Course Title: Data Structure and Algorithms (3 Cr.)
Course Code: CACS201
Year/Semester: II/III
Class Load: 6Hrs. / Week (Theory: 3 Hrs. Practical: 3 Hrs.)

Course Description

This course includes fundamental concept of data structures such as stack, queue, list, linked list, trees and graph; application of these data structures along with several algorithms.

Course Objectives

The general objective of this course is to provide fundamental concept of data structures, different algorithms and their implementation.

Course Contents

Specific Objectives	Content	Hour
• Define Data Structure and Algorithms. • Classify and explain the various Data Structures. • Explain Data Structure operations. • Explain an Abstract Data Type. • List out the Importance of data structures. • Discuss on time and space complexity of algorithm. • Introduce asymptotic notations.	**Unit 1: Introduction to data Structure** Definition, Abstract Data Type, Importance of Data Structure	2 Hrs.
• Define Stack Data Structure. • Explain stack as an ADT. • Write an algorithms for Stack Operations. • List out the applications of Stack. • Evaluate Infix, Postfix and Prefix expressions. • Implement an algorithms to convert infix expression to postfix and prefix expressions.	**Unit 2: The Stack** Introduction, Stack as an ADT, POP and PUSH Operations, Stack Applications, Evaluation of Infix, Postfix, and Prefix Expressions, Conversion of Expression.	3 Hrs.
• Define Queue. Explain Queue as an Abstract Data Type. • Describe the primitive operations in Queue.	**Unit 3: Queue**	3 hrs.

• Classify the Queue. • Implement the Linear Queue. • Implement the Circular Queue. • List out applications of different types of Queue. • Explain the Priority Queue with its operations and Applications.	Introduction, Queue as an ADT, Primitive operations in Queue, Linear and Circular Queue and their applications, Enqueue and Dequeue, Priority Queue.	
• Define List with its applications. • Differentiate between Static and Dynamic List Structure. • Implement the List using Array. • Implement queue as a List.	**Unit 4: List** Introduction, Static and Dynamic List Structure, Array Implementation of Lists, Queues as a List	2 Hrs.
• Define Linked List. Explain Linked List as an ADT. • Explain the primitive operations of Linked List. • Classify the types of Linked List. • Insert an item in linked list at start, end and specific location of linked list. • Delete an item from linked list at start, end and specific location of linked list. • Implement Singly, Circular, Doubly and Circular Doubly Linked List. • Implement Linked Stack. • Implement Linked Queue. • List out the applications and advantages of Linked List.	**Unit 5: Linked List** Introduction, Linked List as an ADT, Dynamic implementation, Insertion & Deletion of Node to and from a List, Insertion and Deletion after and before Nodes, Linked Stakes and Queues, Doubly Linked Lists and Its Advantages	5 Hrs.
• Define Recursion. Explain the Principle of Recursion. • Differentiate between Recursion and Iterations. • List out advantages and disadvantages of Recursion. • Use Recursion to solve the different problems (calculate factorial, reversing an integer, checking prime number, TOH, Fibonacci Series etc.). • Implement recursion to find form search tree.	**Unit 6: Recursion** Introduction, Principle of Recursion, Recursion vs. Iteration, Recursion Examples: TOH, Fibonacci Series, Application of Recursion, Search Tree	4 Hrs.
• Define tree and tree terminologies. • Explain the basic operations in binary tree.	**Unit 7: Trees**	5 Hr.

