ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

A Unit of Alva's Education Foundation (R)
(Affiliated to Visvesvaraya Technological University, Belagavi.
Approved by AICTE, New Delhi & & Recognized by Government of Karnataka)
(Accredited by NAAC A+)

Checklist of Course File

S.No	Title
1.	Cover Page
2.	College Calendar/Department Calendar
3.	Time Table(Class + Personal)
4.	Course Content (Syllabus)
5.	Course Outcomes 8s Mapping Matrix
6.	Lesson Plan/Execution
7.	Student List
8.	Attendance Register
9.	Internal Test-1 Question Paper 8s Scheme
10.	Internal Test-11 Question Paper & Scheme
11.	Internal Test-III Question Paper & Scheme
12.	All Assignment Questions and Schemes(Rubrics)
13.	All Assignment Marks
14.	Final Internal Marks & Attendance Statement(As per Attendance Register)
15.	University Question Papers
16.	Notes or PPTs/Handouts or Materials
17.	University Result Analysis Statement
18.	Attainment of the Course as per CO/PO-PSO Mapping Matrix's
19.	List of Weak students and Bright students
20.	Sample Laboratory Observation and Record Book(only for Lab Courses)
21.	Sample Laboratory Internal Test Answer Sheets(only for Lab Courses)
22.	Project/ Seminar Report Formats and Regulations
23.	Other documents (If any)

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY
A Unit of Alva's Education Foundation (R)
(Affiliated to Visvesvaraya Technological University, Belagavi.
Approved by AICTE, New Delhi & & Recognized by Government of Karnataka)
Bhoomiyana Campus, Mijar, Moodbidri- 574 225, Mangalore, D.K., Karnataka State.
Phone: 08258-262724 (O), 262725(P), Telefax: 08258-262726
Email: principalaiet08@gmail.com, web: www.aiet.org.in

Department

Of

Computer Science & Design

Course Name	:	Analysis & Design of Algorithm
Course Code	:	BCS401
Semester	:	4th (IV)
Academic Year	:	2023-24 (Even Sem)

Faculty Incharge :

Dr. Shivaprasad B J

Assistant Professor

ALVA'S INSTITUTE OF ENGINEERING AND TECHNOLOGY
A Unit of Alva's Education Foundation (R)
(Affiliated to Visvesvaraya Technological University, Belagavi
Approved by AICTE, New Delhi & Recognised by Government of Karnataka)
Shobhavana Campus, Mijar, Moodbidri - 574 225, Mangalore, D.K., Karnataka State
Phone : 08258-262724 (O), 262725 (P), Telefax 08258-262726
Email : principalaiet08@gmail.com, Web www.aiet.org.in 28.06.2022

Department of Computer Science and Design

Vision of the department:

To be a globally recognized center in education, research and innovation in the field of computer science and design.

Mission of the department:

M1. To practice the latest industry trends in teaching learning process to make students ready for their professional life.

M2. To encourage interactions with experts in various domains from the industry/academia to inspire students to upgrade themselves in emerging trends andtechnologies.

M3. To collaborate with industry and professional bodies for the students to gauge the market trends and train accordingly.

M4. To inculcate ethics and human values among students and also to nurture innovation, entreprenеurship, research among students by providing a supportive environment where learning through active participation is encouraged.

Principal
PRINCIPAL
Alva's Institute of Engg. & Technology,
Mijar. MOODBIDRI - 574 225, D.K

Chairman
Chairman
Alva's Education Foundation (R)
MOODBIDRI - 574227, D.K.

ALVA'S INSTITUTE OF ENGINEERING AND TECHNOLOGY
A Unit of Alva's Education Foundation (R)
(Affiliated to Visvesvaraya Technological University, Belagavi
Approved by AICTE, New Delhi & Recognised by Government of Karnataka)
Shobhavana Campus, Mijar, Moodbidri - 574 225, Mangalore, D.K., Karnataka State
Phone : 08258-262724 (O), 262725 (P), Telefax 08258-262726
Email : principalaiet03@gmail.com, Web www.aiet.org.in

28.06.2022

Department of Computer Science and Design

Program Educational Objectives (PEO):

At the end of this course graduates will be able to

PEO 1: Utilize the knowledge and skillsets to work in diverse fields of computer science leading toinnovation and inter-disciplinary collaborative work

PEO 2: Get adjusted to the corporate work culture, focusing on timely completion of the assignedtasks, and while having the zeal to be updated in technological frontier

PEO 3: Be sensible towards the society and contribute through technology to address societal issuesthrough research and innovation while being loyal to the ethical and moral values.

Program Specific Outcomes (PSO)

PSO 1: Apply typical Software Engineering approaches and practices to develop software projectusing the concepts of Computer Science

PSO 2: Bring in innovative ideas in the areas like digital marketing, multimedia, gaming, Graphics, UI/UX and allied fields using modern

Principal
PRINCIPAL
Alva's Institute of Engg. & Technology,
Mijar, MOODBIDRI - 574 225, D.K

Chairman
Chairman
Alva's Education Foundation (R)
MOODBIDRI - 574227, D.K.

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(A Unit of Alva's Education Foundation)
Shobhavana Campus, Mijar-574225, Moodbidri, D.K,
Affiliated to VTU Belagavi and Approved by AICTE, New Delhi
(Accredited by NAAC with A+ Grade)

CALENDAR OF EVENTS – B.E. AND M.B.A. EVEN SEMESTER 2023-24

VISION

"Transformative education by pursuing excellence in Engineering and Management through enhancing skills to meet the evolving needs of the community"

MISSION

- To bestow quality technical education to imbibe knowledge, creativity and ethos to students community.
- To inculcate the best engineering practices through transformative education.
- To develop a knowledgeable individual for a dynamic industrial scenario.
- To inculcate research, entrepreneurial skills and human values in order to cater the needs of the society.

WEEK	MONTH	DAYS							ACTIVITIES
		MON	TUE	WED	THU	FRI	SAT	SUN	
01	APRIL 2024	1	2	3	4	5	6	7	9: Yugadi 11: Ramadan 22: Commencement of IV Semester 29: Commencement of VI Semester 20-24: CIE Test – I for II Semester 20-22: CIE Test – II for VIII Semester
02		8	9	10	11	12	13	14	
03		15	16	17	18	19	20	21	
04		22	23	24	25	26	27	28	
05		29	30						
06	MAY 2024			1	2	3	4	5	1: May Day 9-11: CIE Test – III for VIII Semester 10: Basava Jayanthi/ Akshaya Trithiya 11: Last Working Day of VIII Semester 30-31: Student Mentoring
07		6	7	8	9	10	11	12	
08		13	14	15	16	17	18	19	
09		20	21	22	23	24	25	26	
10		27	28	29	30	31			
11	JUNE 2024						1	2	3-5: CIE Test – I for IV & VI Semester 10: Commencement of IV Semester MBA 15-20: CIE Test – II for II Semester 17: Bakrid 27-28: Student Mentoring 29: Last Working Day of II Semester
12		3	4	5	6	7	8	9	
13		10	11	12	13	14	15	16	
14		17	18	19	20	21	22	23	
15		24	25	26	27	28	29	30	
16	JULY 2024	1	2	3	4	5	6	7	1-3: CIE Test – II for IV & VI Semester 10-12: CIE Test – I for IV Semester MBA 15: Commencement of II Semester MBA 17: Muharram 24-25: Student Mentoring 27-30: CIE Test – III for IV & VI Semester 31: Last Working Day of VI Semester
17		8	9	10	11	12	13	14	
18		15	16	17	18	19	20	21	
19		22	23	24	25	26	27	28	
20		29	30	31					
21	AUGUST 2024				1	2	3	4	7: Last Working Day of IV Semester 7-9: CIE Test – II for IV Semester MBA 15: Independence Day
22		5	6	7	8	9	10	11	
23		12	13	14	15	16	17	18	
24		19	20	21	22	23	24	25	
25		26	27	28	29	30	31		
26	SEPTEMBER 2024							1	7: Vara Siddi Vinayaka Vratha 11-13: CIE Test – III for IV Semester MBA 16: Id Milad
27		2	3	4	5	6	7	8	
28		9	10	11	12	13	14	15	
29		16	17	18	19	20	21	22	
30		23	24	25	26	27	28	29	
31		30							

Approved by IQAC Chairman

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(Unit of Alva's Education Foundation (R), Moodbidri)
Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi.
Recognized by Government of Karnataka.
A+, Accredited by NAAC & NBA (ECE & CSE)
Shobhavana Campus, MIJAR-574225, Moodbidri, D.K., Karnataka

Department of Computer Science and Design

Fourth semester time table with effect from 22-04-2024

Academic Year	Scheme	Semester	Section	Class Coordinator	Room No
2023-24	2022	4th	A	Dr. Shivaprasad B J	211

TIME / DAY	09.00 To 09.50	09.50 To 10.40	10.40 To 11.00	11.00 To 11.50	11.50 To 12.40	12.40 To 01.40	01.40 To 02.30	02.30 To 03.20	03.30 To 5.00
MON	ADA (SBJ)	UHV (PD)	TEA BREAK	DBMS (JKR)	DMS (RSH)	LUNCH BREAK	CGV (SBJ)	VAC	NSS/PE/YOGA
TUE	LaTex (RHY)	CGV (SBJ)		DBMS (JKR)	DMS (RD)		<...ADA-LAB (SHJ) (CO:JKR)...>		
WED	ADA (SBJ)	DBMS (JKR)		LaTex (RH)	ADA (SBJ)		<.DBMS LAB(JKR) (CO:SHJ)..>		
THU	LaTex (RH)	CGV (SBJ)		.VAC	ADA (SBJ)		<..... CGV-LAB(SHJ)(CO:RHY) ..>		
FRI	DMS (RSH)	CGV (SBJ)		DBMS	UHV (RC)		Aptitude/ LaTex (RH)	CGV (SBJ)	BFE(KR)
SAT	DMS (RD)	DBMS (JKR)		ADA (SBJ)	LaTex (RH)				

Allocation of Courses

Course Code	Course Initial	Couse Title	Name of the Faculty	Faculty Initial
BCS401	ADA	Analysis & Design of Algorithms	Dr. Shivaprasad B J	SBJ
BCG402	CGV	Computer Graphics and Visualization	Dr. Shivaprasad B J	SBJ
BCS403	DBMS	Database Management Systems	Prof. Jayanth KumarA rathod	JKR
BCSL404	ADA-LAB	Analysis & Design of Algorithms Lab	Dr. Shivaprasad B J	SHJ
BCG456C	LaTex	Technical Writing using LaTeX	Mr. Ramachandra HY	RHY
BCS405B	DMS	Discrete Mathematical Structures	Mrs. Radhika/Mrs Rashimi	RD/RSH
BBOK407	BFE	Biology For Engineers	Dr. Keerthi Raj	KR
BUHK408	UHV	Universal human values course	(MON) Dr. Bharath (Homeopathy)	PD
			(FRI) (Naturopathy) Dr.	
MC	BNSK459	National Service Scheme (NSS)	Dr. Suresh PS (Mechanical)	SPS
		Physical Education (PE) (Sports and Athletics)	Mr. Dilip (PD)	DPD
		Yoga		
Aptitude		Aptitude	Deepak Kottari	DK
VAC		Value added course	Mr. Ravichandra	RC

H.O.D
Dept. of Computer Science and Design
Alva's Institute of Engg. & Technology
Mijar, Moodbidri - 574 225

Dean Academics

Principal
Alva's Institute of Engg. & Technology
Mijar, MOODBIDRI - 574 225, D.K.

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(Unit of Alva's Education Foundation (R), Moodbidri)
Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi.
Recognized by Government of Karnataka.
A+, Accredited by NAAC & NBA (ECE & CSE)
Shobhavana Campus, MIJAR-574225, Moodbidri, D.K., Karnataka

Department of Computer Science and Design

Faculty personal time table Even Semester 2023-24

Academic Year		Scheme	Semester		Section	Class Coordinator			Room No
2023-24		2022	IV		A	Dr. Shivaprasad B J			211
TIME DAY	09.00 To 09.50	09.50 To 10.40	10.40 To 11.00	11.00 To 11.50	11.50 To 12.40	12.40 To 01.40	01.40 To 02.30	02.30 To 03.20	03.30 To 5.00
MON	ADA (SBJ)					LUNCH BREAK	CGV (SBJ)		
TUE		CGV (SBJ)					<...ADA-LAB (SHJ) (CO:JKR)...>		
WED	ADA (SBJ)				ADA (SBJ)		<.DBMS LAB(JKR) (CO:SHJ)..>		
THU		CGV (SBJ)			ADA (SBJ)		<.... CGV-LAB(SHJ)(CO:RHY) ..>		
FRI		CGV (SBJ)						CGV (SBJ)	
SAT	ADA (SBJ)			ADA (SBJ)					

Allocation of Subjects

Course Code	Course Initial	Course title	Class
BCS401	ADA	Analysis & Design of Algorithms	4th SEM
BCG402	CGV	Computer Graphics and Visualization	4th SEM
BCSL404	ADA-LAB	Analysis & Design of Algorithms Lab	4th SEM
BCS401	CGV-lab	Computer Graphics and Visualization-LAB	4th SEM
BCS403-LAB	DBMS-Lab	Database Management Systems-Lab	4th SEM

Workload Units

Theory	Lab	Others	Total
(5*2)*2= 20	3*3=9	10	39

Additional Responsibilities

Class Coordinator – IV Semester, NAAC Criteria 3, 4 Coordinator, Research Coordinator, Mentor for 4th sem

Placement Coordinator, Certification Course (Wipro,NPTEL), Students competition coordinator,

HOD
H.O.D
Dept. of Computer Science and Design
Alva's Institute of Engg. & Technology

Dean Academics

PRINCIPAL
Alva's Institute of Engg. & Technolo
Mijar, MOODBIDRI - 574 225, D.K

Analysis & Design of Algorithms		Semester	4
Course Code	BCS401	CIE Marks	50
Teaching Hours/Week (L: T:P: S)	3:0:0:0	SEE Marks	50
Total Hours of Pedagogy	40	Total Marks	100
Credits	03	Exam Hours	03
Examination type (SEE)	Theory		

Course objectives:

- To learn the methods for analyzing algorithms and evaluating their performance.
- To demonstrate the efficiency of algorithms using asymptotic notations.
- To solve problems using various algorithm design methods, including brute force, greedy, divide and conquer, decrease and conquer, transform and conquer, dynamic programming, backtracking, and branch and bound.
- To learn the concepts of P and NP complexity classes.

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

1. Lecturer method (L) does not mean only the traditional lecture method, but different types of teaching methods may be adopted to achieve the outcomes.
2. Utilize video/animation films to illustrate the functioning of various concepts.
3. Promote collaborative learning (Group Learning) in the class.
4. Pose at least three HOT (Higher Order Thinking) questions in the class to stimulate critical thinking.
5. Incorporate Problem-Based Learning (PBL) to foster students' analytical skills and develop their ability to evaluate, generalize, and analyze information rather than merely recalling it.
6. Introduce topics through multiple representations.
7. Demonstrate various ways to solve the same problem and encourage students to devise their own creative solutions.
8. Discuss the real-world applications of every concept to enhance students' comprehension.

Module-1

INTRODUCTION: What is an Algorithm?, Fundamentals of Algorithmic Problem Solving.
FUNDAMENTALS OF THE ANALYSIS OF ALGORITHM EFFICIENCY: Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non recursive Algorithms, Mathematical Analysis of Recursive Algorithms.
BRUTE FORCE APPROACHES: Selection Sort and Bubble Sort, Sequential Search and Brute Force String Matching.
Chapter 1 (Sections 1.1,1.2), Chapter 2(Sections 2.1,2.2,2.3,2.4), Chapter 3(Section 3.1,3.2)

Module-2

BRUTE FORCE APPROACHES (contd..): Exhaustive Search (Travelling Salesman probem and Knapsack Problem).
DECREASE-AND-CONQUER: Insertion Sort, Topological Sorting.
DIVIDE AND CONQUER: Merge Sort, Quick Sort, Binary Tree Traversals, Multiplication of Large Integers and Strassen's Matrix Multiplication.

Chapter 3(Section 3.4), Chapter 4 (Sections 4.1,4.2), Chapter 5 (Section 5.1,5.2,5.3, 5.4)
Module-3
TRANSFORM-AND-CONQUER: Balanced Search Trees, Heaps and Heapsort. **SPACE-TIME TRADEOFFS:** Sorting by Counting: Comparison counting sort, Input Enhancement in String Matching: Horspool's Algorithm. **Chapter 6 (Sections 6.3,6.4), Chapter 7 (Sections 7.1,7.2)**
Module-4
DYNAMIC PROGRAMMING: Three basic examples, The Knapsack Problem and Memory Functions, Warshall's and Floyd's Algorithms. **THE GREEDY METHOD:** Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees and Codes. **Chapter 8 (Sections 8.1,8.2,8.4), Chapter 9 (Sections 9.1,9.2,9.3,9.4)**
Module-5
LIMITATIONS OF ALGORITHMIC POWER: Decision Trees, P, NP, and NP-Complete Problems. **COPING WITH LIMITATIONS OF ALGORITHMIC POWER:** Backtracking (n-Queens problem, Subset-sum problem), Branch-and-Bound (Knapsack problem), Approximation algorithms for NP-Hard problems (Knapsack problem). **Chapter 11 (Section 11.2, 11.3), Chapter 12 (Sections 12.1,12.2,12.3)**

Course outcome (Course Skill Set)

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

1. Apply asymptotic notational method to analyze the performance of the algorithms in terms of time complexity.
2. Demonstrate divide & conquer approaches and decrease & conquer approaches to solve computational problems.
3. Make use of transform & conquer and dynamic programming design approaches to solve the given real world or complex computational problems.
4. Apply greedy and input enhancement methods to solve graph & string based computational problems.
5. Analyse various classes (P,NP and NP Complete) of problems
6. Illustrate backtracking, branch & bound and approximation methods.

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50) and for the SEE minimum passing mark is 35% of the maximum marks (18 out of 50 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together.

Continuous Internal Evaluation:

- For the Assignment component of the CIE, there are 25 marks and for the Internal Assessment Test component, there are 25 marks.
- The first test will be administered after 40-50% of the syllabus has been covered, and the second test will be administered after 85-90% of the syllabus has been covered
- Any two assignment methods mentioned in the 22OB2.4, if an assignment is project-based then only one assignment for the course shall be planned. The teacher should not conduct two assignments at the end of the semester if two assignments are planned.
- For the course, CIE marks will be based on a scaled-down sum of two tests and other methods of assessment.

Internal Assessment Test question paper is designed to attain the different levels of Bloom's taxonomy as per the outcome defined for the course.

