Nirma University
Institute of Technology

Computer Science and Engineering department

Course Policy

B.Tech. in Computer Science and Engineering

Semester: III, **Academic Year:** 2022-23, **Term:** ODD

Course Code & Name	:	2CS301 Data Structures and Algorithms
Credit Details	:	Lecture-3, Tutorial-0, Practicals-2 **Credits-4**
Course Co-ordinator	:	Prof. Malaram Kumhar
Contact No. & Email	:	07971652570, malaram.kumhar@nirmauni.ac.in
Office	:	NF-7(New Block)
Visiting Hours **Course Blog**	: :	9:00 AM to 4:00 PM https://2cs301mkk.wordpress.com/
Course Faculty	:	**1. Prof. Malaram Kumhar** **Visiting Hours**: Saturday(1st and 3rd): 02:00 p.m. to 04:00 pm
2. Prof. Jitali patel **Email:** jitali.patel@nirmauni.ac.in Office: NF-3(New Block) **Visiting Hours:** Saturday(1st and 3rd): 02:00 p.m. to 04:00 pm	**3. Dr. Jitendra Bhatia** **Email:** jitendrabbhatia@gmail.com Office: 5th floor(New Block) **Visiting Hours:** Saturday(1st and 3rd): 02:00 p.m. to 04:00 pm	
3. Dr. Sanjay Patel **Email:** sanjay.patel_cse@nirmauni.ac.in Office: 5th floor(New Block) **Visiting Hours:**		

Saturday(1st and 3rd): 02:00 p.m. to 04:00 pm	
Course Blog:	https://2cs301mkk.wordpress.com/

1. Introduction to Course:

- **Importance of course:**

  ✓ Data Structure is process through which we can collect and organize data in best way as well as perform operation on that in most effective way. If we have good understanding of data structures then we are specialized in organizing and storing data. This course covers the modern theory of algorithms, focusing on the themes of efficient algorithms and intractable problems. The course goal is to provide a solid background in algorithms for computer science students, in preparation either for a job in industry or for more advanced courses at the graduate level.

- **Course objective:**

  ✓ To develop students' knowledge in data structures and the associated algorithms. To introduce the concepts and techniques of structuring and operating on Abstract Data Types in problem solving.

  ✓ To discuss common sorting, searching and graph algorithms, and to study the complexity and comparisons among these various techniques.

Pre-requisite: Preliminary knowledge basic programming languages like C and C++ will help students to understand basic concepts of this subject.

2. Course Learning Outcomes(CLO):

After successful completion of this course, student will be able to

CLO 1 : Analyse various data structures and their applicability

CLO 2 : Comprehend and Implement various techniques for searching and sorting

CLO 3 : Identify the appropriate data structure to design efficient algorithm for the given application

3. Syllabus:

Syllabus:	**Teaching Hours:**
Unit I **Introduction to Data Structures:** Basic Terminology, Elementary Data Structure Organization, Classification of Data Structures: Primitive and Non-primitive, Linear and Non-linear,	**06**

Operations on Data structures, Asymptotic notations, Notion of recursive algorithms.	
Unit II **Linear Data Structures:** Introduction, variations, operations and applications of array, queue, stack and linked list	**12**
Unit III **Non Linear Data Structures:** Concepts and types of trees, tree traversal algorithms, search trees, Priority queue implementation and applications, Representations of Graphs, Graph algorithms i.e. traversals, minimum spanning tree, shortest path, Traveling Salesman Problems	**12**
Unit IV **Indexing structure:** Concepts and implementations of B-Tree, B+ tree, Hashing, Dictionary	**07**
Unit V **Searching and Sorting Algorithms:** Linear search, Binary search, internal and external sorting algorithms, sorting without comparison.	**08**
Total:	**45**

Suggested Readings^:

1. Jean-Paul Tremblay and Paul G. Sorenson, An Introduction to Data Structures with Applications, Tata McGraw Hill
2. Tanenbaum, Data Structures using C & C++, PHI
3. Robert L. Kruse, Data Structures and Program Design in C, PHI
4. Mary E.S. Loomis, Data Management and file processing, PHI

L = Lecture, T = Tutorial, P = Practical, C = Credit

^ this is not an exhaustive list

Self-study:

Non Linear Data Structures: Priority queue implementation using heap, Traveling Salesman Problems
Indexing structure: B+ tree and Dictionary

Reference and Text books:

1. Jean-Paul Tremblay and Paul G. Sorenson, An Introduction to Data Structures with Applications, Tata McGraw Hill
2. Tanenbaum, Data Structures using C & C++, PHI
3. Robert L. Kruse, Data Structures and Program Design in C, PHI
4. Mary E.S. Loomis, Data Management and file processing, PHI