• Insert and delete node in binary tree. • Traverse binary tree using pre-order, post-order, and in-order traversal methods. • Explain the applications of Binary Tree. • Construct AVL tree using given set of data. • Implement Huffman Algorithm. • Implement Game Tree with its applications. • Construct B-Tree.	Introduction, Basic Operations in Binary Tree, Tree Search and Insertion/Deletion, Binary Tree Traversals (pre-order, post-order and in-order), Tree Height, Level and Depth, Balanced Tree: AVL Balanced Trees, Balancing Algorithm, The Huffman Algorithm, Game Tree, B-Tree.	
• Define Sorting. Differentiate between Internal & External Sorting. • Write a program to implement: Insertion Sort, Selection Sort, Bubble Sort, Quick Sort, Merge Sort, Heap Sort. • Demonstrate the sorting using Exchange Sort, Radix Sort, Shell Sort, and Binary Sort. • Compare Quick sort and Heap Sort. • Construct a Heap using given set of data. • Explain Heap as a Priority Queue. • Analyze Efficiency (best, average & worst case) of each Sorting Algorithms.	**Unit 8: Sorting** Introduction, Internal & External Sort, Insertion and Selection Sort, Exchange Sort, Bubble and Quick Sort, Merge and Radix Sort, Shell Sort, Binary Sort, Heap Sort as Priority Queue, Efficiency of Sorting, Big 'O' Notation	6 Hrs.
• Introduce Searching. Explain dictionary as ADT. • List out applications of Searching. • Implement Sequential search. Analyze an efficiency of sequential search. • Implement Binary search. Analyze an efficiency of binary search with compare to sequential search. • Implement binary search tree as searching mechanism. • Introduce Hashing. Explain and implement Hash Functions and Hash Tables. • Discuss efficiency of Rehashing method. • Identify and implement different types of collision resolution techniques.	**Unit 9: Searching** Introduction to Search Technique; essential of search, Sequential Search, Binary Search, Tree Search, General Search Tree, Hashing: Hash functions and Hash tables, Collision resolution techniques, Efficiency comparison of different search technique	5 Hrs.

• Introduce graph. Explain graph as an ADT. • Explain the Applications of graph theory. • Explain the transitive closure property of graph. • Implement the Warshall's algorithm for both directed and undirected graph. • Explain graph traversal methods with their efficiency. • Implement shortest path algorithm to solve graph problems. • Implement Dijkstra's algorithm to solve single source shortest path graph problem. • Explain the minimum spanning tree and forest. • Use Kruskal's Algorithm to solve MST problem. • Use Prim's Algorithm to solve MST problems. • Implement the Round Robin Algorithm to solve graph problem. • Explain the greedy algorithm to traverse graph with its efficiency	**Unit 10: Graph** Introduction, Graph as an ADT, Transitive Closure, Warshall's Algorithm, Types of Graph, Graph Traversal and Spanning Forests, Kruskal's and Round Robin Algorithms, Shortest Path Algorithm, Greedy Algorithm, Dijkstrs's Algorithm	5 Hrs.
• Explain the behavior of Deterministic and non-deterministic algorithms with their applications, performance and examples. • Demonstrate the Divide and Conquer algorithm with its performance. • Differentiate between Series and Parallel Algorithms with their applications. • Demonstrate the Heuristic and Approximate algorithm with their performance.	**Unit 11: Algorithms** Deterministic and Non-deterministic Algorithm, Divide and Conquer Algorithm, Series and Parallel Algorithm, Heuristic and Approximate Algorithms	5 Hrs.

Laboratory Works

Laboratory Topics	Laboratory Activities
There shall be 10 lab exercises based on C or Java 1. Implementations of different operations related to Stack.	1. Write a program to show the Stack operations. 2. Write a program to implement the linear queue operations.