Semester-End Examination:

Theory SEE will be conducted by the University as per the scheduled timetable, with common question papers for the course (**duration 03 hours).**

1. The question paper will have ten questions. Each question is set for 20 marks.
2. There will be 2 questions from each module. Each of the two questions under a module (with a maximum of 3 sub-questions), **should have a mix of topics** under that module.
3. The students have to answer 5 full questions, selecting one full question from each module.
4. Marks scored shall be proportionally **reduced to 50 marks**

Suggested Learning Resources:

Textbooks

1. Introduction to the Design and Analysis of Algorithms, By Anany Levitin, 3rd Edition (Indian), 2017, Pearson.

Reference books

1. Computer Algorithms/C++, Ellis Horowitz, SatrajSahni and Rajasekaran, 2nd Edition, 2014, Universities Press.
2. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 3rd Edition, PHI.
3. Design and Analysis of Algorithms, S. Sridhar, Oxford (Higher Education)

Web links and Video Lectures (e-Resources):

- Design and Analysis of Algorithms: https://nptel.ac.in/courses/106/101/106101060/

Activity Based Learning (Suggested Activities in Class) / Practical Based learning

- Promote real-world problem-solving and competitive problem solving through group discussions to engage students actively in the learning process.
- Encourage students to enhance their problem-solving skills by implementing algorithms and solutions through programming exercises, fostering practical application of theoretical concepts.

Assessment Methods -

1. Problem Solving Assignments (Hacker Rank/ Hacker Earth / Leadcode)
2. Gate Based Aptitude Test

Alva's Institute of Engineering & Technology

Shobhavana Campus, Mijar, Moodbidri, D.K – 574225

DEPARTMENT OF COMPUTER SCIENCE & DESIGN

	SEMESTER 1V
Course Code: **BCS401**	Course Name: **Analysis & Design of Algorithms**
Course Teacher: **Dr.Shivaprasad B J**	

Course Outcomes: After studying this course, students will be able to:

CO Numbers	Course Outcomes	Blooms Level	Target Level
BCS401.1	Understand the notion of an algorithm, asymptotic notations and divide and conquer.	L2 Understand	2
BCS401.2	Analyze the recursive and non-recursive algorithms and divide and conquer.	L4 Analyze	2
BCS401.3	Understand the algorithm design techniques using greedy method.	L2 Understand	2
BCS401.4	Understand the algorithm design techniques using dynamic programming.	L2 Understand	2
BCS401.5	Explain the algorithm design techniques using backtracking, branch and bound and NP-complete and NP-hard problems.	L2 Understand	2

CO-PO/CO-PSO Mapping Matrix:

CO Numbers	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
BCS401.1	3	3	1	1	1							1	3	1	
BCS401.2	2	3	1	1	1							1	3	1	
BCS401.3	2	2	3	1	1							1	3	1	
BCS401.4	2	2	3	1	1							1	3	1	
BCS401.5	2	2	3	1	1							1	3	1	
AVG.	**2.2**	**2.4**	**2.2**	**1**	**1**							**1**	**3**	**1**	

CO	PO	Level	Justification
BCS401.1	PO1	3	Strongly mapped as students will be able to gain the knowledge of asymptotic notations, divide and conquer.
	PO2	3	Strongly mapped as students will be able to analyze the algorithms of divide and conquer.
	PO3	1	Slightly mapped as students will be able to design new algorithms using divide and conquer technique.
	PO4	1	Slightly mapped as students will be able to choose the appropriate algorithms such as merge and quick sort to design the applications to solve complex problems.
	PO5	1	Slightly mapped as students will be able use tools such as eclipse, netbeans to design and develop applications.
	PO12	1	Slightly mapped as students will be able to apply the concept of divide and conquer technique in searching and sorting etc. problem types.

	PSO1	3	Strongly mapped as students will be able to apply the searching and sorting techniques in real world problems.
	PSO2	1	Slightly mapped as students will be able to apply the concept of algorithms in system software such as compilers and debuggers.
BCS401.2	PO1	2	Moderately mapped as students will be able to gain the knowledge of general plan of recursive and non-recursive algorithms and theory of backward substitution in divide and conquer technique.
	PO2	3	Strongly mapped as students will be able to analyze the time and space complexity of recursive and non-recursive algorithms.
	PO3	1	Slightly mapped as students will be able to design new algorithms using divide and conquer technique.
	PO4	1	Slightly mapped as students will be able to choose the appropriate algorithms in design of applications to solve complex problems.
	PO5	1	Slightly mapped as students will be able use tools such as eclipse, netbeans to design and develop applications.
	PO12	1	Slightly mapped as students will be able to analyze the efficiency of any new algorithm designed by divide and conquer.
	PSO1	3	Strongly mapped as students will be able to apply the searching and sorting techniques in real world problems.
	PSO2	1	Slightly mapped as students will be able to apply the concept of algorithms in system software such as compilers and debuggers.
BCS401.3	PO1	2	Moderately mapped as students will be able to gain the knowledge of greedy method concepts.
	PO2	2	Moderately mapped as students will be able to analyze the time and space complexity of greedy algorithms.
	PO3	3	Strongly mapped as students will be able to design new algorithms using greedy technique.
	PO4	1	Slightly mapped as students will be able to choose the greedy algorithms in design of applications to solve complex problems.
	PO5	1	Slightly mapped as students will be able use tools such as eclipse, netbeans to design and develop applications.
	PO12	1	Slightly mapped as students will be able to analyze the efficiency of any new algorithm designed by greedy techniques.
	PSO1	3	Strongly mapped as students will be able to apply the greedy techniques in real world problems such as Resource management in software.
	PSO2	1	Slightly mapped as students will be able to apply the concept of algorithms in system software such as compilers and debuggers.
BCS401.4	PO1	2	Moderately mapped as students will be able to gain the knowledge of dynamic method concepts.
	PO2	2	Moderately mapped as students will be able to analyze the time and space complexity of dynamic programming algorithms.
	PO3	3	Strongly mapped as students will be able to design new algorithms using dynamic programming technique.
	PO4	1	Slightly mapped as students will be able to choose the dynamic algorithms in design of applications to solve complex problems.
	PO5	1	Slightly mapped as students will be able use tools such as eclipse, netbeans to design and develop applications.
	PO12	1	Slightly mapped as students will be able to analyze the efficiency of any new algorithm designed by dynamic programming techniques.
	PSO1	3	Strongly mapped as students will be able to apply the greedy techniques in real world problems such as TSP, reliability design.

	PSO2	1	Slightly mapped as students will be able to apply the concept of algorithms in system software such as compilers and debuggers.
BCS401.5	PO1	2	Moderately mapped as students will be able to gain the knowledge of backtracking, branch and bound concepts.
	PO2	2	Moderately mapped as students will be able to analyze the time and space complexity of backtracking, branch and bound algorithms.
	PO3	3	Strongly mapped as students will be able to design new algorithms using backtracking, branch and bound technique.
	PO4	1	Slightly mapped as students will be able to choose the backtracking and branch & bound algorithms in design of applications to solve complex problems.
	PO5	1	Slightly mapped as students will be able use tools such as eclipse, netbeans to design and develop applications.
	PO12	1	Slightly mapped as students will be able to analyze the efficiency of any new algorithm designed by backtracking, branch and bound techniques.
	PSO1	3	Strongly mapped as students will be able to apply the backtracking, branch and bound in real world problems such as event scheduling, stratification of triangle meshes in computer graphics.
	PSO2	1	Slightly mapped as students will be able to apply the concept of algorithms in system software such as compilers and debuggers.

Course Teacher
Signature with date

IQAC Member
Signature with date

IQAC Chairman
Signature with date

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

A Unit of Alva's Education Foundation (R)
(Affiliated to Visvesvaraya Technological University, Belagavi.
Approved by AICTE, New Delhi & & Recognized by Government of Karnataka)
Shobhavana Campus, Mijar, Moodbidri- 574 225, Mangalore, D.K., Karnataka State.
(Accredited by NAAC with A+ Grade)

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

4th Sem (2022 Batch)
Student List

Sr. No	USN	Name of the student
1.	4AL22CG001	ABAYCHANDRASURYA J K
2.	4AL22CG002	ALOK G
3.	4AL22CG003	ANANT R KULKARNI
4.	4AL22CG004	ANVESH A SHETTY
5.	4AL22CG005	APEKSHA K
6.	4AL22CG006	ASHWINI
7.	4AL22CG007	BIBIFATIMA
8.	4AL22CG008	CHANDRASHEKHAR P C
9.	4AL22CG009	CHANNAKESHAVASHREE D R
10.	4AL22CG010	CHIDVILAS N
11.	4AL22CG011	DARSHAN HALAKERIMATH
12.	4AL22CG012	DARSHAN RAGHURAM SHETTY
13.	4AL22CG013	DARSHAN S GOWDA
14.	4AL22CG014	DHANRAJ S H
15.	4AL22CG015	DHANUSHREE N
16.	4AL22CG016	DHANYASHREE
17.	4AL22CG017	DRUSHYA HEGDE D
18.	4AL22CG018	FAHAD UMAR FAROOQ
19.	4AL22CG019	GANESH
20.	4AL22CG020	GOUSIYA G
21.	4AL22CG021	HANAMANT RANGAPPA PUJARI
22.	4AL22CG022	HEMANTH K V
23.	4AL22CG023	HITHASHREE
24.	4AL22CG024	JAGADISH GOWDA B J
25.	4AL22CG025	KARTIK MYAGERI
26.	4AL22CG026	KATHANIKA MOGERAYA
27.	4AL22CG027	KHUSHI SHETTY
28.	4AL22CG028	KOMAL KALIKAR
29.	4AL22CG029	MAHAMMED ASIM
30.	4AL22CG030	MANJUNATH
31.	4AL22CG031	MOHAMMED IRFAN

32.	4AL22CG032	MOUNASHREE D P
33.	4AL22CG033	NAGARAJ DADDI
34.	4AL22CG034	NAVEEN HIREMATH
35.	4AL22CG035	NIKHITH NAIK
36.	4AL22CG036	NIRIKSHA RAI
37.	4AL22CG037	NISHCHAL
38.	4AL22CG038	PARESH PRAKASH TALEKAR
39.	4AL22CG039	PAVAN RAJ
40.	4AL22CG040	POOJARI SURAKSHA SURENDRA
41.	4AL22CG041	POONAM NAIR
42.	4AL22CG042	PRADEEP U NAIK
43.	4AL22CG043	PRANAMI M SHETTY
44.	4AL22CG044	PRATHEEKSHA SHETTY
45.	4AL22CG045	PREMKUMAR INGALE
46.	4AL22CG046	PRIYA BHEEMARAJ WADAWADIGI
47.	4AL22CG047	PRIYANKA IRANNA SAKHARE
48.	4AL22CG048	RAKSHITHA
49.	4AL22CG049	RASHMI M
50.	4AL22CG050	SAI SANKARSH N S
51.	4AL22CG051	SANIKA GOWDA N S
52.	4AL22CG052	SEJAL RAJESH NAIK
53.	4AL22CG053	SHAYANASHREE N
54.	4AL22CG054	SHRUTHI N
55.	4AL22CG055	SNEHA DATTATRYA GOUDA
56.	4AL22CG056	SNEHA S
57.	4AL22CG057	SOWJANYA S
58.	4AL22CG058	SURAJ
59.	4AL22CG059	TEJESVIN R
60.	4AL22CG060	THANVI S SANIL
61.	4AL22CG061	THIRUMALA
62.	4AL22CG062	VAISHNAVI G

USN | | | | | | | | | | |

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(Accredited by NAAC with A+ Grade)

Department of Computer Science & Design

Continuous Internal Evaluation Test – I AY 2023-24

Course Title : Analysis & Design of Algorithms		Course Code: BCS401
Date: 03/06/2024	Time:9.30 AM -11.00 AM	Semester/Section: IV
Faculty: Dr. Shivaprasad B J		Max. Marks:25

Note: Answer ONE FULL question from each part.

Q. No.		Questions	Marks	COs	BTL
		PART – I			
1	a)	Give a mathematical analysis of non-recursive matrix multiplication algorithms.	7	CO1	L3
	b)	What are the various basic Asymptotic efficiency classes? Explain Big-O, Big-Ω, Big- Θ with an example.	6	CO1	L2
		OR			
2	a)	Give the general plan for analyzing time efficiency of recursive algorithms and also analyses the Tower of Hanoi recursive algorithms.	7	CO1	L3
	b)	Explain fundamentals of algorithmic problem solving.	6	CO1	L2
		PART-II			
3	a)	Write an algorithm for Selection Sort and mention the time complexity.	6	CO2	L3
	b)	Design an algorithm for Merge Sort and apply the same to sort the following input 50, 30, 10, 90, 80, 20, 40, 70.	6	CO2	L3
		OR			
4	a)	Sort the given below given array of elements using Bubble Sort and mention the time complexity. 4, 19 , 0, 6, 8, 9, 2, 3, 12	6	CO2	L3
	b)	Design an algorithm for Quick Sort and apply the same to sort the following input 5, 3, 1, 9, 8, 2, 4, 7.	6	CO2	L3

Levels of Bloom's Taxonomy

No.	L1	L2	L3	L4	L5	L6
Level	Remember	Understand	Apply	Analyze	Evaluate	Create

Course Outcomes

CO1	Apply asymptotic notational method to analyze the performance of the algorithms in terms of time complexity.
CO2	Demonstrate divide & conquer approaches and decrease & conquer approaches to solve computational problems.
CO3	Make use of transform & conquer and dynamic programming design approaches to solve the given real world or complex computational problems.
CO4	Apply greedy and input enhancement methods to solve graph & string based computational problems.
CO5	Analyse various classes (P,NP and NP Complete) of problems
CO6	Illustrate backtracking, branch & bound and approximation methods.

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(Accredited by NAAC with A+ Grade)

QUESTION PAPER REVIEW REPORT

Continuous Internal Evaluation (CIE) Test: I AY 2023-24

Department : CSD Semester/Section: 4 (A) Max Marks: 25

Course Title: ADA Course Code (BCS401) Date: 01/06/2024

Faculty: Dr. Shivaprasad B J

Qn. No.	Course Outcome (CO)	Bloom's Taxonomy Level	Marks
1a	CO1	L3	7
1b	CO1	L2	6
2a	CO1	L3	7
2b	CO1	L2	6
3a	CO2	L3	6
3b	CO2	L3	6
4a	CO2	L3	6
4b	CO2	L3	6
Total Marks			50

BT Level: L1-Remember, L2-Understand, L3 -Apply, L4 -Analyze, L5- Evaluate, L6- Create

Consolidated Marks for Different BT Levels:

BT Level	Marks for Each Level	% of Marks	Remarks
L2	6	24	
L3	6/7	76	

Scrutinizer/Reviewer Remark:

Approved	*Approved*	Approved with Correction		Rejected	
Reason for Rejection					

Dr. Shivaprasad B J

Name & Signature of the Scrutinizer

Date: 01/06/2024

Mr. Jayantkumar A Rathod

Name & Signature of the IQAC Coordinator

Date: 01/06/2024

Mr. Jayantkumar A Rathod

Signature of Head of the Department

Alva's Institute of Engineering & Technology, Moodbidri

(Accredited by NAAC with A+ Grade)

Department of Computer Science & Design

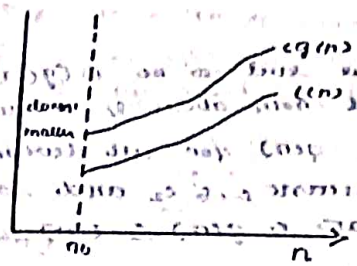
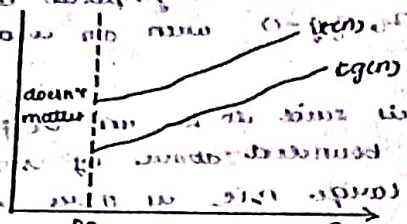
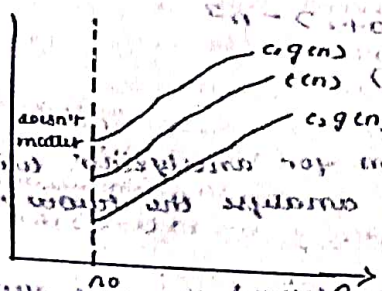
Scheme of Evaluation - Even Semester 2023-24

Continuous Internal Evaluation Test-I

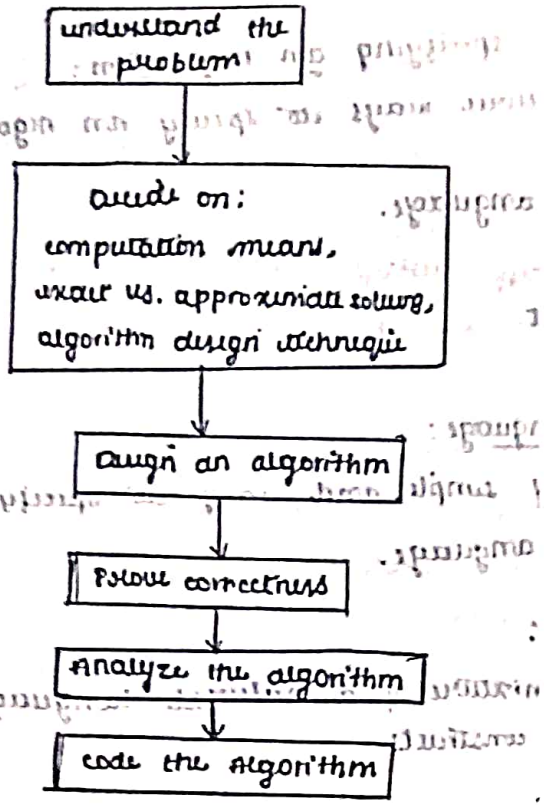
Course Title: **Analysis & Design of Algorithms**		Course Code: BCS401
Date: 03/06/2024	Time: 9.30 AM- 11.00 AM	Semester: IV
Faculty: Dr. Shivaprasad B J		Max. Marks: 25

Note: Answer ONE FULL question from each Module.