4. Laboratory details: (List of Experiments, Schedule, assessment policy)

Prac. No.	Topic	Hour (s)	Mapped CLO's
1.	a) An organization has to maintain it's employee's details. There is need of accessing details of employees frequently. Taking this information into consideration, use an appropriate data structure to implement system for providing functionality of adding details of new employee, removing employee's detail from the system and listing all employees' details. b) Design anagram game using array. Allow a user to enter N words and store it in an array. Generate a random number between 0 to N-1. Based on the random number generated display the word stored at that index of an array and allow user to enter its anagram. Check whether the word entered by the user is an anagram of displayed number or not and display an appropriate message. [Given a word A and word B. B is said to be an anagram of A if and only if the characters present in B is same as characters present in A, irrespective of their sequence. For ex: "LISTEN" == "SILENT"]	04	CLO1, CLO3
2.	a) Write a program to reverse the elements in the stack using recursion. b) Write a program to convert fully parenthesized infix expression into postfix expression. Show all the intermediate results in the table format.	04	CLO1, CLO3
3.	a) Write a program to simulate printer spooler application. Assume maximum 5 users are using this printer. Use appropriate data structure to implement the system. b) Write a program to implement priority queue using 2D array.	04	CLO1, CLO3
4.	Write a program to implement doubly linked list where each node consists of integer values. The program should support following functionalities. i. Create a doubly linked list ii. Delete a node if it is found, otherwise display an appropriate message	02	CLO1

	iii. Search a given integer value in the list iv. Display the doubly linked list		
5.	a) Write a program to simulate music player application using suitable data structure. There is no estimation about number of music files to be managed by the music player. Your program should support all the basic music player operations to play and manage the playlist. b) Write a program to perform addition of two polynomial equations using appropriate data structure.	**04**	**CLO1, CLO3**
6.	Write a program to construct a binary tree from the given postorder and inorder traversal sequence.	**02**	**CLO2**
7.	Write a program to implement phone book dictionary using Binary Search Tree which provides following operations: (a) Add new entry in phone book, (b) Remove entry from phone book, (c) Search phone number (d) List all entries in ascending order of name and (e) List all entries in descending order of name.	**04**	**CLO2**
8.	Write a program to traverse connected undirected graph using Depth First Search(DFS) traversing technique and give the traversing sequence.	**02**	**CLO3**
9.	a) Write a program to implement Selection sort for sorting a given set of integers in ascending order. b) Write a program to implement Quick sort algorithm for sorting a given set of integers in ascending order.	**04**	**CLO3**
10.	Implement Binary search technique, which takes a list of unique values sorted in descending order and a value to search for, and returns either the index of the value or None, if the value isn't in the list.	**02**	**CLO1, CLO3**
	Total:	**32**	

5. Assessment Policy

5.1 Component wise Continuous Evaluation (CE), Laboratory and Project Work (LPW) & Semester End Examination (SEE) weightage

Assessment scheme	***CE***				***LPW***		***SEE***
Component weightage	*0.4*				*0.2*		*0.4*
	Quiz-1 *15%*	*Quiz-2* *20%*	*Sessional Examination* *35%*	*Assignments* *30%*	*Continuous Evaluation* *75%*	*Viva Voce* *25%*	-

5.2 Assessment Policy for Continuous Evaluation (CE)

Assessment of Continuous Evaluation comprises of three components.

1. Quiz-1 will be conducted as per academic calendar. It will be conducted online/ offline and covers the 15% of total CE component weightage.
2. Quiz-2 will be conducted as per academic calendar. It will be conducted online/ offline and covers the 20% of total CE component weightage.
3. Sessional Exam will be conducted as per academic calendar. It will be conducted online/offline and covers the 35% of total CE component weightage
4. Assignments covers the 30% of total CE component weightage

5.3 Assessment Policy for Laboratory and Project Work (LPW)

Assessment of Laboratory and Project Work comprises of two components.

1. Continuous assessment for laboratory experiments will be conducted. There will be 10 experiments, each carrying weightage of 10 marks. At the end of the course total marks obtained out of 100 will be converted according to weightage assigned. Assessment of Experiment will be carried out based on overall logic design and timely submission of practical.
2. A Viva voce examination for LPW component will be conducted as per academic calendar. It will carry a weightage of 25 marks.