2. Implementation of different operations related to linear and circular queue. 3. Solution of TOH and Fibonacci Series using Recursion. 4. Implementations of different operations related to linked list: Singly and doubly Linked List. 5. Implementation of Trees: AVL trees, Balancing AVL. 6. Implementation of merge sort. 7. Implementation of different searching technique: sequential, Tree and Binary. 8. Implementation of Graphs: Graph traversal 9. Implementation of Hashing 10. Implementation of Heap	3. Write a program to implement circular queue. 4. Write a program to calculate factorial number using recursion. 5. Write a program to check prime number using recursion. 6. Write a program to reverse integer number using recursion. 7. Write a program to print Fibonacci series up to given number using recursion. 8. Write a program to solve TOH problem using recursion. 9. Write a program to count the nodes in Linked List. 10. Write a program to insert and delete an item at the beginning of Singly Linked list. 11. Write a program to insert and delete an item at the end of Singly Linked list. 12. Write a program to insert and delete an item at specified location of Singly Linked list. 13. Write a program to insert and delete an item at the beginning of Doubly Linked list. 14. Write a program to insert and delete an item at the end of Doubly Linked list. 15. Write a program to insert and delete an item at specified location of Doubly Linked list. 16. Insert and delete node form circular linked list. 17. Write a program to create binary search tree using given set of nodes. 18. Write a program to implement binary tree traversal methods (pre-order, in-order and post-order). 19. Write a program to construct AVL tree using given set of data. 20. Write a program to sort an array using Bubble Sort. 21. Write a program to sort an array using Insertion Sort. 22. Write a program to sort an array using Selection Sort. 23. Write a program to sort an array using Merge Sort.

	24. Write a program to sort an array using Quick Sort. 25. Write a program to construct a heap using given set of data. 26. Write a program to sort an array using Heap Sort. 27. Write a program to search an item from an array using sequential search. 28. Write a program to search an item from an array using binary search. 29. Write a program to represent the graph. 30. Write a program to traverse (BFS, DFS) in a graph. 31. Write a program to implement Dijkstra's Algorithm for finding single source shortest path problem. 32. Write a program to implement open addressing hashing approach. **Note: Teacher's can add extra practical activities to make clearer the content and to fulfill the requirement of course.**

Teaching Methods

The general teaching methods includes class lectures, group discussions, case studies, guest lectures, research work, project work, assignments(theoretical and practical), and exams, depending upon the nature of the topics. The teaching faculty will determine the choice of teaching pedagogy as per the need of the topics.

Evaluation

Evaluation Scheme				
Internal Assessment		External Assessment		Total
Theory	Practical	Theory	Practical	100
20	20 (3 Hrs.)	60 (3 Hrs.)	-	

Internal/Practical Assessment Format [FM = 40]

Internal Assessment Format [FM = 20] – Subject Teacher

Term Examination		Assignment	Attendance	Total
Mid - Term	Pre - Final			
5	5	5	5	20

Practical Assessment Format [FM = 20] – External Examiner will be assigned by Dean Office, FOHSS.

Practical	Viva	Lab Reports	Total
10	5	5	20

Note: Assignment may be subject specific case study, seminar paper preparation, report writing, project work, research work, presentation, problem solving etc.

Final Examination Questions Format [FM = 60, PM = 24, Time = 3 Hrs.]

SN	Question Type	Number of Questions Given	Marks per Question	Total Marks
1	Group – 'A' Objective Type Questions(Multiple Choice Questions)	10	1	10 x 1 = 10
2	Group – 'B' Short Questions (Attempt any SIX questions)	7	5	6 x 5 = 30
3	Group – 'C' Long Questions (Attempt any TWO questions)	3	10	2 x 10 = 20

- Student must pass 'Internal Assessment', 'Practical Assessment' and 'Final Examination' separately.
- Student must attend each and every activity of 'Internal Assessment' otherwise he/she will be declared as 'Not Qualified' for final Examination.

Text Books

1. Y. Langsam, M.J. Augenstein and A. M, Tanenbaum, "Data Structures using C and C++", PHI

Reference Books

1 G. W. Rowe, "Introduction to Data Structures and Algorithms with C and C++", PHI.
2 Robert Lafore, "Data Structures and Algorithms in Java", Sams Publishing.
3 G. S. Baluja, "Data Structures through C", Dhanpat Rai & Co.