Q. No.		Scheme of Evaluation	Marks
1	(a)	Algorithm: 1] Input parameter - n * n order of matrix both A & B. 2] Basic operation - multiplication A[i, k] * B[k, j] 3] Depends only on order of matrix - No variance. 4] $M(n) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1}$ 5] $M(n) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} 1$ $= \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (n-1-0)$ $= n^2 (n-1+0+1) = n^3$ $M(n) \in \Theta(n^3)$	3 marks 4 marks ---------- 7 marks

1	(b)	**Big-O :** A function $f(n)$ is said to be in $O(g(n))$, denoted $f(n) \in O(g(n))$. If $f(n)$ is bounded above by some constant multiple of $g(n)$ for all large n, i.e., there exist some positive constant c and some non negative integer n_0 such that $f(n) \le c\,g(n)$ for all $n \ge n_0$  **Big-Ω :** A function $f(n)$ is said to be in $\Omega(g(n))$, denoted $f(n) \in \Omega(g(n))$, if $f(n)$ is bounded below by some positive constant multiple of $g(n)$ for all large n, i.e., if there exists some positive constant c & some non-negative integer n_0 such that, $f(n) \ge c\,g(n)$ for all $n \ge n_0$  Example: $4n^2 = \Omega(n^2)$, with $c = 1$ & $n_0 = 16$. Use of function in $\Omega(n^2)$. **Big-Θ :** A function $f(n)$ is said to be $\Theta(g(n))$ denoted $f(n) \in \Theta(g(n))$ if $f(n)$ is bounded both above & below by some positive constant multiples of $g(n)$ for all large n, if there exist some positive constant c_1 & c_2 and some non-negative integer n_0 such that $c_2\,g(n) \le f(n) \le c_1\,g(n)$ for all $n \ge n_0$.  eg: $n^2/2 - 2n = \Theta(n^2)$, with $c_1 = 1/4$, $c_2 = 1/2$ & $n_0 = 8$.	2 marks 2 marks 2 marks 6 marks

		➢ General plan for analyzing the time efficiency of recursive algorithm. 1. Decide on a parameter indicating an input size. 2. Identify the algorithm basic operation. 3. Check whether the number of times the basic operation is executed can vary on different inputs of the same size. If it can, the worst time, average case, best-case efficiencies must be investigated separately. 4. Setup a recurrence relation with an appropriate initial condition for the number of times the basic operation is executed. 5. Solve the recurrence or at least ascertain the order of growth of its solution. Analysis: 1. Input parameter : no of disks 'n'. 2. Basic operation : moving disks. 3. depends only on 'n' : no variation. 4. $\text{tower}(n,S,T,D) = \begin{cases} \text{move disc 1 from } S \text{ to } D & n=1 \\ \text{tower}(n-1,S,D,T) & \\ \text{move disc } (n) \text{ from } S \text{ to } D & n>1 \end{cases}$	3 marks
2	(a)	Based on basic operation build recursive relation. $m(n) = \begin{cases} 1 & n=1 \\ m(n-1)+1+m(n-1) & n>1 \end{cases}$ $m(n) = 2m(n-1)+1$ until $m(1)=1$ $m(n) = 2m(n-1)+1$ where $= 2(2m(n-2)+1)+1$ — $m(n) = 2m(n-1)+1$ $= 2^2 m(2m(n-3)+1)+1+1$ — $m(n-1) = 2m(n-2)+1$ $= 2^3 m(n-3)+2^2+2+1$ — $m(n-2) = 2m(n-3)+1$ $= 2^3 m(n-3)+2^2+2+1$ $\vdots$ $= 2^{i} m(n-i)+2^{i-1}+2^{i-2}+\dots+2+1$ $= 2^{n-1} m(n(n-1))+2^{n-2}+2^{n-3}+4+2+1$ — $m(1)=2$, $n-i=1$, $i=n-1$ $= 2^{n-1} m(1)+2^{n-2}+2^{n-3}\dots 4+2+1$ $= 2^{n-1}+2^{n-2}+2^{n-3}\dots 4+2+1$ $= \underline{2^{n-1}}$ $m(n) \in \Theta(2^n)$ $ar^0 + ar^1 + ar^2 \dots ar^n = a\frac{(r^{n+1}-1)}{r-1}$ $1+2+4+8+\dots 2^{i-1} = 2^n = \frac{2^{n+1}-1}{2-1}$ $a=1 \quad r=2 = \underline{2^{n+1}-1}$	4 marks ---------- 7 marks

2	(b)		2 marks 2 marks 2 marks --- 6 marks

3	(a)	<u>Selection Sort Algorithm:</u>
		Algorithm selection sort (A[0....n-1])
		// sort a given array by selection sort
		// Input: An array of A[0....n-1] of orderable element.
		// output: Array A[0...n-1] sorted in ascending order.
		for i ← 0 to n-2 do
		min ← i
		for j ← i+1 to n-1 do
		if A[j] < A[min]
		min ← j
		swap A[i] and A[min]
		Time Coimplexity
			3 marks
			
			3 marks
			--- 	
			6marks

merge sort:

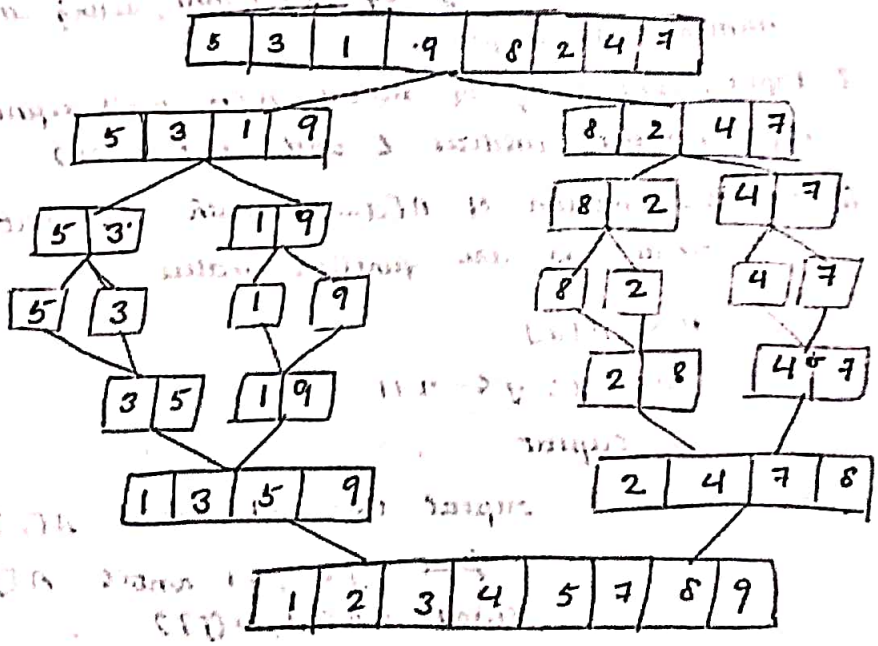
```
algorithm merge (B[0... p-1], C[0... q-1], A[0... p+q-1])
// merge 2 two sorted arrays into one sorted array.
// input : arrays B[0... p-1] and C[0... q-1] both sorted.
```

2 marks

```
// output: sorted array A[0... p+q-1] of the elements of B and C.
    i ← 0; j ← 0; k ← 0
    while i<p and j<q do
        if B[i] ≤ C[j]
            A[k] ← B[i]
            i ← i+1
        else
            A[k] ← C[j]
            j ← j+1
            k ← k+1
    if i=p
        copy C[j... q-1] to A[k... p+q-1]
    else
        copy B[i... p-1] to A[k... p+q-1]
```

3 (b)

2 marks



2 marks

---

6 marks

4	(b)	P = 5 0 1 2 3 4 5 6 7 5 3 1 9 8 2 4 7 5 3 1 4 8 2 9 7 5 3 1 4 2 8 9 7 P = 2 2 3 1 4 5 8 9 7 2 3 1 4 2 1 3 4 1 2 3 4 3 4 P = 8 8 9 7 8 7 9 7 8 9 1 2 3 4 5 7 8 9	3 marks 3 marks --- 6 marks

USN | | | | | | | | | | |

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

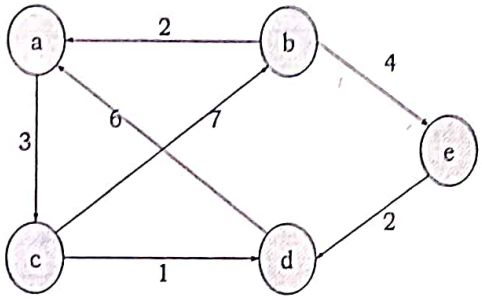
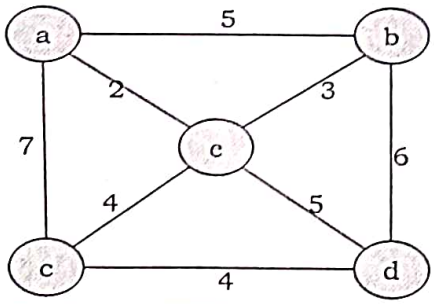
(Accredited by NAAC with A+ Grade)

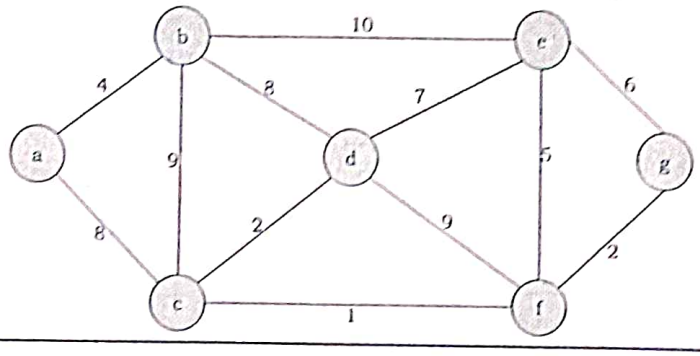
Department of Computer Science & Design

Continuous Internal Evaluation Test – II AY 2023-24

Course Title : Analysis & Design of Algorithms		Course Code: BCS401
Date: 08/07/2024	Time:9.30 AM -11.00 AM	Semester/Section: IV
Faculty: Dr. Shivaprasad B J		Max. Marks:25

Note: Answer ONE FULL question from each part.

Q. No.		Questions	Marks	COs	BTL
		PART – I			
1	a)	Construction of the AVL Tree for the given Sequence 10 15 9 12 13 79 45 36 22.	7	CO3	L4
	b)	Write an algorithm for sort using Distribution Count and apply it for 10 15 9 12 13 79 45 36 22	6	CO3	L3
		OR			
2	a)	Sort the given list of numbers using heapsort: 5,3,1,9,8,2,4,7	7	CO3	L4
	b)	Write Harspoo's algorithm for string matching. Find the Pattern BARBER in the textJIM_SAW_ME_IN_A_BARBERSHOP.	6	CO3	L3
		PART-II			
3	a)	Define all pair shortest path. Write Floyd's algorithm to solve all pair shortest path and apply Floyd's algorithm to the following graph. 	6	CO3	L3
	b)	Define Minimum Cost Spanning Tree, Write Prim's algorithm to find Minimum Cost Spanning Tree and apply Prim's algorithm to the following graph. 	6	CO4	L4
		OR			
4	a)	Write Knapsack Problem algorithm using dynamic programming. Using dynamic programming solve the following knapsack n=4, Capacity W=5, Weight (W1,W2,W3,W4)=(2,1,3,2), Profit (P1,P2,P3,P4)=(8,6,16,11).	6	CO3	L3

	b)	Define Single Source Shortest Path Algorithm. Write Dijkstra's algorithm to find Single Source Shortest Path and apply Dijkstra's algorithm to the following graph. 	6	CO4	L4

Levels of Bloom's Taxonomy

No.	L1	L2	L3	L4	L5	L6
Level	Remember	Understand	Apply	Analyze	Evaluate	Create

Course Outcomes

CO1	Apply asymptotic notational method to analyze the performance of the algorithms in terms of time complexity.
CO2	Demonstrate divide & conquer approaches and decrease & conquer approaches to solve computational problems.
CO3	Make use of transform & conquer and dynamic programming design approaches to solve the given real world or complex computational problems.
CO4	Apply greedy and input enhancement methods to solve graph & string based computational problems.
CO5	Analyse various classes (P,NP and NP Complete) of problems
CO6	Illustrate backtracking, branch & bound and approximation methods.

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(Accredited by NAAC with A+ Grade)

QUESTION PAPER REVIEW REPORT

Continuous Internal Evaluation (CIE) Test: II AY 2023-24

Department : CSD Semester/Section: 4 (A) Max Marks: 25

Course Title: ADA Course Code (BCS401) Date: 6/07/2024

Faculty: Dr. Shivaprasad B J

Qn. No.	Course Outcome (CO)	Bloom's Taxonomy Level	Marks
1a	CO3	L4	7
1b	CO3	L3	6
2a	CO3	L4	7
2b	CO3	L3	6
3a	CO3	L3	6
3b	CO4	L4	6
4a	CO3	L3	6
4b	CO4	L4	6
Total Marks			**50**

BT Level: L1-Remember, L2-Understand, L3 -Apply, L4 -Analyze, L5- Evaluate, L6- Create

Consolidated Marks for Different BT Levels:

BT Level	Marks for Each Level	% of Marks	Remarks
L3	6	48	
L4	7	52	

Scrutinizer/Reviewer Remark:

Approved	*Approved*	Approved with Correction		Rejected	
Reason for Rejection					

06/07/24

Dr. Shivaprasad B J

Name & Signature of the Scrutinizer

Date: 06/07/2024

Mr. Jayantkumar A Rathod

Name & Signature of the IQAC Coordinator

Date: 06/07/2024

Mr. Jayantkumar A Rathod

Signature of Head of the Department

Alva's Institute of Engineering & Technology, Moodbidri

(Accredited by NAAC with A+ Grade)

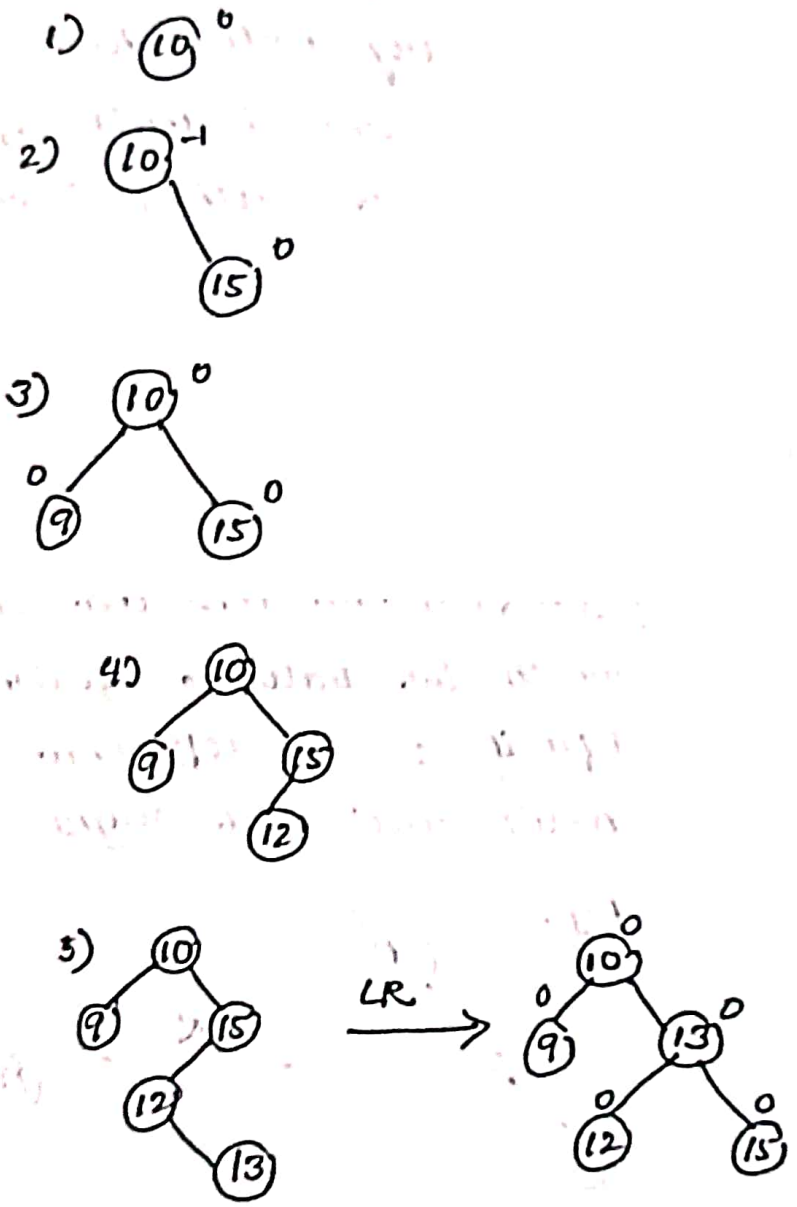
Department of Computer Science & Design

Scheme of Evaluation - ODD Semester 2023-24

Continuous Internal Evaluation Test-II

Course Title: **Analysis & Design of Algorithms**		Course Code: BCS401
Date: 08/07/2024	Time: 9.30 AM- 11.00AM	Semester: V
Faculty: Dr. Shivaprasad B J		Max. Marks: 20

Note: Answer ONE FULL question from each Module.

Q. No.		Scheme of Evaluation	Marks
1	(a)		3 marks 4 marks ---------- 7 marks

1 (b)

-> ALGORITHM DistributionCountingSort (A[0.. n-1], l, u)

// sorts an array of integers from a limited range by distribution counting.