5.4 Assessment Policy for Semester End Examination (SEE)

A written examination of 3 hours duration will be conducted for the course as per academic calendar. It will carry 100 marks and marks obtained out of 100 will be converted as per weightage assigned.

6. Lesson Plan

Sr. No.	Topic	Hour (s)	CLO	Applications
1	**Introduction to Data Structures** ➢ Basic Terminology, Elementary Data Structure Organization ➢ Classification of Data Structures: Primitive and Non-primitive, Linear and Non-linear ➢ Operations on Data structures ➢ Asymptotic notations, Notion of recursive algorithms.	[4] 1 1 1 1	1	Data structure used in various technical and real life application. Array is used to implement vectors, matrices, hash table, stack and queue.
2	**Linear Data Structures** **Array** ➢ Array operations: Insertion and deletion ➢ Address calculation **Stack** ➢ Stack operations: PUSH, POP, PEEP, CHANGE ➢ Applications of stack -Recursion -Infix, Postfix & Prefix notations, Infix to Postfix conversion - Evaluation of postfix expression -Tower of Hanoi using stack **Queue** ➢ Simple: Algorithms & Implementation of simple Queue an array	[17] 1 1 1 1 1 1 1 1 1	1,3	**Stack:** Stack is used in undo\redo operation in word processors, Expression evaluation and syntax parsing **Queue:** Transport and operations research where various entities are stored and held to be processed. **Linked List:** In line editor, implementation of sparse matrix and also to implement stack and queue

	➢ Circular: Algorithms & Implementation of Circular Queue an array ➢ Doubly ended queue: Algorithms & Implementation of Circular Queue an array ➢ Priority Queue: Algorithms & Implementation of using an array **Linked Lists** ➢ Singly linked list: Insertion and Deletion algorithms ➢ Circular linked lists: Insertion and Deletion algorithms ➢ Doubly linked list: Insertion and Deletion algorithms ➢ Applications of linked list	1 1 1 1 2 2		
3	**Trees & Graphs** ➢ Definitions of Trees & graphs ➢ Creation of Simple Binary tree and tree traversing ➢ Conversion of General Tree in to Binary tree ➢ Threaded representation of tree ➢ Insertion, Deletion and Traversal of binary search tree ➢ Priority queue implementation using heap ➢ Graph structure & Representation of graph ➢ Minimum Spanning Tree ➢ Traversing of Graph (DFS &BFS) ➢ Traveling Salesman Problems ➢ Applications of graph	[11] 1 2 1 1 2 1 1 1 1	1,3	**Trees:** Trees are used in Parsers, Filesystem, to represent network topology as a tree **Graphs**: Graph are used in Connections/relations in social networking sites, Routing ,networks of communication

4	**Indexing Structure** ➤ Concepts and implementations of B-Tree ➤ B-Trees and B+ tree ➤ Dictionary	[3] 1 2	1,3	used in database management systems
5	**Sorting algorithms** ➤ Insertion sort & Selection Sort ➤ Shell sort & Bubble sort ➤ Quick sort ➤ Heap sort ➤ Merge Sort ➤ Radix sort and Tree sort	[6] 1 1 1 1 1 1	1,2	Sorting techniques are basic in any computing which makes many problem easy such as searching, frequency distribution, selection, closest pairs
6	**Searching techniques** ➤ Linear Search and Binary Search ➤ Height Balanced Trees ➤ Hashing techniques	[4] 1 2 1	1,2	Searching techniques are used to search the required data in stack, queue, linked list and also finding a node in a tree.
	Total:	**45**		

Note: Topics marked as red are for self-study.

7. Mapping of Session Learning Outcomes (SLO) with Course Learning Outcomes (CLO)

Session No.	**Session Learning Outcomes:** **After successful completion of the session, student will be able to**	**CLO**
1.	Understand importance, scope and policy of the course Understand the need of various data structures.	-
2.	Prerequisite : Pointer, Structure and function	-
3.	Prerequisite: Asymptotic notation: Time and Space Complexity.	-
4.	Understand the fundamentals of array data storage in 1-D, 2-D and N-D.	1
5.	Ability to calculate physical address for the array data storage in Row Major and Column major formats:	1
6.	Understand stack data structure and its basic operations. Identify applicability of stack data structure	1,3