Internal Assessment marks Submission format

Campus Name:									
Subject Name: Data Structure and Algorithms						**Subject Code: CACS201**			
SN	**TU Registration No.**	**Name**	**Symbol No.**	**Mid – Term [5]**	**Pre – Final [5]**	**Assignment [5]**	**Attendance [5]**	**Total [20]**	**Remarks**

Name of Subject Teacher:

Signature:

Date:

Name of Director/HoD/Coordinator:

Signature:

Date:

Model Question

Tribhuvan University

Faculty of Humanities & Social Sciences

OFFICE OF THE DEAN

2019

Bachelor in Computer Applications **Full Marks: 60**
Course Title: Data Structures & Algorithms **Pass Marks: 24**
Code No: CACS 201 **Time: 3 hours**
Semester: III

Name: **Symbol No:**

Candidates are required to answer the questions in their own words as far as possible.

Group A

Attempt all the questions. **[10x1 = 10]**

1. Circle (O) the correct answer.

i) What is the measurement for time complexity of an algorithm?

a)Counting microseconds b) Counting kilobytes of algorithms

c) Counting number of key operations d) Counting number of statements

ii) Which of the following is the result of evaluation of 5 7 4 - * 8 4 / +?

a)5 b) 8 c) 10 d) 17

iii) What is the recursive formula for post order traversal of binary tree?

a) Left-Root-Right b) Root-Left-Right

c) Left-Right-Root d) Right-Left-Root

iv) What is the number of disk movement in TOH with 4 disks?

a) 9 b) 14 c) 17 d) 15

v) What is the Big-Oh of best case complexity of insertion sort?

a) O (n) b) O (nlogn) c) O (1) d) O (n^2)

vi) How does the rear index incremented in circular queue?

a) front=(rear+1)%SIZE b) rear=(rear+1)%SIZE

c) rear=rear+1 d) rear=(rear-1)%SIZE

vii) A variation of linked list in which none of the node contains NULL pointer is ……

a) Singly b) Multiple c) Circular d) Doubly

viii) Which of the following data structure is used in depth first search of graph?

a) Stack b) Queue c) Linked List d) None of the above

ix) Which of the following is true for B-Tree of order M?

a) Leaf nodes should be at different level

b) All the key values within a node must be in descending order

c) Every node has at least M children

d) All non-leaf nodes with M-1 keys must have M number of children

x) Which of the following is not a hash function?

a) Division remainder b) Folding

c) Chaining d) Mid square

Tribhuvan University

Faculty of Humanities & Social Sciences

OFFICE OF THE DEAN

2019

Bachelor in Computer Applications **Full Marks: 60**
Course Title: Data Structures & Algorithms **Pass Marks: 24**
Code No: CACS 201 **Time: 3 hours**
Semester: III

Candidates are required to answer the questions in their own words as far as possible.

Group B

Attempt any SIX questions. **[6x5 = 30]**

2. What is Data Structure? Show the status of stack converting following infix expression to prost fix P + Q – (R*S/T+U)-V*W [1+4]
3. Write binary search. Consider a hash table of size 10; insert the keys 62, 37, 36, 44, 67, 91 and 107 using linear probing. [2+3]
4. What are deterministic and non-deterministic algorithms? Explain greedy algorithm. [3+2]
5. Draw a BST from the string DATASTRUCTURE and traverse the tree in post order and preorder. [3+2]
6. Define circular queue? How does circular queue overcome the limitation of linear queue? Explain. [2+3]
7. What is singly linked list? Write an algorithm to add a node at the beginning and end of singly linked list. [1+4]
8. Define AVL tree. Construct AVL tree from given data set: 4, 6, 12, 9, 5, 2, 13, 8, 3, 7, 11. [2+3]

Group C

Attempt any TWO questions. **[2x10 = 20]**

9. What is stack? List the applications of stack. Write an algorithm or procedure to perform PUSH and POP operation in stack. [1+2+7]
10. What is heap? Explain quick sort algorithm with Big-oh notation in best case, average case and worst case and trace it to sort the data: 8, 10, 5, 12, 14, 5, 7, 13. [2+2+6]
11. Define graph and tree data structure. Explain breadth first traversal and depth first traversal with example.