// Input: An array A[0... n-1] of integers between l and u (l ≤ u)

// Output: Array S[0... n-1] of A's elements sorted in non-decreasing order.

```
for j ← 0 to u-l do D[j] ← 0
for i ← 0 to n-1 do D[A[i]-l] ← D[A[i]-l]+1
for i ← n-1 column to 0 do
    j ← A[i]-l
    S[D[j]-1] ← A[i]
    D[j] ← D[j]-1
return S
```

3 marks

Array	9	10	12	13	15	22	36	45	79
frequency	1	1	1	1	1	1	1	1	1
Distribution value	1	2	3	4	5	6	7	8	9

	9	10	12	13	15	22	36	45	79
	1	2	3	4	5	6	7	8	9
A[8] = 22	1	2	3	4	5	5	7	8	9
A[7] = 36	1	2	3	4	5	5	6	8	9
A[6] = 45	1	2	3	4	5	5	6	7	9
A[5] = 79	1	2	3	4	5	5	6	7	8
A[4] = 13	1	2	3	3	5	5	6	7	8
A[3] = 12	1	2	2	3	5	5	6	7	8
A[2] = 9	0	2	2	3	5	5	6	7	8
A[1] = 15	0	2	2	3	4	5	6	7	8
A[0] = 10	0	1	2	3	4	5	6	7	8

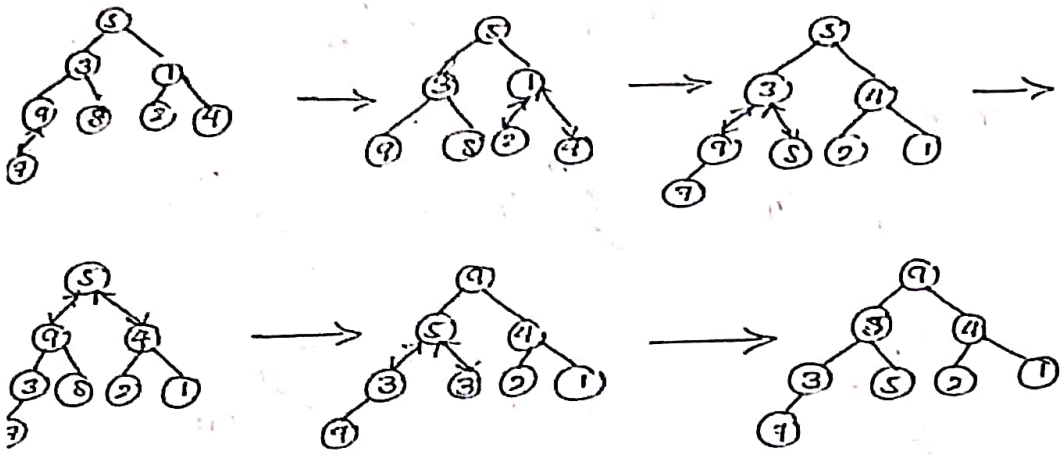
0	1	2	3	4	5	6	7	8
9	10	12	13	15	22	36	45	79

3 marks

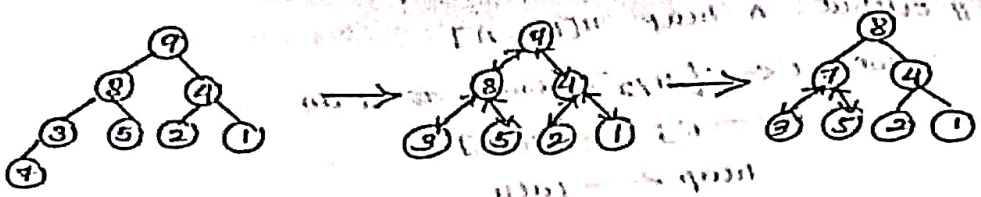
---

6 marks

2 (a)

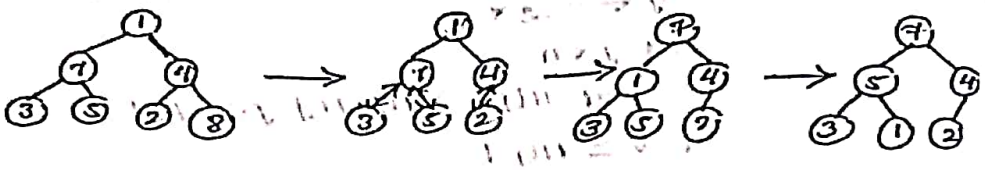


Step 2 : Deletion

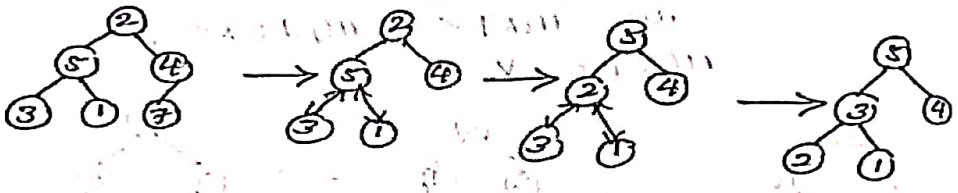


4 marks

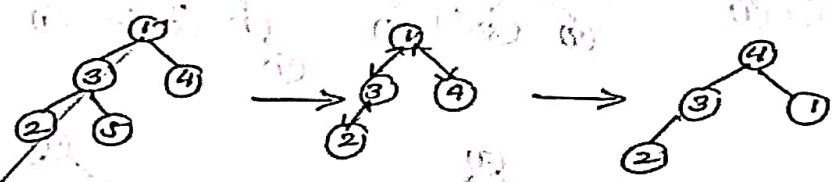
Step 3 :



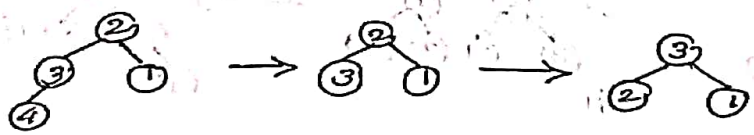
Step 4 :



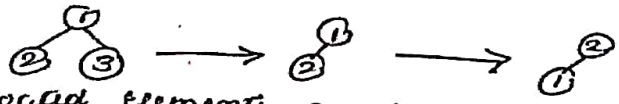
Step 5 :



Step 6 :



Step 7 :



∴ Sorted elements are 1, 2, 3, 4, 5, 7, 8, 9.

3 marks

---

7 marks

2	(b)		3 marks

character	A	B	C	D	E	K	Z	–
freq	4	2	6	6	1	9	6	6

m - 1 - d

B = 6 - 1 - 0 = 5

A = 6 - 1 - 1 = 4

R = 6 - 1 - 2 = 3

D = 6 - 1 - 3 = 2

E = 6 - 1 - 4 = 1

R = 6 - 1 - 5 = 0

V	I	M	–	S	A	W	–	M	E	–	I	N	–	A	–	B	A	R	B	E	R	S	H	O	P
B	A	R	B	E	R																				
				B	A	R	B	E	R																
					B	A	R	B	E	R															
											B	A	R	B	E	R									
													B	A	R	B	E	R							
																B	A	R	B	E	R				

3 marks

---

6 marks

3	(a)	**Algorithm:**	3 marks

```
Floyd(W[1...n, 1...n])
// Implements Floyd's Algorithm for all-pairs shortest-path
   problem.
// Input: The weight matrix W of a shortest path lengths
// output: The distance matrix of the shortest path length.
    D ← W // is not necessary if W can be overwritten.
    for k ← 1 to n do
      for i ← 1 to n do
        for j ← 1 to n do
            D[i,j] ← min{ D[i,j], D[i,k]+ D[k,j]}
    return D
```

$$W = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty & \infty \\ 2 & 0 & \infty & \infty & 4 \\ \infty & 7 & 0 & 1 & \infty \\ 6 & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 2 & 0 \end{bmatrix} \end{array}$$

3 marks

---

6 marks

$$W(1) = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty & \infty \\ 2 & 0 & 5 & \infty & 4 \\ \infty & 7 & 0 & 1 & \infty \\ 6 & \infty & 9 & 0 & \infty \\ \infty & \infty & \infty & 2 & 0 \end{bmatrix} \end{array}$$

$$W(2) = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty & \infty \\ 2 & 0 & 5 & \infty & 4 \\ 9 & 7 & 0 & 1 & 11 \\ 6 & \infty & 9 & 0 & \infty \\ \infty & \infty & \infty & 2 & 0 \end{bmatrix} \end{array}$$

$$W(3) = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 0 & 10 & 3 & 4 & 14 \\ 2 & 0 & 5 & 6 & 4 \\ 9 & 7 & 0 & 1 & 11 \\ 6 & 16 & 9 & 0 & 20 \\ \infty & \infty & \infty & 2 & 0 \end{bmatrix} \end{array}$$

$$W(4) = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 0 & 10 & 3 & 4 & 14 \\ 2 & 0 & 5 & 6 & 4 \\ 7 & 7 & 0 & 1 & 11 \\ 6 & 16 & 9 & 0 & 20 \\ 8 & 18 & 11 & 2 & 0 \end{bmatrix} \end{array}$$

algorithm Num (G)

// Prim's algorithm for constructing a minimum spanning tree

// Input: A weighted connected graph $G = \langle V, E \rangle$

// output: E_T the set of edges.

$V_T \leftarrow \{v_0\}$

$E_T \leftarrow \emptyset$

for $i \leftarrow 1$ to $|V| - 1$ do

find a minimum-weight-edge $e^* = (v^*, u^*)$ among the edges (v, u) such that v is in V_T & u is,

$V - V_T$

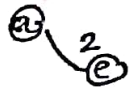
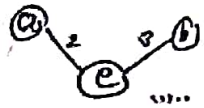
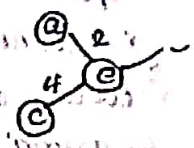
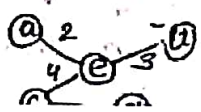
$V_T \leftarrow V_T \cup \{u^*\}$

$E_T \leftarrow E_T \cup \{e^*\}$

return E_T

3 marks

3 (b)

Tree vertex	Remaining vertex	Illustration
1] a(-,-)	b(a,5), c(a,7) e(a,2), d(-,∞)	
2] e(a,2)	b(a,3), c(e,4) d(e,5)	
3] b(e,3)	d(b,6), c(e,4)	
4] c(e,4)	d(c,4)	

3 marks

---

6 marks

∴ the minimum cost spanning tree is

$ab + eb + ec + cd = 2 + 3 + 4 + 4 = \underline{13}$

4 (a)

```
Algorithm (MF-knapsack [i,j])
// Implement the method of the knapsack problem.
// Input: Non-negative integer i indicating number of the
          first item and non-negative integer j
          indicating the capacity.
// output:
    if F[i,j] < 0
        if j < weight [i]
            value ← MFknapsack (i-1, j);
        else
            value ← max (MFknapsack (i-1, j);
                        values [i] + MFknapsack (i-1, j - weight
            F[i,j] ← value
        return F[i,j]
```

3 marks

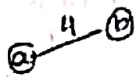
Item	weight	value
1	2	8
2	1	6
3	3	16
4	2	11

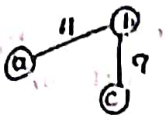
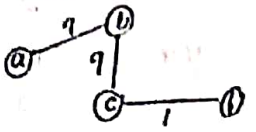
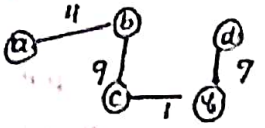
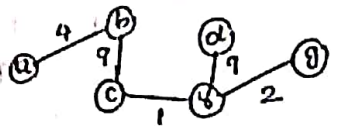
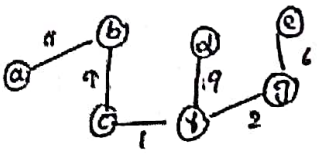
j - capacity = 5.

i \ j	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	8	8	8	8
2	0	6	8	14	14	14
3	0	6	8	16	22	24
4	0	6	11	17	22	27

3 marks

6 marks

4	(b)	Algorithm Dijkstra (G,s)
		// Dijkstra's algorithm for single-source shortest paths.
		// Input: A weighted connected graph G = <V,E> with non negative weights and its vertex s.
		// Output: The length d_v of a shortest path from s to v. and its penultimate vertex p_v for every vertex v in V.
		Initialize (Q)
		for every vertex v in V.
		$d_v \leftarrow \infty$; $p_v \leftarrow$ null
		Insert (Q, v, d_v)
		$d_s \leftarrow 0$; decrease (Q, s, d_s)
		$V_T \leftarrow \emptyset$
		for $i \leftarrow 0$ to $\lvert V \rvert - 1$ do
		$u^* \leftarrow$ delete min(Q)
		$V_T \leftarrow V_T \cup \{u^*\}$
		for every vertex u in V − V_T that is adjacent to u^* do
		if $d_{u^*} + w(u^*, u) < d_u$.
		$d_u \leftarrow d_{u^*} + w(u^*, u)$; $p_u \leftarrow u^*$
		decrease (Q, u, d_u).
		
		Tree vertex: a
		Remaining vertices: b, c, d, e, f, g
		d[b] = 4 → min
		d[c] = 8.
		d[d] = ∞
		
		 
		
			3 marks
			3 marks
			--- 	
			6 marks

	$d[c] = \infty$ $d[f] = \infty$ $d[g] = \infty$	
a,b	c, d, e, f, g $d[c] = min(8, 4+9) = 8 = min$ $d[d] = min(\infty, 4+8) = 12$ $d[e] = min(\infty, 4+10) = 14$ $d[f] = min(\infty, 4+\infty) = \infty$ $d[g] = min(\infty, 4+\infty) = \infty$	
a,b,c	d, e, f, g $d[d] = min(12, 8+2) = 10$ $d[e] = min(14, 8+6) = 14$ $d[f] = min(\infty, 8+1) = 9$ $d[g] = min(\infty, 8+\infty) = \infty$	
a,b,c,f	d, e, g $d[d] = min(10, 9+\infty) = 10$ $d[e] = min(14, 9+5) = 14$ $d[g] = min(\infty, 9+2) = 14$	
a,b,c,f,d	e, g $d[e] = min(14, 10+7)$ $= 14$ $d[g] = min(14, 10+2) = 11$	
a,b,c,f,d,g	e $d[e] = min(14, 11+6)$ $= 14$	

USN | | | | | | | | | | |

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

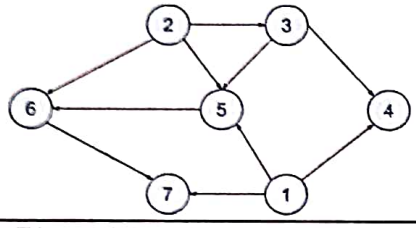
(Accredited by NAAC with A+ Grade)

Department of Computer Science & Design

Continuous Internal Evaluation Test – III AY 2023-24

Course Title : Analysis & Design of Algorithms		Course Code: BCS401
Date: 26/07/2024	Time:3.00 PM-4.30 PM	Semester/Section: IV
Faculty: Dr. Shivaprasad B J		Max. Marks:25

Note: Answer ONE FULL question from each part.

Q. No.		Questions	Marks	COs	BTL
		PART – I			
1	a)	Apply strassen's algorithm for matrix multiplication to multiply the following matrices and justify how the strassen's algorithm is better. $\begin{bmatrix} 4 & 3 \\ 1 & 2 \end{bmatrix} * \begin{bmatrix} 1 & 2 \\ 6 & 5 \end{bmatrix}$	7	CO2	L4
	b)	Explain N-Queens problem with an example.	6	CO5	L3
		OR			
2	a)	Obtain the topological sort for the graph using (1) Source Removal Method (2) DFS Method. 	7	CO2	L4
	b)	Explain Binary Tree Traversals.	6	CO2	L3
		PART-II			
3	a)	Apply Branch and Bound approach to solve the instance of 0/1 Knapsack problem. Knapsack Capacity W=10	6	CO6	L3

Items	1	2	3	4
Weights	4	7	5	3
Value	40	42	25	12

	b)	Explain Decision trees for sorting and searching .	6	CO6	L4
		OR			
4	a)	Apply a backtracking method to solve the sum of subset problems for the instance d=30 and s={5,10,12,13,15,18} .	6	CO6	L3
	b)	Write a note P, NP, and NP-Complete Problems.	6	CO5	L4

Levels of Bloom's Taxonomy

No.	L1	L2	L3	L4	L5	L6
Level	Remember	Understand	Apply	Analyze	Evaluate	Create

Course Outcomes

CO1	Apply asymptotic notational method to analyze the performance of the algorithms in terms of time complexity.
CO2	Demonstrate divide & conquer approaches and decrease & conquer approaches to solve computational problems.
CO3	Make use of transform & conquer and dynamic programming design approaches to solve the given real world or complex computational problems.
CO4	Apply greedy and input enhancement methods to solve graph & string based computational problems.
CO5	Analyse various classes (P,NP and NP Complete) of problems
CO6	Illustrate backtracking, branch & bound and approximation methods.

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(Accredited by NAAC with A+ Grade)

QUESTION PAPER REVIEW REPORT

Continuous Internal Evaluation (CIE) Test: III AY 2023-24

Department : CSD Semester/Section: 4 (A) Max Marks: 25

Course Title: ADA Course Code (BCS401) Date: 25/07/2024

Faculty: Dr. Shivaprasad B J

Qn. No.	Course Outcome (CO)	Bloom's Taxonomy Level	Marks
1a	CO2	L4	7
1b	CO5	L3	6
2a	CO2	L4	7
2b	CO5	L3	6
3a	CO6	L3	6
3b	CO6	L4	6
4a	CO6	L3	6
4b	CO3	L4	6
Total Marks			**50**

BT Level: L1-Remember, L2-Understand, L3 -Apply, L4 -Analyze, L5- Evaluate, L6- Create

Consolidated Marks for Different BT Levels:

BT Level	Marks for Each Level	% of Marks	Remarks
L3	6	48	
L4	7 /6	52	

Scrutinizer/Reviewer Remark:

Approved	*Approved*	Approved with Correction		Rejected	
Reason for Rejection					

Dr. Shivaprasad B J

Name & Signature of the Scrutinizer

Date: 25/07/2024

Mr. Jayantkumar A Rathod

Name & Signature of the IQAC Coordinator

Date: 25/07/2024

Mr. Jayantkumar A Rathod

Signature of Head of the Department

Alva's Institute of Engineering & Technology, Moodbidri

(Accredited by NAAC with A+ Grade)

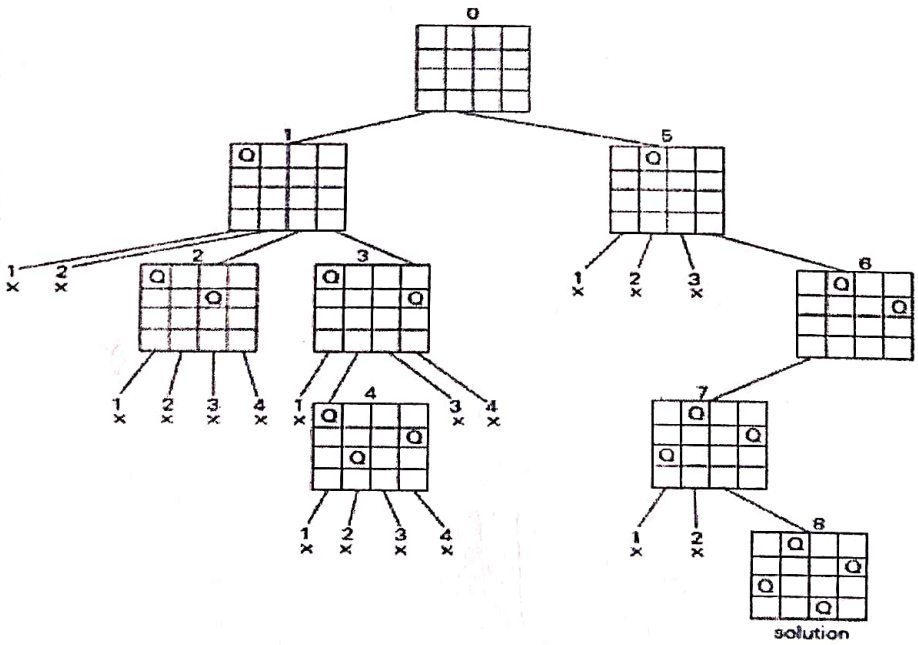
Department of Computer Science & Design

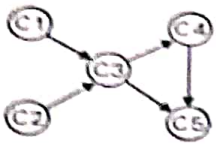
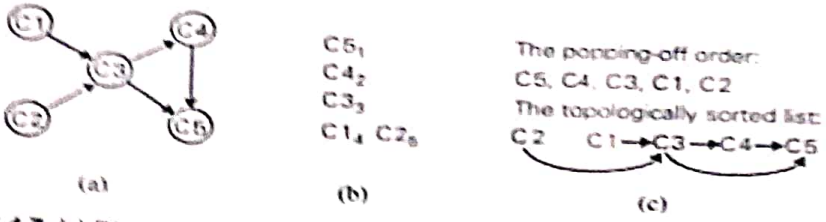
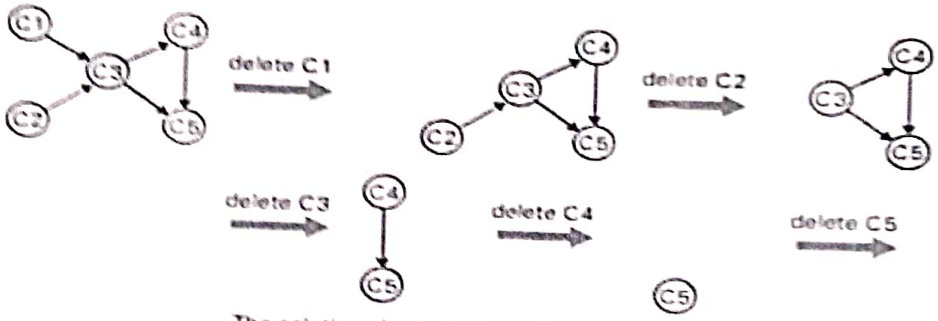
Scheme of Evaluation - ODD Semester 2023-24