7.	Understand recursive process and applications where recursion is used.	3
8.	Understand the process of conversion from infix to postfix. Apply the knowledge and convert expressions in suggested formats	3
9.	Understand the process of conversion from infix to prefix. Apply the knowledge and convert expressions in suggested formats	3
10.	Understand the process of evaluating a given expression. Ability to solve expressions in given formats	3
11.	Ability to trace recursive applications such as tower of Hanoi.	3
12.	Understand queue data structure and its basic operations. Identify applicability of queue data structure	1
13.	Understand circular queue data structure and its basic operations.	1
14.	Understand double ended simple queue data structure and its basic operations.	1
15.	Understand Priority queue data structure and its basic operations.	1,3
16.	Understand dynamic data structures and their applicability. Understand singly linked list and its basic operations and their algorithms.	1
17.	Understand doubly linked list and its basic operations	1
18.	Ability to write algorithms to basic operations of doubly linked list.	1
19.	Understand circular doubly linked list and its basic operations	1
20.	Ability to write algorithms to basic operations of circular doubly linked list.	1
21.	*A*bility to Identify areas where linked list is applicable.	3
22.	Understand tree construction process. Ability to implement the basic structure.	1
23.	Understand height balanced trees and their data structure with applications	1,3
24.	Understand hashing technique and their real world application	1,3
25.	Ability to resolve hash collisions and calculate solutions to given hashing problem.	1,3
26.	Understand tree and graph structure and their real world implementations and applications.	1,3
27.	Ability to create Simple Binary tree	1
28.	Ability to convert General Tree in to Binary tree	1

29.	Understand binary search tree its construction and application	1,3
30.	Understand the basic binary search tree operations.	1
31.	Ability to implement basic binary search tree operations and traversals.	2
32.	Understand Graph structure & Representation of graph	1
33.	Understand DFS traversal technique Ability to solve problems of traversals	1,3
34.	Understand BFS traversal technique Ability to solve problems of traversals. Travelling Salesman problem.	1,3
35.	Understand the Implementation and Traders of Sequential Access, B-Trees,	3
36.	Minimum Spanning Tree: Prim's and Kruskal's Algorithm	3
37.	Understand B-tree and solve problem related to B-trees	1,2
38.	Understand the sorting techniques and trace insertion and selection sort.	1,2
39.	Understand the bubble and shell sorting techniques Ability to trace bubble and shell sorting techniques.	2
40.	Understand the quick sorting techniques Ability to trace quick sorting techniques.	2
41.	Understand the heap sorting techniques Ability to trace heap sorting techniques.	2
42.	Understand the merge sorting techniques Ability to trace merge sorting techniques.	2
43.	Understand the radix sorting techniques Ability to trace radix sorting techniques.	2
44.	Understand the radix sorting techniques Ability to trace radix sorting techniques.	2
45.	Understand the searching techniques. Ability to trace linear and binary searching techniques.	2

8. Teaching-learning methodology

1. Lectures: Primarily Chalk and Black board, Power Point Presentations (PPTs) and Demonstration of concepts through programs will be used to conduct the course. However, where required, Video Lectures, Animations etc. will be used to enhance the teaching-learning process.
2. Laboratory: Explanation of Experiment to be performed along with co-relation with theory will be given. At the end of each session

assessment will be carried out based on the logic developed and program written and also timely submission of practical.

9. **Active learning techniques**

Active learning is a method of learning in which students are actively or experientially involved in the learning process. Following active learning techniques will be adopted for the course.

1. Recall, Summarize, Question, Connect, and Comment: At the beginning of class, students are asked to recall and list the most important points from the previous class. They then summarize these points in sentences. Next students write one questions from the previous material that they wanted answered. Fourth, they are instructed to make one connection between what they learned in the previous class and any of the classes before that. Finally, they are asked to comment on how confident they felt.
2. Flipped Class-room: In the flipped classroom, instructors assign video lectures or reading material as homework, and use class time for active learning exercises and direct engagement with students.

10. **Course Material**

Following course material is uploaded on the course website:

https://sites.google.com/a/nirmauni.ac.in/2cs301-data-structures-and-algorithms/

Course Policy

- Lecture Notes
- Books / Reference Books / NPTEL video lectures
- Assignments, Lab Manuals
- Question bank
- Web-links, Blogs, Video Lectures, Journals
- Softwares
- Advanced topics

11. **Course Learning Outcome Attainment**

Following means will be used to assess attainment of course learning outcomes.

- Use of formal evaluation components of continuous evaluation, laboratory work, semester end examination
- Informal feedback during course conduction

12. Academic Integrity Statement

Students are expected to carry out assigned work under Continuous Evaluation (CE) component and LPW component independently. Copying in any form is not acceptable and will invite strict disciplinary action. Evaluation of corresponding component will be affected proportionately in such cases. Turnitin software will be used to check plagiarism wherever applicable. Academic integrity is expected from students in all components of course assessment.