with an example?	L4	CO4	
3	Explain the usefulness of the following fundamental operations on sets (a) FIND (b) UNION	L1	CO4	
4	Determine weighted union and collapsing find algorithm With an Example.	L4	CO5	
5	Implement Optimal binary search trees and dynamic programming	L2	CO6	
	Batch-11			
1	Explain Strassen's Matrix multiplication with example.	L1	CO3	23D41A66J9 To 23D41A66K0
2	What is Performance analysis? Define different time complexity notations.	L1	CO1	
3	Write Algorithm for graph coloring problem	L2	CO2	
4	What is general method of back tracking? Explain N queen problem	L2	CO2	
5	Show the step by step procedure of deriving the minimum cost spanning tree using Kruskal's algorithm.	L2	CO3	

	Batch-12			
1	Given $f(n) = 20n^3 - 3$, then prove that $f(n) = O(n^3)$?	L2	CO3	23D41A66K1 To 24D41A6610
2	What is binary search?	L6	CO4	
3	Write and explain the find algorithm using collapse rule with an example.	L1	CO4	
4	What is general method of backtracking? Explain N queen problem	L3	CO5	
5	Explain the general method of Dynamic programming	L2	CO6	



SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC, New Delhi) - Recognized under 2(f) and 12(B) of UGC Act 1956

III B.Tech. I Semester (REGULAR) End Examinations, DECEMBER – 2024.

(R22CSE3111) DESIGN AND ANALYSIS OF ALGORITHMS

30/12/2024

(For CSE, AIML & AIDS)

Day-1(FN)

Duration: 3 Hrs

Maximum Marks: 60M

Blooms Taxonomy : (I-Remembering, II-Understanding, III-Applying, IV-Analyzing, V-Evaluating and VI-Creating)

Course Outcomes : CO

PART – A

Answer ALL the following questions.

(10Qx1M=10M)

1. a) Define Algorithm.
- b) Explain about Big-oh notation.
- c) What is Backtracking?
- d) Write and explain about general iterative backtracking method.
- e) What is OBST?
- f) Briefly explain about Travelling Sales Man Problem.
- g) What is Greedy method?
- h) Distinguish between Prim's and Kruskal's algorithms.
- i) Define NP-Hard Problem.
- j) Explain about Non-deterministic algorithms.

I	CO1
II	CO1
I	CO2
III	CO2
II	CO3
I	CO3
I	CO4
I	CO4
II	CO5
III	CO5

PART – B

Answer FIVE questions choosing at least one from each unit.

(5Qx10M=50M)

UNIT-I

- 2a. i) Analyze the time complexity of Quick Sort Algorithm.
- ii) What are the Asymptotic notations? Explain with examples.

II CO1

(OR)

- 2b. i) Write and explain Recursive algorithm of binary search method.
- ii) What is Space Complexity? Explain with suitable examples.

III CO1

UNIT-II

- 3a. i) Illustrate about Sets and Disjoint sets.
- ii) Explain the weighted rule for finding UNION of sets.

II CO2

(OR)

- 3b. i) Explain about 4-queens problem with backtrack solution.
- ii) Discuss about number of connected components of a graph using disjoint set union.

III CO2

UNIT-III

- 4a. Use the function OBST to compute $w(i,j)$, $r(i,j)$, and $c(i,j)$, $0 \leq i < j \leq 4$, for the identifier set $(a_1, a_2, a_3, a_4) = (\text{cout}, \text{float}, \text{if}, \text{while})$ with $p(1) = 1/20$, $p(2) = 1/5$, $p(3) = 1/10$, $p(4) = 1/20$, $q(0) = 1/5$, $q(1) = 1/10$, $q(2) = 1/5$, $q(3) = 1/20$ and $q(4) = 1/20$. Using the $r(i,j)$'s construct the optimal binary search tree.

IV CO3

(OR)

- 4b. Discuss about all pairs shortest problem using dynamic programming.

II CO3

UNIT-IV

- 5a. Briefly discuss about an Optimal randomized algorithm of minimum-cost spanning trees.

III CO4

(OR)

- 5b. Explain about single source shortest path problem in Greedy method.

II CO4

UNIT-V

- 6a. Explain about the following

III CO5

i) LC Branch and Bound

ii) Cook's theorem

(OR)

III CO5

- 6b. Explain about the following

i) Nondeterministic knapsack algorithm

ii) FIFO branch and bound.

No.: R22CSE3212

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC, New Delhi)

Recognized under 2(f) and 12(B) of UGC Act 1956

DESIGN AND ANALYSIS OF ALGORITHM

Year: III Sem:I

Time: 3 Hours

Max Marks: 60

Answer <i>All</i> Questions			Marks	Course outcome.	BT Level
1	a	What do you mean by performance analysis of an algorithm? Explain (understanding)	(1)	CO1	I
	b	Define algorithm. Explain the characteristics of the algorithm (remembering)	(1)	CO1	I
	c	Write Algorithm for graph coloring problem	(1)	CO2	I
	d	Explain Sum of subset problem with example .(understanding)	(1)	CO2	I
	e	Explain the general method of Dynamic programming	(1)	CO3	I
	f	What is reliability? How to design reliability	(1)	CO4	IV
	g	Explain minimum cost spanning tree	(1)	CO5	I
	h	Explain kruskal's algorithm with an example	(1)	CO5	I
	i	Describe LC Branch and Bound Solution	(1)	CO6	I
	J	Distinguish NP-Hard and NP Complete classes	(1)	CO6	I

Part – B

(5Q X 10M = 50 Marks)

Answer all questions choosing ONE from each unit.		Marks	Course outcome.	BT Level
2	UNIT-I A) Find Big-oh notation and Little-oh notation for $f(n) = 7n^3 + 50n^2 + 200$. (evaluating) (OR) B) Explain Strassen's Matrix multiplication with example. (understanding)	(10)	CO1	III I
3	UNIT-II A) Define chromatic number? Write algorithm for graph coloring problem using backtracking (OR) B) Give example for sum of subsets problem	(10)	CO2	I II
4	UNIT-III A) Write and explain Algorithm for OBST. (OR) B) What is the Significance of 0/1 knapsack problem? how can you solve it	(10)	CO3 CO4	II I
5	UNIT-IV A) Explain General Method of Greedy Approach OR B) Explain Job sequencing algorithm with an example	(10)	CO5	III IV
6	UNIT-V Explain Cook's Theorem with example	(10)	CO6	IV