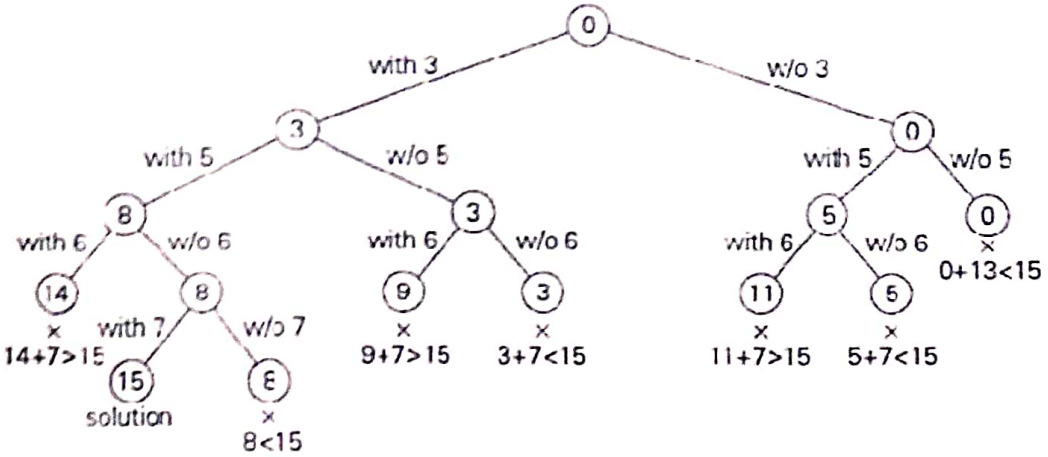
Continuous Internal Evaluation Test-III

Course Title: **Analysis & Design of Algorithms**		Course Code: BCS401
Date: 26/07/2024	Time: 9.30 AM- 11.00AM	Semester: V
Faculty: Dr. Shivaprasad B J		Max. Marks: 20

Note: Answer ONE FULL question from each Module.

Q. No.		Scheme of Evaluation	Marks
1	(a)	$$\begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} * \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix}$$ $$= \begin{bmatrix} m_1 + m_4 - m_5 + m_7 & m_3 + m_5 \\ m_2 + m_4 & m_1 + m_3 - m_2 + m_6 \end{bmatrix}$$ where	4 marks
		$m_1 = (a_{00} + a_{11}) * (b_{00} + b_{11})$, $m_2 = (a_{10} + a_{11}) * b_{00}$, $m_3 = a_{00} * (b_{01} - b_{11})$, $m_4 = a_{11} * (b_{10} - b_{00})$, $m_5 = (a_{00} + a_{01}) * b_{11}$, $m_6 = (a_{10} - a_{00}) * (b_{00} + b_{01})$, $m_7 = (a_{01} - a_{11}) * (b_{10} + b_{11})$.	3 marks
			---------- 7 marks
1	(b)		3 marks
			3 marks
			---------- 6 marks

2	(a)	In terms of this digraph, the question is whether we can list its vertices in such an order that for every edge in the graph, the vertex where the edge starts is listed before the vertex where the edge ends. This problem is called topological sorting.  **FIGURE 4.6** Digraph representing the prerequisite structure of five courses.  **FIGURE 4.7** (a) Digraph for which the topological sorting problem needs to be solved. (b) DFS traversal stack with the subscript numbers indicating the popping-off order. (c) Solution to the problem.  **FIGURE 4.8** Illustration of the source-removal algorithm for the topological sorting problem. On each iteration, a vertex with no incoming edges is deleted from the digraph.	4 marks 3 marks 7 marks
2	(b)	Inorder Preorder Postorder	3 marks 3 marks 6 marks

3	(a)		3 marks 3 marks ---------- 6 marks

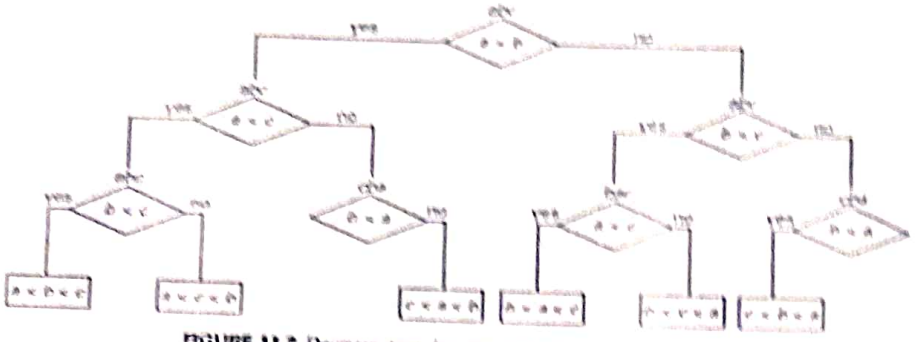


FIGURE 11.2 Decision tree for the three-element selection sort. A node above a node indicates the state of the array being sorted. Note two redundant comparisons $b < a$ with a single possible outcome because of the results of some previously made comparisons.

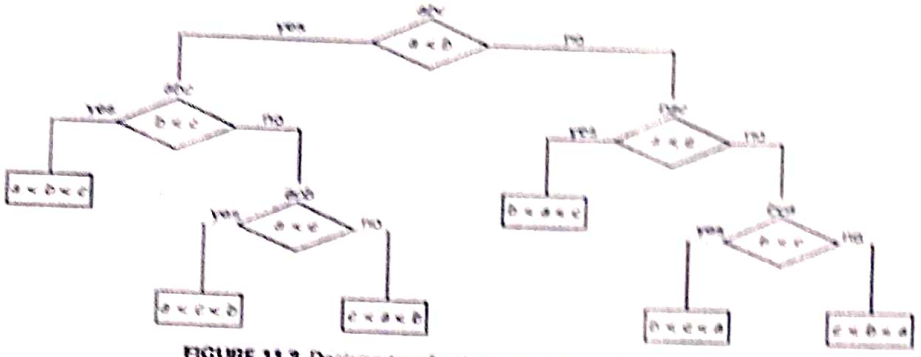


FIGURE 11.3 Decision tree for the three-element insertion sort.

3 (b) 3 marks

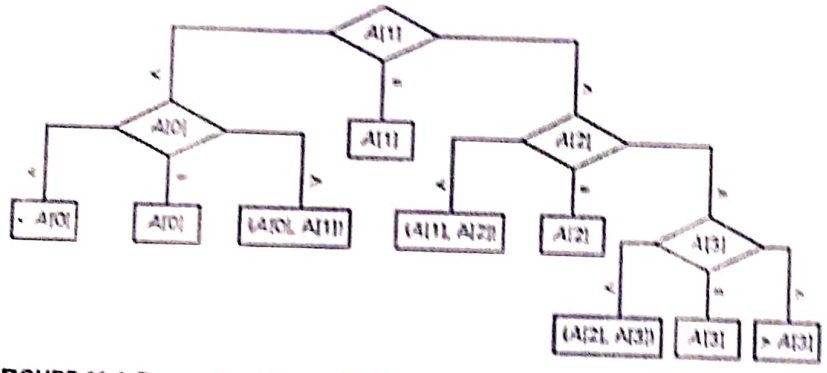


FIGURE 11.4 Ternary decision tree for binary search in a four-element array.

3 marks

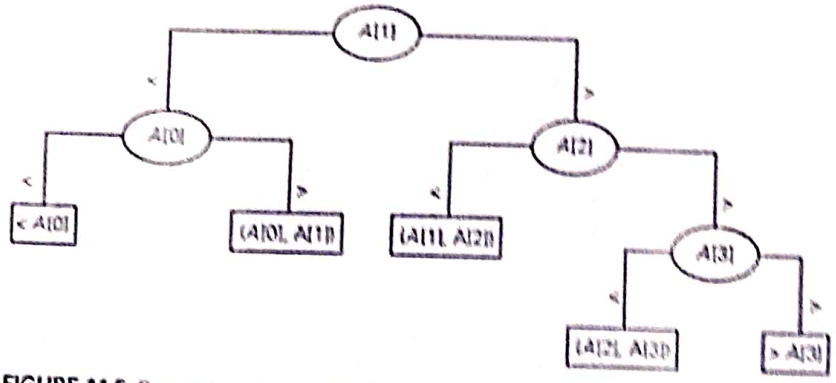
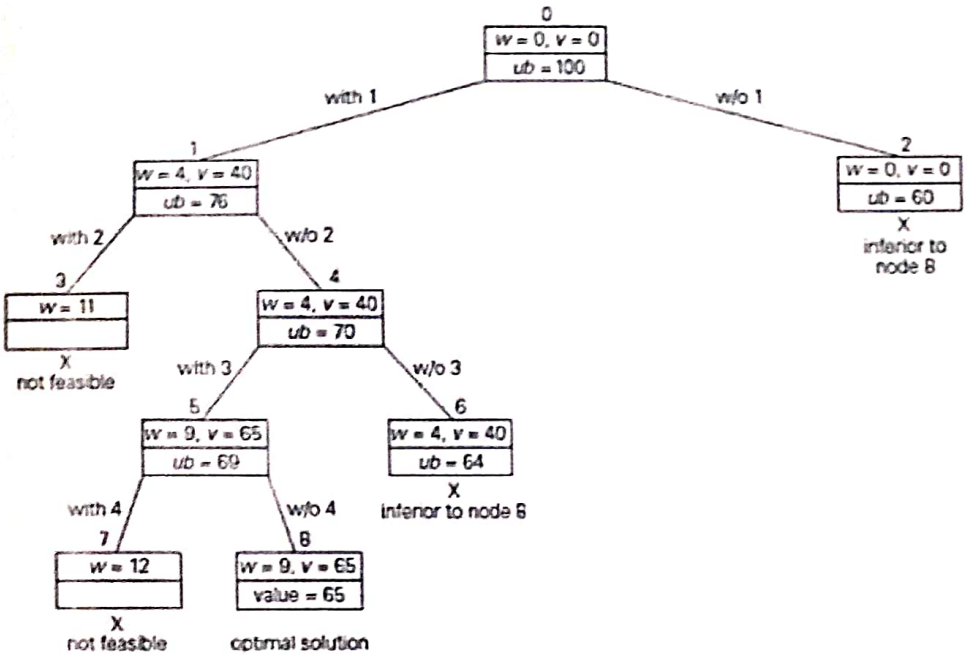
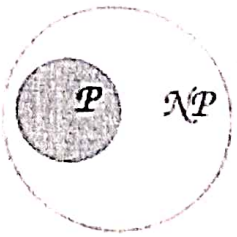
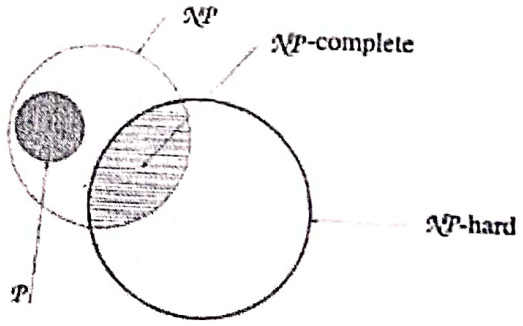


FIGURE 11.5 Binary decision tree for binary search in a four-element array.

6 marks

4	(a)	 **FIGURE 12.8** State-space tree of the best-first branch-and-bound algorithm for the instance of the knapsack problem.	3 marks 3 marks ---------- 6 marks

4	(b)	**Definition:** P is a set of all decision problems solvable by a deterministic algorithm in polynomial time. **Definition:** NP is the set of all decision problems solvable by a nondeterministic algorithm in polynomial time. This also implies P ⊆ NP Problems known to be in P are trivially in NP — the nondeterministic machine just never troubles itself to fork another process, and acts just like a deterministic one. One example of a problem not in P but in NP is ***Integer Factorization.*** But there are some problems which are known to be in NP but don't know if they're in P. The traditional example is the decision-problem version of the Travelling Salesman Problem (***decision-TSP***). It's not known whether decision-TSP is in P: there's no known poly-time solution, but there's no proof such a solution doesn't exist. There are problems that are known to be neither in P nor NP; a simple example is to enumerate all the bit vectors of length n. No matter what, that takes 2^n steps. Now, one more concept: given decision problems P and Q, if an algorithm can transform a solution for P into a solution for Q in polynomial time, it's said that Q is ***poly-time reducible*** (or just reducible) to P. The most famous unsolved problem in computer science is "whether P=NP or P≠NP? "  Figure: Commonly believed relationship between P and NP  Figure: Commonly believed relationship between P, NP, NP-Complete and NP-hard problems	3 marks 3 marks --- 6 marks

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

RUBRICS TO BE FOLLOWED FOR ASSIGNMENT EVALUATION

Parameters	Not Submitted	Incomplete Assignment	Correctness of the problem/ program	Timely submitted with correctness of the Problem/ Program
Marks allocation	0	3-9	10-12	12.5

Subject Teacher

HoD

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY
A Unit of Alva's Education Foundation (R)
(Affiliated to Visvesvaraya Technological University, Belagavi.
Approved by AICTE, New Delhi & & Recognized by Government of Karnataka)
Shobhavana Campus, Mijar, Moodbidri- 574 225, Mangalore, D.K., Karnataka State.
(Accredited by NAAC with A+ Grade)

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

Assignment No: 1

Subject: Analysis & Design of Algorithms (BCS401) Sem:4
Faculty: Dr. Shivaprasad B J Date:30/05/2024

1. Explain fundamentals of algorithmic problem solving.
2. What are the various basic Asymptotic efficiency classes?. Explain Big-O, Big-Ω, Big- θ with an example.
3. Give a mathematical analysis of non-recursive matrix multiplication algorithms.
4. Give the general plan for analyzing time efficiency of recursive algorithms and also analyse the tower of hanoi recursive algorithms.
5. Write an algorithm to find uniqueness of elements in an array and give the mathematical analysis of this non recursive algorithm with steps.
6. Write an algorithm for Selection Sort and mention the time complexity.
7. Sort the given below given array of elements using Bubble Sort and mention the time complexity.
   4, 9 , 0, 6, 8, 9, 2, 3, 12
8. Design an algorithm for Insertion Sort and apply the same to sort the following input 4, 9 , 0, 6, 8, 9, 2, 3, 12.
9. Design an algorithm for Merge Sort and apply the same to sort the following input 5, 3, 1, 9, 8, 2, 4, 7.
10. Design an algorithm for Quick Sort and apply the same to sort the following input 5, 3, 1, 9, 8, 2, 4, 7.

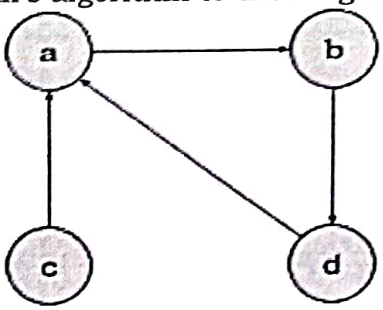
ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY
A Unit of Alva's Education Foundation (R)
(Affiliated to Visvesvaraya Technological University, Belagavi.
Approved by AICTE, New Delhi & & Recognized by Government of Karnataka)
Shobhavana Campus, Mijar, Moodbidri- 574 225, Mangalore, D.K., Karnataka State.
(Accredited by NAAC with A+ Grade)

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

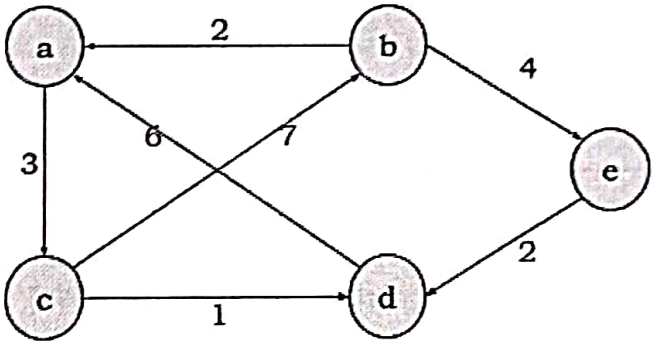
Assignment No: 2

Subject: Analysis & Design of Algorithms (BCS401) Sem:4
Faculty: Dr. Shivaprasad B J Date:01/07/2024

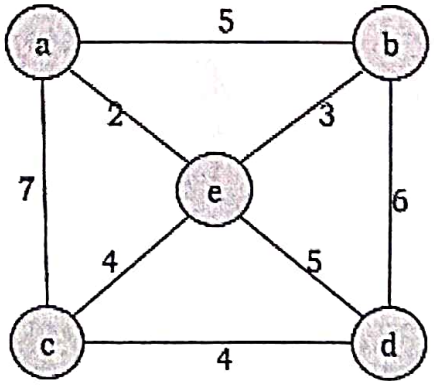
1. Define transitive closure. Write warshall's algorithm to compute transitive closure and apply warshall's algorithm to the diagraph given below.



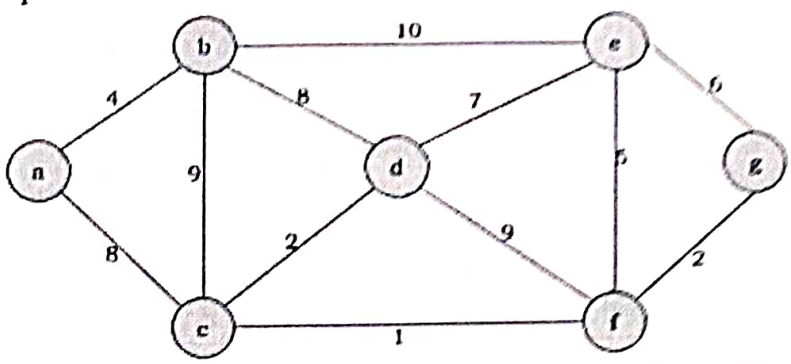
2. Define all pair shortest path. Write Floyd's algorithm to solve all pair shortest path and apply Floyd's algorithm to the following graph.



3. Write Knapsack Problem algorithm using dynamic programming. Using dynamic programming solve the following knapsack n=4, Capacity W=5, Weight (W1,W2,W3,W4)=(2,1,3,2), Profit (P1,P2,P3,P4)=(8,6,16,11).
4. Define Minimum Cost Spanning Tree, Write Prim's and Kruskal's algorithm to find Minimum Cost Spanning Tree and apply Prim's and Kruskal's algorithm to the following graph.



5. Define Single Source Shortest Path Algorithm. Write Dijkstra's algorithm to find Single Source Shortest Path and apply Dijkstra's algorithm to the following graph.



6. Define Huffman trees. Construct a Huffman tree and code for the following data.

Symbol	A	B	C	D	E
Frequency	22	13	18	16	31
Encode	CABADDBADACE				
Decode	110011 and 1000110001				

7. Sort the given list of numbers using heapsort: 5,3,1,9,8,2,4,7
8. Write the algorithm for the bottom up algorithm to construct the heap and construct the heap for the list 5,3,1,9,8,2,4,7.
9. Construction of the AVL Tree for the given Sequence 10 15 9 12 13 79 45 36 22.
10. Explain the AVL tree with an example.
11. Write an algorithm for sort using Comparison Count and apply it for 10 15 9 12 13 79 45 36 22
12. Write an algorithm for sort using Distribution Count and apply it for 10 15 9 12 13 79 45 36 22
13. Write Harspoo's algorithm for string matching. Find the Pattern BARBER in the textJIM_SAW_ME_IN_A_BARBERSHOP.

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY
A Unit of Alva's Education Trust®, Moodbidri
(Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE, New Delhi)
Shobhavana Campus, Mijar, Moodbidri- 574 225, Mangalore, D.K., Karnataka State.
(Accredited by NAAC with A+ Grade & Accredited by NBA New Delhi (CSE & ECE))

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

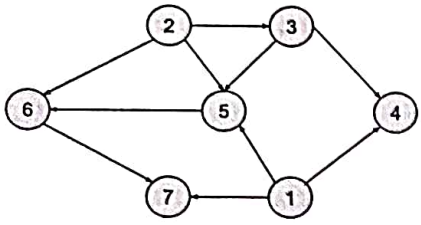
Assignment No: 3

Subject: Analysis & Design of Algorithms (BCS401) Sem:4
Faculty: Dr. Shivaprasad B J Date:24/07/2024

1. Apply strassen's algorithm for matrix multiplication to multiply the following matrices and justify how the strassen's algorithm is better.

$$\begin{bmatrix} 4 & 3 \\ 1 & 2 \end{bmatrix} * \begin{bmatrix} 1 & 2 \\ 6 & 5 \end{bmatrix}$$

2. Explain Binary Tree Traversals.
3. Obtain the topological sort for the graph using (1) Source Removal Method (2) DFS Method.



4. Apply Branch and Bound approach to solve the instance of 0/1 Knapsack problem. Knapsack Capacity W=10.

Items	1	2	3	4
Weights	4	7	5	3
Value	40	42	25	12

5. Apply a backtracking method to solve the sum of subset problems for the instance d=30 and s={5,10,12,13,15,18} .
6. Explain N-Queens problem with an example.
7. Explain Decision trees for sorting and searching .
8. Write a note P, NP, and NP-Complete Problems.

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY
A Unit of Alva's Education Trust®, Moodbidri
(Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE, New Delhi)
Shobhavana Campus, Mijar, Moodbidri- 574 225, Mangalore, D.K., Karnataka State.
(Accredited by NAAC with A+ Grade & Accredited by NBA New Delhi (CSE & ECE))

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN
Analysis & Design of Algorithms [BCS401]
Average IA Marks

Sl. No.	USN	NAME	Attendance %	IA1	IA2	IA3	Assignment Marks	Average Marks
				25	25	25	25	50
1	4AL22CG001	ABAYCHANDRASURYA J K	91	16	23	25	25	49
2	4AL22CG002	ALOK G	92	20	17	25	24	47
3	4AL22CG003	ANANT R KULKARNI	85	4	4	20	12	24
4	4AL22CG004	ANVESH A SHETTY	98	25	20	24	25	50
5	4AL22CG005	APEKSHA K	89	18	14	21	25	45
6	4AL22CG006	ASHWINI	96	AB	19	24	25	47
7	4AL22CG007	BIBIFATIMA	85	16	20	25	24	47
8	4AL22CG008	CHANDRASHEKHAR P C	98	18	19	25	24	46
9	4AL22CG009	CHANNAKESHAVASHREE D R	92	4	19	19	24	43
10	4AL22CG010	CHIDVILAS N	89	14	19	25	24	46
11	4AL22CG011	DARSHAN HALAKERIMATH	85	21	23	24	24	48
12	4AL22CG012	DARSHAN RAGHURAM SHETTY	86	21	18	19	24	44
13	4AL22CG013	DARSHAN S GOWDA	91	16	14	16	24	40
14	4AL22CG014	DHANRAJ S H	89	25	23	24	25	50
15	4AL22CG015	DHANUSHREE N	94	17	21	14	23	42
16	4AL22CG016	DHANYASHREE	89	17	24	25	25	50
17	4AL22CG017	DRUSHYA HEGDE D	85	AB	10	25	23	41
18	4AL22CG018	FAHAD UMAR FAROOQ	89	10	13	25	23	42
19	4AL22CG019	GANESH	94	15	15	23	25	44
20	4AL22CG020	GOUSIYA G	85	24	AB	25	25	50
21	4AL22CG021	HANAMANT RANGAPPA PUJARI	98	10	16	25	25	46
22	4AL22CG022	HEMANTH K V	85	15	16	25	24	45
23	4AL22CG023	HITHASHREE	92	21	24	25	25	50
24	4AL22CG024	JAGADISH GOWDA B J	85	14	6	25	24	44
25	4AL22CG025	KARTIK MYAGERI	92	13	14	19	24	41
26	4AL22CG026	KATHANIKA MOGERAYA	91	7	18	23	24	45
27	4AL22CG027	KHUSHI SHETTY	94	17	21	25	24	47
28	4AL22CG028	KOMAL KALIKAR	96	19	25	25	25	50
29	4AL22CG029	MAHAMMED ASIM	92	15	19	25	25	47
30	4AL22CG030	MANJUNATH	85	11	19	22	22	43
31	4AL22CG031	MOHAMMED IRFAN	85	4	10	20	24	39
32	4AL22CG032	MOUNASHREE D P	91	25	25	25	25	50

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

A Unit of Alva's Education Trust®, Moodbidri
(Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE, New Delhi)
Shobhavana Campus, Mijar, Moodbidri- 574 225, Mangalore, D.K., Karnataka State.
(Accredited by NAAC with A+ Grade & Accredited by NBA New Delhi (CSE & ECE))

33	4AL22CG033	NAGARAJ DADDI	80	20	6	15	22	40
34	4AL22CG034	NAVEEN HIREMATH	89	25	10	24	25	50
35	4AL22CG035	NIKHITH NAIK	85	10	12	18	24	39
36	4AL22CG036	NIRIKSHA RAI	91	16	15	21	24	43
37	4AL22CG037	NISHCHAL	89	24	13	21	24	47
38	4AL22CG038	PARESH PRAKASH TALEKAR	92	20	10	24	24	46
39	4AL22CG039	PAVAN RAJ	94	25	15	25	25	50
40	4AL22CG040	POOJARI SURAKSHA SURENDRA	89	21	22	25	24	48
41	4AL22CG041	POONAM NAIR	92	20	11	25	24	47
42	4AL22CG042	PRADEEP U NAIK	96	24	21	19	25	48
43	4AL22CG043	PRANAMI M SHETTY	85	20	14	25	24	47
44	4AL22CG044	PRATHEEKSHA SHETTY	92	21	23	22	25	48
45	4AL22CG045	PREMKUMAR INGALE	91	19	12	25	20	42
46	4AL22CG046	PRIYA BHEEMARAJ WADAWADIGI	92	19	10	22	24	45
47	4AL22CG047	PRIYANKA IRANNA SAKHARE	85	21	10	17	25	44
48	4AL22CG048	RAKSHITHA	89	23	24	25	25	50
49	4AL22CG049	RASHMI M	96	24	20	25	25	50
50	4AL22CG050	SAI SANKARSH N S	76	21	7	22	24	46
51	4AL22CG051	SANIKA GOWDA N S	91	21	17	23	24	46
52	4AL22CG052	SEJAL RAJESH NAIK	87	16	0	25	24	45
53	4AL22CG053	SHAYANASHREE N	96	21	16	25	24	47
54	4AL22CG054	SHRUTHI N	98	25	25	25	25	50
55	4AL22CG055	SNEHA DATTATRYA GOUDA	85	25	13	25	25	50
56	4AL22CG056	SNEHA S	92	24	23	25	25	50
57	4AL22CG057	SOWJANYA S	94	24	22	25	25	50
58	4AL22CG058	SURAJ	94	14	10	19	25	42
59	4AL22CG059	TEJESVIN R	89	21	10	19	25	45
60	4AL22CG060	THANVI S SANIL	96	25	23	25	25	50
61	4AL22CG061	THIRUMALA	87	19	16	21	24	44
62	4AL22CG062	VAISHNAVI G	85	21	12	25	25	48

Faculty

HoD

ALVAS INSTITUTE OF ENGINEERING AND TECHNOLOGY MOODBIDRI

Branch : CG Semester : 4

Sl NO.	USN	BCS401
1	4AL22CG001	50
2	4AL22CG002	47
3	4AL22CG003	24
4	4AL22CG004	50
5	4AL22CG005	45
6	4AL22CG006	48
7	4AL22CG007	47
8	4AL22CG008	46
9	4AL22CG009	43
10	4AL22CG010	46
11	4AL22CG011	48
12	4AL22CG012	44
13	4AL22CG013	40
14	4AL22CG014	50
15	4AL22CG015	42
16	4AL22CG016	50
17	4AL22CG017	41
18	4AL22CG018	42
19	4AL22CG019	44
20	4AL22CG020	50
21	4AL22CG021	46
22	4AL22CG022	45
23	4AL22CG023	50
24	4AL22CG024	44
25	4AL22CG025	41
26	4AL22CG026	45
27	4AL22CG027	47
28	4AL22CG028	50
29	4AL22CG029	47
30	4AL22CG030	43
31	4AL22CG031	39
32	4AL22CG032	50
33	4AL22CG033	40
34	4AL22CG034	50
35	4AL22CG035	39
36	4AL22CG036	43

SI NO.	USN	BCS401
37	4AL22CG037	47
38	4AL22CG038	46
39	4AL22CG039	50
40	4AL22CG040	48
41	4AL22CG041	47
42	4AL22CG042	48
43	4AL22CG043	47
44	4AL22CG044	48
45	4AL22CG045	42
46	4AL22CG046	45
47	4AL22CG047	44
48	4AL22CG048	50
49	4AL22CG049	50
50	4AL22CG050	46
51	4AL22CG051	46
52	4AL22CG052	45
53	4AL22CG053	47
54	4AL22CG054	50
55	4AL22CG055	50
56	4AL22CG056	50
57	4AL22CG057	50
58	4AL22CG058	42
59	4AL22CG059	45
60	4AL22CG060	50
61	4AL22CG061	44
62	4AL22CG062	48

CBCS SCHEME

USN | 4 | A | L | 2 | 2 | C | S | 0 | 1 | 4 | **BCS401**

Fourth Semester B.E./B.Tech. Degree Examination, June/July 2024
Analysis and Design of Algorithms

Time: 3 hrs. Max. Marks: 100

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M : Marks , L: Bloom's level , C: Course outcomes.

		Module – 1	M	L	C
Q.1	a.	What is an algorithm? Explain the fundamentals of algorithmic problem solving.	10	L2	CO1
	b.	Develop an algorithm to search an element in an array using sequential search. Calculate the best case, worst case and average case efficiency of this algorithm.	10	L3	CO1
		OR			
Q.2	a.	Explain asymptotic notations with example.	10	L2	CO1
	b.	Give the general plan for analyzing the efficiency of the recursive algorithm. Develop recursive algorithm for computing factorial of a positive number. Calculate the efficiency in terms of order of growth.	10	L3	CO1
		Module – 2			
Q.3	a.	Explain Strassen's matrix multiplication approach with example and derive its time complexity.	10	L3	CO2
	b.	What is divide and conquer? Develop the quick sort algorithm and write its best case. Make use of this algorithm to sort the list of characters: E, X, A, M, P, L, E.	10	L2	CO2
		OR			
Q.4	a.	Distinguish between decrease & conquer and divide & conquer algorithm design techniques with block diagram. Develop insertion sort algorithm to sort a list of integers and estimate the efficiency.	10	L3	CO2
	b.	Define topological sorting. List the two approaches of topological sorting and illustrate with examples.	10	L2	CO2
		Module – 3			
Q.5	a.	Define AVL tree with an example. Give worst case efficiency of operations on AVL tree. Construct an AVL tree of the list of keys: 5, 6, 8, 3, 2, 4, 7 indicating each step of key insertion and rotation.	10	L3	CO3
	b.	Define Heap. Explain the bottom-up heap construction algorithm. Apply heap sort to sort the list of numbers 2, 9, 7, 6, 5, 8 in ascending order using array representation.	10	L3	CO3
		OR			
Q.6	a.	Define 2-3 tree. Give the worst case efficiency of operations on 2-3 tree. Build 2-3 tree for the list of keys 9, 5, 8, 3, 2, 4, 7 by indicating each step of key insertion and node splits.	10	L3	CO3
	b.	Design Horspool algorithm for string matching. Apply this algorithm to find the pattern BARBER in the text: JIM_SAW_ME_IN_A_BARBERSHOP	10	L3	CO3
		Module – 4			
Q.7	a.	Apply Dijkstra's algorithm to find the single source shortest path for given graph [Fig.Q7(a)] by considering 's' as source vertex. Illustrate each step.	10	L3	CO4

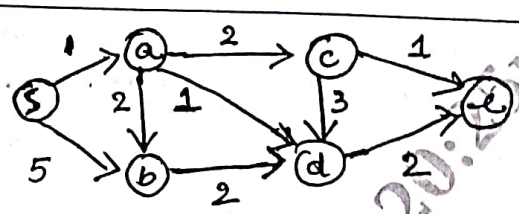


Fig.Q7(a)

	b.	Define transitive closure. Write Warshall's algorithm to compute transitive closure. Illustrate using the following directed graph.	10	L3	CO4

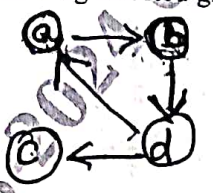


Fig.Q7(b)

OR

Q.8	a.	Define minimum spanning tree. Write Kruskal's algorithm to find minimum spanning tree. Illustrate with the following undirected graph.	10	L3	CO4

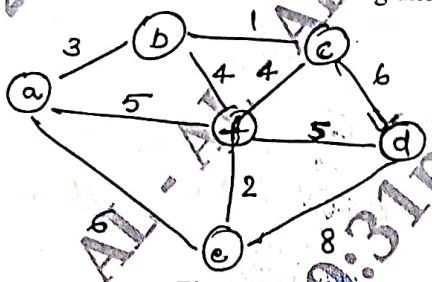


Fig.Q8(a)

	b.	Construct Huffman Tree and resulting code for the following:	10	L3	CO4

Character	A	B	C	D	E
Probability	0.4	0.1	0.2	0.15	0.15

(i) Encode the text : ABACABAD

(ii) Decode the text : 100010111001010

Module – 5

Q.9	a.	Explain n-Queen's problem with example using backtracking approach.	10	L2	CO5
	b.	Solve the following instance of the knapsack problem by the branch-and-bound algorithm. Construct state-space tree.	10	L3	CO5

Item	Weight	Value
1	4	$ 40
2	7	$ 42
3	5	$ 25
4	3	$ 12

The knapsack's capacity W is 10.

OR

Q.10	a.	Differentiate between Branch and Bound technique and Backtracking. Apply backtracking to solve the following instance of subset-sum problem S = {3, 5, 6, 7} and d = 15. Construct a state space tree.	10	L3	CO5
	b.	Explain greedy approximation algorithm to solve discrete knapsack problem.	10	L2	CO5

* * * * *

Modified

CBCS SCHEME

USN | | | | | | | | | | |

BCS401

Fourth Semester B.E./B.Tech. Degree Examination, June/July 2024
Analysis and Design of Algorithms

Time: 3 hrs. Max. Marks: 100

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M : Marks , L: Bloom's level , C: Course outcomes.

		Module – 1	M	L	C
Q.1	a.	What is an algorithm? Explain the fundamentals of algorithmic problem solving.	10	L2	CO1
	b.	Develop an algorithm to search an element in an array using sequential search. Calculate the best case, worst case and average case efficiency of this algorithm.	10	L3	CO1
		OR			
Q.2	a.	Explain asymptotic notations with example.	10	L2	CO1
	b.	Give the general plan for analyzing the efficiency of the recursive algorithm. Develop recursive algorithm for computing factorial of a positive number. Calculate the efficiency in terms of order of growth.	10	L3	CO1
		Module – 2			
Q.3	a.	Explain Strassen's matrix multiplication approach with example and derive its time complexity.	10	L3	CO2
	b.	What is divide and conquer? Develop the quick sort algorithm and write its best case. Make use of this algorithm to sort the list of characters: E, X, A, M, P, L, E.	10	L2	CO2
		OR			
Q.4	a.	Distinguish between decrease & conquer and divide & conquer algorithm design techniques with block diagram. Develop insertion sort algorithm to sort a list of integers and estimate the efficiency.	10	L3	CO2
	b.	Define topological sorting. List the two approaches of topological sorting and illustrate with examples.	10	L2	CO2
		Module – 3			
Q.5	a.	Define AVL tree with an example. Give worst case efficiency of operations on AVL tree. Construct an AVL tree of the list of keys: 5, 6, 8, 3, 2, 4, 7 indicating each step of key insertion and rotation.	10	L3	CO3
	b.	Define Heap. Explain the bottom-up heap construction algorithm. Apply heap sort to sort the list of numbers 2, 9, 7, 6, 5, 8 in ascending order using array representation.	10	L3	CO3
		OR			
Q.6	a.	Define 2-3 tree. Give the worst case efficiency of operations on 2-3 tree. Build 2-3 tree for the list of keys 9, 5, 8, 3, 2, 4, 7 by indicating each step of key insertion and node splits.	10	L3	CO3
	b.	Design Horspool algorithm for string matching. Apply this algorithm to find the pattern BARBER in the text: JIM_SAW_ME_IN_A_BARBERSHOP	10	L3	CO3
		Module – 4			
Q.7	a.	Apply Dijkstra's algorithm to find the single source shortest path for given graph [Fig.Q7(a)] by considering 's' as source vertex. Illustrate each step.	10	L3	CO4

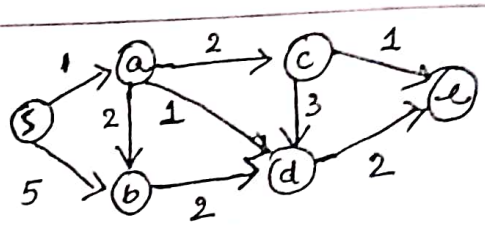


Fig.Q7(a)

	b.	Define transitive closure. Write Warshall's algorithm to compute transitive closure. Illustrate using the following directed graph.	10	L3	CO4

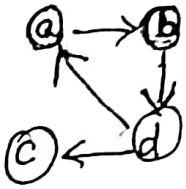


Fig.Q7(b)

OR

Q.8	a.	Define minimum spanning tree. Write Kruskal's algorithm to find minimum spanning tree. Illustrate with the following undirected graph.	10	L3	CO4

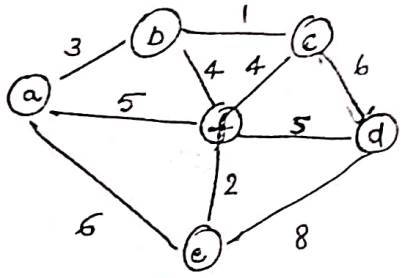


Fig.Q8(a)

	b.	Construct Huffman Tree and resulting code for the following:	10	L3	CO4

Character	A	B	C	D	-
Probability	0.4	0.1	0.2	0.15	0.15

(i) Encode the text : ABACABAD

(ii) Decode the text : 100010111001010

Module – 5

Q.9	a.	Explain n-Queen's problem with example using backtracking approach.	10	L2	CO5
	b.	Solve the following instance of the knapsack problem by the branch-and-bound algorithm. Construct state-space tree.	10	L3	CO5

Item	Weight	Value
1	4	$ 40
2	7	$ 42
3	5	$ 25
4	3	$ 12

The knapsack's capacity W is 10.

OR

Q.10	a.	Differentiate between Branch and Bound technique and Backtracking. Apply backtracking to solve the following instance of subset-sum problem S = {3, 5, 6, 7} and d = 15. Construct a state space tree.	10	L3	CO5
	b.	Explain greedy approximation algorithm to solve discrete knapsack problem.	10	L2	CO5

* * * * *

BCS401

Re: Madam, modifications of scheme and solutions

"Nirmala CR" <nirmala.cr@gmail.com> August 23, 2024 8:09 AM

To: boe@vtu.ac.in

Dear Sir,

I have checked the scheme and found appropriat. Kindly proceed with further process

Thanks and regards

NIRMALA C R

On Thu, 22 Aug 2024 at 12:57 PM, <boe@vtu.ac.in> wrote:

" APPROVED "

Registrar (Evaluation)
Visvesvaraya Technological University
BELAGAVI - 590018

2406BCS40170847

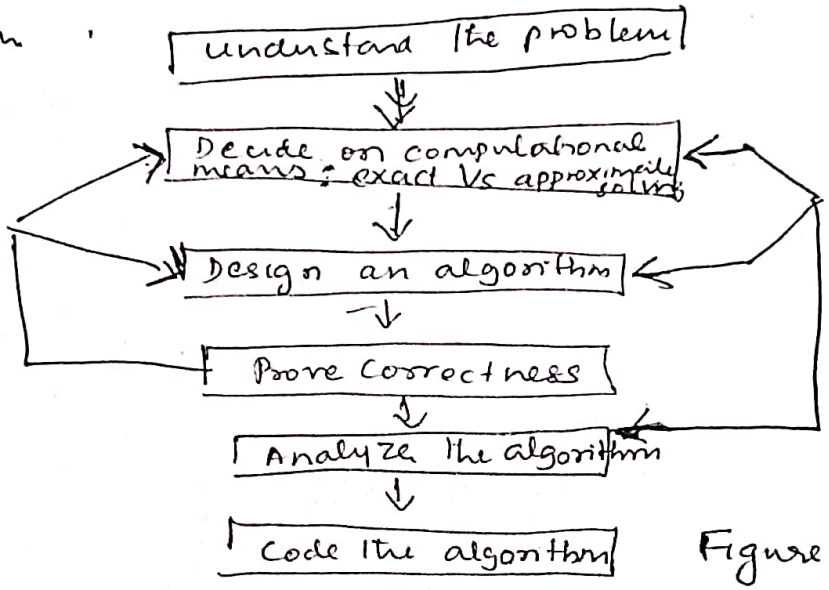
Visvesvaraya Technological University

Belagavi, Karnataka - 590 018.

Signature of Scrutinizer

Scheme & Solutions

Subject Title : Analysis & Design of Algorithms Subject Code : BCS401

Question Number	Solution	Marks Allocated
1.a.	Algorithm Definition →	1M
	Diagram : 	
	Figure →	3M
	explanation of each step →	6M
	total →	10M
1.b.	Sequential Search Algorithm →	2M
	Efficiencies - best case $C_{best}(n) = 1$ →	1M
	worst case $C_{worst}(n) = n$ →	2M
	average case efficiency ~~average case~~ : $C_{avg}(n) = [1 \cdot p/n + 2 \cdot p/n + \dots + n\, p/n] + n(1-p)$ $= p/n\,[1+2+\dots+n] + n(1-p)$	

APPROVED
Registrar (Evaluation)
Visvesvaraya Technological University
BELAGAVI - 590018

Subject Title : Analysis and design of Algorithms Subject Code : BCS401

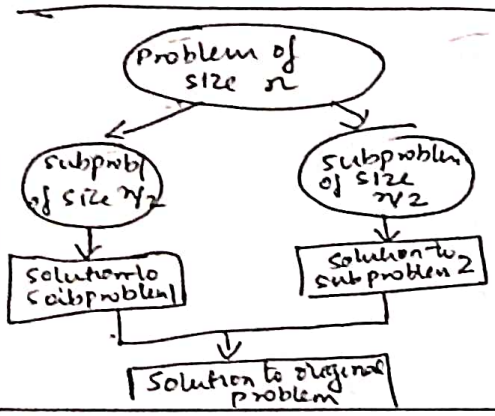
Question Number	Solution	Marks Allocated
	$= \frac{p}{n} \frac{n(n+1)}{2} + n(1-p) = \frac{p(n+1)}{2} + n(1-p)$	1M
	for sucessful search if $P = 1$ $\cancel{C_{ave}}$ $\cancel{\frac{(n+1)}{2}}$ $C_{avg}(n) = \frac{n+1}{2} \rightarrow$	1M
	for unsucessful search if $P = 0$ $C_{avg}(n) = n \longrightarrow$	1M
	total	10M
2.a.	Asymptotic notations: Big-oh (O) → definition, figure & example →	3M
	Big-omega (Ω) - definition, figure & example →	3M
	Big-theta (Θ) - definition, figure & example	4M
	total →	10M
2.b.	General plans listing: → 1. Decide on a parameter indicating an input size. 2. Identify the algorithm's basic operation 3. Check whether the no. of times the basic operation executed can vary on different i/ps of same size.	2M
	4. Set up a Recurrence relation with an initial condition. 5. Solve the recurrence, ascertain the order of growth	1M

Subject Title: Analysis & Design of Algorithm Subject Code: BCC401

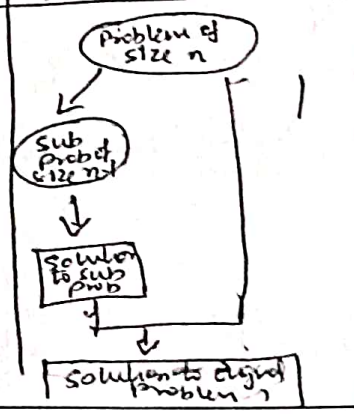
Question Number	Solution	Marks Allocated
	Alg. Recursive algorithm for computing factorial number Algorithm F(n) if n=0 return 1 else return F(n-1)* n	2M
	Recurrence relation : M(n) = M(n-1)+1 for n>0 M(0) = 0	2M
	Solving recurrence relation using back substitutions	3M
3a.	Strassen's Matrix Multiplication formula: $\begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} * \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix}$ $= \begin{bmatrix} m_1+m_4-m_5+m_7 & m_3+m_5 \\ m_2+m_4 & m_1+m_3-m_2+m_6 \end{bmatrix}$ where $m_1 = (a_{00}+a_{11}) * (b_{00}+b_{11})$ $m_2 = (a_{10}+a_{11}) * b_{00}$ $m_3 = a_{00} * (b_{01}-b_{11})$ $m_4 = a_{11} * (b_{10}-b_{00})$ $m_5 = (a_{00}+a_{01}) * b_{11}$ $m_6 = (a_{10}-a_{00}) * (b_{00}+b_{01})$ $m_7 = (a_{01}-a_{11}) * (b_{10}+b_{11})$	3M

Subject Title : Analysis & Design of Algorithms Subject Code : BCS401

Question Number	Solution	Marks Allocated
	explanation →	2M
	Recurrence relation of strassen's matrix multiplication $M(n) = 7M(n/2)$ for $n > 1$ $M(1) = 1$	2M
	derivation with solution $M(n) = 7^{\log_2 n} = n^{\log_2 7} \approx n^{2.807}$	3M
8.b.	Definition of Divide and Conquer →	1M
	quicksort algo :- →	4M
	Sorting the list of characters →	4M
	Quicksort best case efficiency → $\Theta(n \log n)$	1M
	Total →	10M
4.a.	Divide and Conquer technique : Decrease and conquer technique (by one)	2M

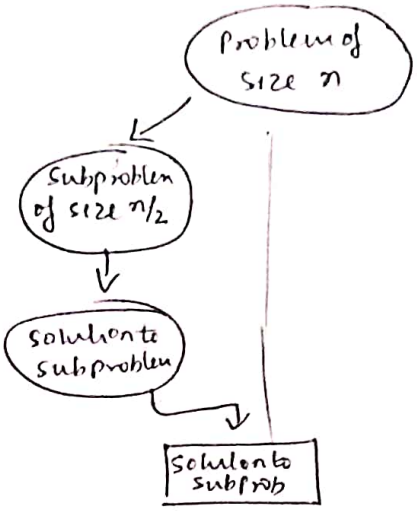


2M



1

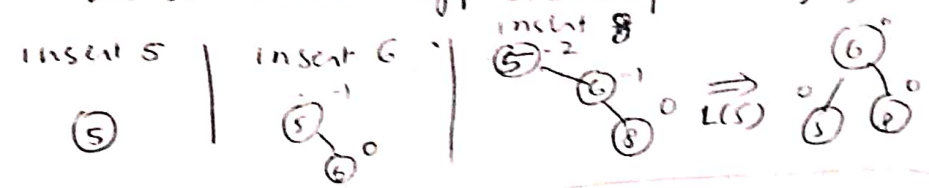
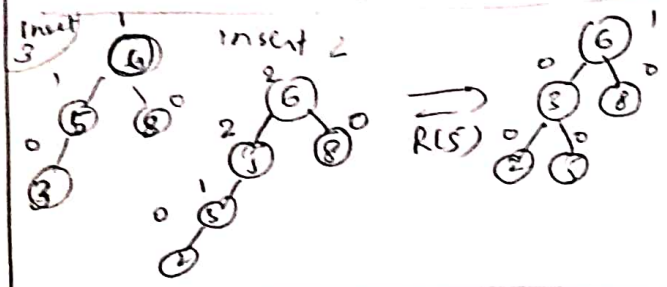
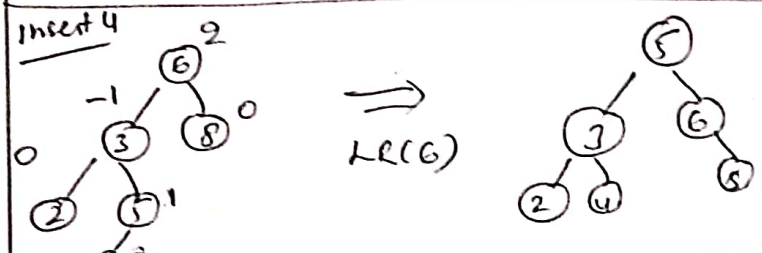
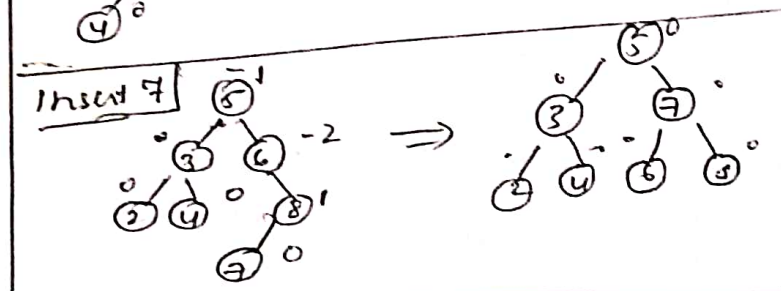
Subject Title : Analysis & Design of Algorithms Subject Code : BCS401

Question Number	Solution	Marks Allocated
	Decrease by half conquer technique: 	2M
	Insertion sort algorithm →	4M

	for $i \leftarrow 1$ to $n-1$ do
	$v \leftarrow A[i]$
	$j \leftarrow i-1$
	while $j \geq 0$ and $A[j] > v$ do
	$A[j+1] \leftarrow A[j]$
	$j \leftarrow j-1$
	$A[j+1] \leftarrow v$	

	$C_{worst}(n) = \Theta(n^2)$ $C_{best}(n) = \Theta(n)$ $C_{avg}(n) = \frac{n^2}{4} \in \Theta(n^2)$	3M
4.b.	Definition of topological sorting: →	1M
	List 2 approaches → 1) Source removal 2) DFS based	2M
	Illustration with example (2 approaches)	7M

Subject Title: Analysis & Design of Algorithms Subject Code: BCS401

Question Number	Solution	Marks Allocated
5.a.	AVL definition with example →	2M
	worst case efficiency → $\Theta(\log n)$ →	1M
		1M
		2M
		2M
		2M
	total	10
5.b.	Heap definition →	1M
	bottom up construction algorithm (next page) →	4M
	2, 9, 7, 6, 5, 8 → heap construction (in array rep) →	3M
	sorting the numbers (array rep) →	3M

Subject Title: Analysis & Design of Algorithms Subject Code: BCS401

Question Number	Solution	Marks Allocated
5.b cont.	**Bottom up heap construction algorithm:**	

```
for i ← ⌊n/2⌋ downto 1 do
    k ← i   v ← H[k]
    heap ← false
while not heap and 2*k ≤ n
    j ← 2*k
    if j < n
        if H[j] < H[j+1]  j ← j+1
    if v ≥ H[j]
        heap ← true
    else
        H[k] ← H[j] ; k ← j
H[k] ← v
```

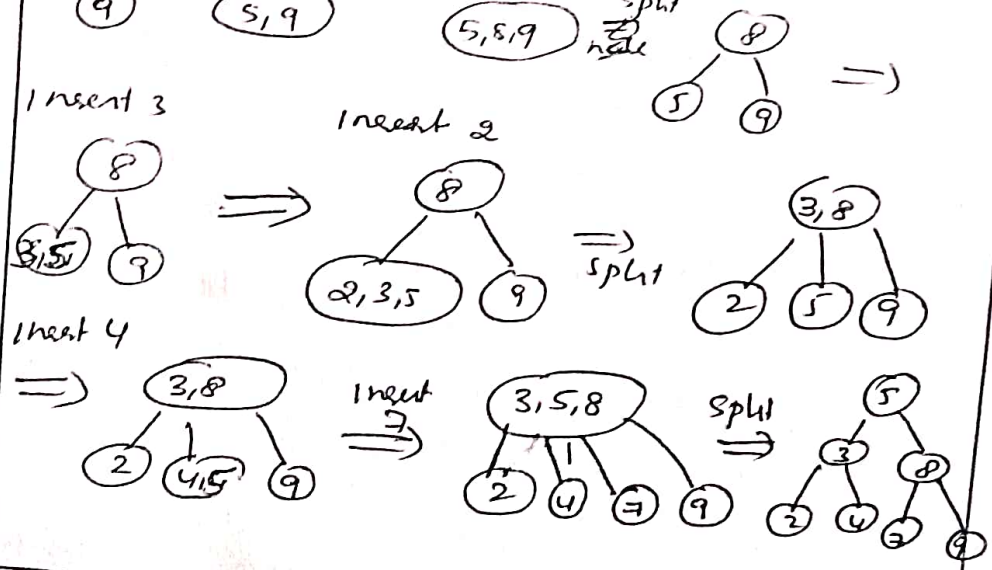
10

6.a

Definition 2-3 tree & ~~example~~ → 2M

Worst case efficiency → 1M

construction 2-3 tree 9, 5, 8 3, 2, 4, 7



2M

2M

3M

Subject Title : Analysing Design of Algorithms Subject Code : BCS4-01

Question Number	Solution	Marks Allocated
6. b.	Horspool's algorithm →	4M
	shift table for BARBER	2M

A	B	C	D	E	F	. . .	R	. . .	Z	-
4	2	6	6	1	6	6 . . .	3	6 . .	6	6

Actual search

T: JIM_SAW_ME_IN_A_BARBERSHOP

P: BARBER

BARBER

BARBER

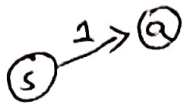
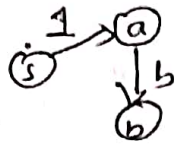
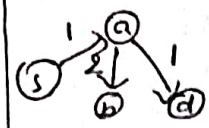
BARBER

BARBER

BARBER

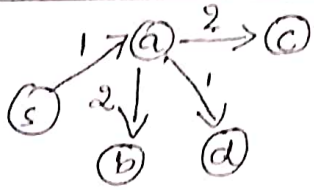
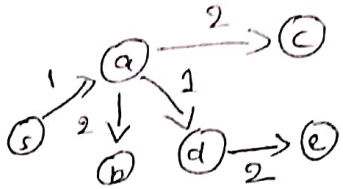
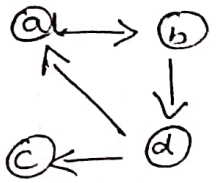
matched.

} 4M 8 4M

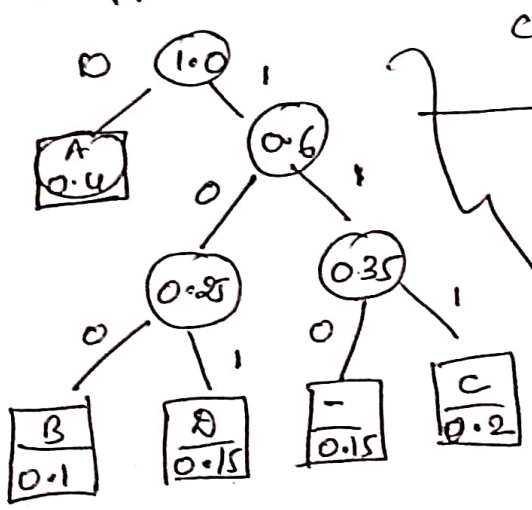
	Tree vertex	Remaining Vertices	Illustrate	
7. a.	s(-,0)	a(s,1) b(s,5) c(-,∞) d(-,∞) e(-,∞)		1M
	a(s,1)	b(a,3) c(a,3) d(a,2) e(-,∞)		2M
	~~b(a,3)~~	(b or c) any one can be selected to proceed as both are have same distance		
	b(a,3)	c(a,3) d(a,2) e(-,∞)		2M

Cont... next page.

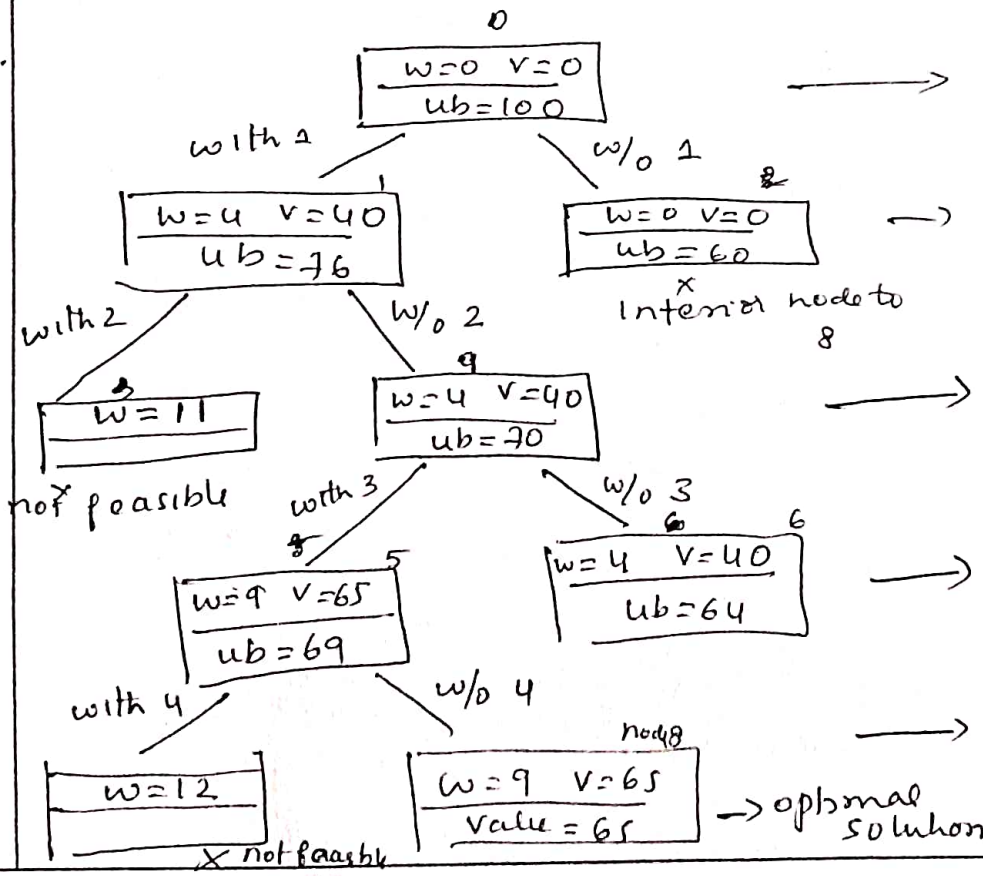
Subject Title : Analysis & Design of Algorithms Subject Code : BCS401

Question Number	Solution	Marks Allocated
	d(a,2) c(a,3) e(d,4) 	2M
	c(a,3) e(d,4) 	2M
	e(d,4)	1M
	Shortest path s → a is 1 s → a - b is 3 s → a → c is 3 | s → a - d is 2 s → a → d → e is 4	
7.b	Definition of transitive closure →	1M
	~~Transit~~ Warshall's algorithm →	3M
	Adjacency matrix  → $\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$	
	$R^{(0)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$ $R^{(1)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$	2M
	$R^{(2)} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$ $R^{(3)} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$	2M

Subject Title : Analysis & Design of Algorithms Subject Code : BCS401

Question Number	Solution	Marks Allocated
	$R^{(4)}$: $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$	2M
	$R^{(4)}$ is the transitive closure. ~~Efficiency is $O(n^3)$~~	
8.a.	Definition of MST →	1M
	algorithm →	3M
	Steps to derive MST 5 steps →	6M
	(bc, ef, ab, bf, df edges)	
8.b.	~~Fixed~~ stepwise presentation of Huffman tree	
		4M
	A → 0 B → 100 C → 111 D → 101 – → 110	2M

Subject Title : Analysis & Design of Algorithms Subject Code : BCS401

Question Number	Solution	Marks Allocated
	~~Avg # of bits per symbol = 2.2 bits~~	~~1M~~
	~~Fixed length encoding require = 3 bits~~	~~1M~~
	~~Compression ratio = $\left(\frac{3-2.2}{3}\right)\times 100$~~	~~2M~~
	~~$= \left(\frac{3-2.2}{3}\right)\times 100\ \%$~~	
	~~$= 26.67\%$~~	
	Encode text →	2M
	Decode text →	2M
9.a.	n-queen's problem explanation —	4M
	example with statespace tree —	6M
9.b.		
	→	2M
	→	3M
	→	2M
	→	2M
	→	1M

Subject Title : Analysis and design of Algorithms Subject Code : BCS401

Question Number	Solution	Marks Allocated
10.a	Difference → Any 2 x 1M	2M
	State space tree for subset sum problem } →	8M
10.b	discrete knapsack problem Algorithm	4M
	Explanation	6M

"APPROVED"

Registrar (Evaluation)
Visvesvaraya Technological University
BELAGAVI - 590018

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

SHOBHAVANA CAMPUS, MIJAR, MOODBIDRI – 574 225

DEPARTMENT OF COMPUTER SCIENCE & DESIGN

June-July 2024 Semester End Examinations (SEE) Overall Results IV Semester AY 2023-24

Result Analysis - Subject Wise - IV Semester 2023-24 (2022 Scheme)

SN	Subject	Sub Code	Total	O	A+	A	B	B+	C	F	AB	Percentage	>= B (70)%
1	Analysis & Design of Algorithms(Dr. Shivaprasad B J)	BCS401	62	3	13	26	0	18	0	1	0	96.77	67.74
2	Computer Graphics and Visualization(Dr. Shivaprasad B J)	BCG402	62	1	8	19	3	31	0	0	0	100.00	45.16
3	Database Management Systems(Prof. Jayanth Kumar A Rathod)	BCS403	62	1	7	10	11	23	5	5	0	91.94	29.03
4	Analysis & Design of Algorithms Lab(Dr. Shivaprasad B J)	BCSL404	62	21	6	14	2	15	0	3	0	93.55	66.13
5	Biology For Engineers(Dr. Keerthi Raj)	BBOC407	62	2	19	23	3	14	0	1	0	98.39	70.97
6	Universal human values course(Dr. Bharath)	BUHK408	62	1	13	32	1	13	0	0	0	96.77	74.19
7	National Service Scheme (NSS)(Dr. Suresh P S)	BNSK459	39	32	7	0	0	0	0	0	0	100.00	100.00
8	Physical Education (PE) (Mr. Dilip)	BPEK459	23	13	10	0	0	0	0	0	0	100.00	100.00
9	Discrite Mathematical Structures(Mrs. Radika / Mrs. Rashimi)	BCS405A	62	1	11	22	6	16	3	2	0	95.16	54.84
10	LaTex(Mr. Ramachendra H Y)	BCSL456D	62	50	9	1	1	1	0	0	0	100.00	96.77

Overall	Total	FCD	FC	SC	FL	AB	>= 9	>= 8	>= 7	>= 6	Percentage
IV Sem CSD	62	25	26	3	8	0	3	20	51	55	87.10

CLASS COORDINATOR

HOD

H.O.D
Dept. of Computer Science and Design
Alva's Institute of Engg. & Technology
Mijar, Moodubidire - 574 225

DEAN ACADEMICS

PRINCIPAL

PRINCIPAL
Alva's Institute of Engg. & Technology,
Mijar, MOODBIDRI - 574 225, D.K

Alva's Institute of Engineering and Technology

Department of Computer Science & Design

Students' Rating	.1	.2	.3	.4	.5	
	Q1	Q2	Q3	Q4	Q5	
No. of students	62					
	100.00	100.00	100.00	100.00	100.00	
Indirect Assessment Attainment Level	3	3	3	3	3	
Zero (No Ability)	0	0	0	0	0	
One (Some Ability)	2	3	3	2	2	
Two (Adequate Ability)	6	7	6	7	7	
Three (More than Adequate Ability)	4	6	8	8	8	
Four (High Ability)	50	46	45	45	45	

Attainment Level 1: 50% students rated more than or equal to 50% of maximum marks
Attainment Level 2: 60% students rated more than or equal to 50% of maximum marks
Attainment Level 3: 70% students rated more than or equal to 50% of maximum marks

Questionaries on Course End Survey	
.1	Understand the notaion of an algorithum, asymptotic notations and divide and conquer.
.2	Analyze the recursive and non-recursive algorithum and divide and conquer .
.3	Understand the algorithum design techniques using greedy method .
0.4	Understand the algorithum design techniques using dynamic programming.
0.5	explain the algorithum design techniques using backtracking, bracnch and bound and NP Compllete and NP Hard Problems.

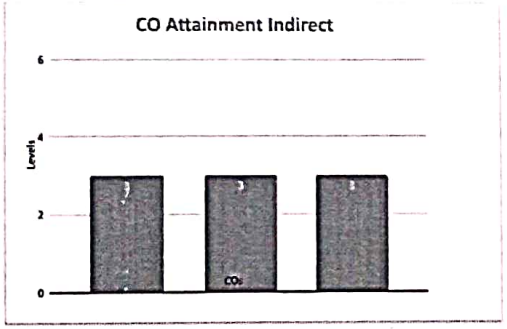
COURSE OUTCOMES (COs) ASSESSMENT MATRIX					
Alva's Institute of Engineering and Technology, Moodbidri					
Department of Computer Science & Design					
Academic Year:	2023-24			CO Attainment Indirect	CO Attainment
Course Name & Course Code:	Analysis & Design of Algorithum /				
Faculty Name:	Dr. Shivaprasad B J				
CO Attainment - Direct					
Cos	Formative Assessment	Summative Assessment	Total Attainment Direct		
.1	3	2	2.50	3	2.55
.2	3	2	2.50	3	2.55
.3	3	2	2.50	3	2.55
Average			2.50	3	2.55

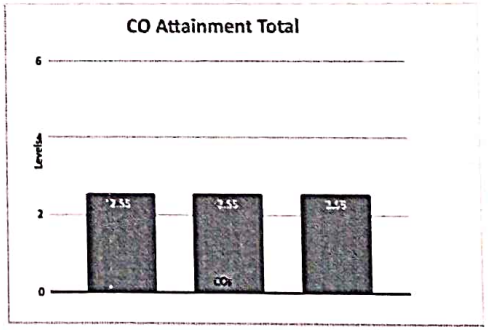
Attainment Level 1: 50% students rated more than or equal to 50% of maximum marks
Attainment Level 2. 60% students rated more than or equal to 50% of maximum marks
Attainment Level 3. 70% students rated more than or equal to 50% of maximum marks

Note:
Total Attainment Direct = (Weightage*Formative Assessment)+(Weightage*Summative Assessment) Weightage for Formative Assessment = 50%; Weightage for Summative Assessment = 50%

CO Attainment = (Weightage*Total Attainment Direct)+(Weightage*CO Attainment Indirect) Weightage for Total

Faculty Name & Signature:		HOD Signature:	





PROGRAMME OUTCOME & PROGRAMME SPECIFIC OUTCOME ASSESSMENT MATRIX

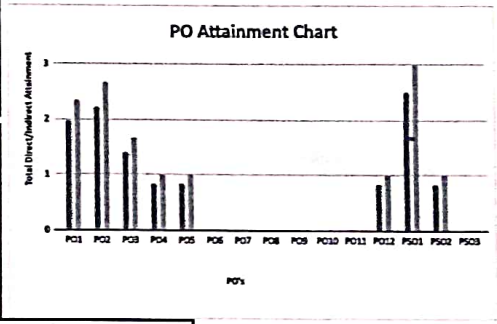
Alva's Institute of Engineering and Technology, Moodbidri

Department of Computer Science & Design

Academic Year:	2023-24
Course Name:	Analysis & Design of Algorithm
Course Code:	BCS401
Faculty Name:	Dr. Shyamvasai B.J

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
1	3	3	1	1	1							1	3	1	
2	2	3	1	1	1							1	3	1	
3	2	2	3	1	1							1	3	1	
4	2	2	3	1	1							1	3	1	
AVG	2.34	2.67	1.67	1	1							1	3	1	

Key Words (POs)	Apply Knowledge	Solve Problems	Design/ Development of Solutions	Conduct Investigations	Use Modern Tools	Engineer and Society	Environment and Sustainability	Professional Ethics	Individual and Team Work	Communicate Effectively	Project Management and Finance	Life-long Learning	PSO 1	PSO 2	PSO 3
Direct (D)	1.96	2.23	1.4	0.84	0.84	0	0	0	0	0	0	0.84	2.5	0.84	0
Indirect (I)	2.34	2.67	1.67	1	1	0	0	0	0	0	0	1	3	1	0
Attainment Level	2	2.28	1.43	0.86	0.86	0	0	0	0	0	0	0.86	2.55	0.86	0



PO Attainment Calculation Direct

COs	CO Attainment Grade	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
1	2.5	2.5	2.5	0.84	0.84	0.84	0	0	0	0	0	0	0.84	2.5	0.84	0
2	2.5	2.67	2.5	0.84	0.84	0.84	0	0	0	0	0	0	0.84	2.5	0.84	0
3	2.5	2.67	2.67	2.5	0.84	0.84	0	0	0	0	0	0	0.84	2.5	0.84	0
4	2.5	2.67	2.67	2.5	0.84	0.84	0	0	0	0	0	0	0.84	2.5	0.84	0
	Weighted Sum —>	5.84	6.67	4.18	2.52	2.52	0	0	0	0	0	0	2.52	7.5	2.52	0
	Max Weight —>	7	8	5	3	3	0	0	0	0	0	0	3	9	3	0
PO Attainment in percentage	—>	83.43	83.38	83.60	84.00	84.00	0.00	0.00	0.00	0.00	0.00	0.00	84.00	83.33	84.00	0.00
	PO Attained Grade	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
		1.96	2.23	1.4	0.84	0.84	0	0	0	0	0	0	0.84	2.5	0.84	0

PO Attainment Calculation Indirect

COs	CO Attainment Grade	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
1	3	3	3	1	1	1	0	0	0	0	0	0	1	3	1	0
2	3	2	3	1	1	1	0	0	0	0	0	0	1	3	1	0
3	3	2	2	3	1	1	0	0	0	0	0	0	1	3	1	0
4	3	2	2	3	1	1	0	0	0	0	0	0	1	3	1	0
	Weighted Sum —>	7	8	5	3	3	0	0	0	0	0	0	3	9	3	0
	Max Weight —>	7	8	5	3	3	0	0	0	0	0	0	3	9	3	0
PO Attainment in percentage	—>	100.00	100.00	100.00	100.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	100.00	100.00	0.00
	PO Attained Grade	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
		2.34	2.67	1.67	1	1	0	0	0	0	0	0	1	3	1	0

Faculty Name & Signature:	

HOD Signature:	

Academic Year:	2023-24
Course Name:	Analysis & Design of Algorithum
Course Code:	BCS401
Faculty Name:	Dr. Shivaprasad B J

Observations and Action Plan (Direct)

Course	Target	Attainment	Gap	Observation	Action Proposed to bridge the Gap
	2	2.50	0.00		

Outcomes on Actions for CAYm1 Observations/Suggestions

Sl. No.	Action Taken	Change Observed
1		

Faculty Name & Signature:	

HOD Signature:	

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

4th Semester (2023-24)

Bright Students List

Sr. No	USN	Name
01	4AL22CG004	ANVESH A SHETTY
02	4AL22CG020	GOUSIYA G
03	4AL22CG032	MOUNASHREE D P
04	4AL22CG034	NAVEEN HIREMATH
05	4AL22CG048	RAKSHITHA
06	4AL22CG049	RASHMI M
07	4AL22CG055	SNEHA DATTATRYA GOUDA
08	4AL22CG056	SNEHA S
09	4AL22CG057	SOWJANYA S
10	4AL22CG060	THANVI S SANIL

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

4th Semester (2023-24)

Weak Students List

Sr. No	USN	Name
01	4AL22CG003	ANANT R KULKARNI
02	4AL22CG013	DARSHAN S GOWDA
03	4AL22CG015	DHANUSHREE N
04	4AL22CG017	DRUSHYA HEGDE D
05	4AL22CG018	FAHAD UMAR FAROOQ
06	4AL22CG025	KARTIK MYAGERI
07	4AL22CG031	MOHAMMED IRFAN
08	4AL22CG033	NAGARAJ DADDI
09	4AL22CG035	NIKHITH NAIK
10	4AL22CG045	PREMKUMAR INGALE