Tribhuvan University
Institute of Science and Technology
Bachelor of Information Technology
Third Semester

Course Title: Data Structures and Algorithms **Full Marks: 60+20+20**

Course Number: BIT 201 **Pass Marks: 24+8+8**

Year/Part: II/I **Total Hours: 45**

Nature of Course: Theory (3 Hrs.) + Lab (3 Hrs.) **Credit Hours: 3**

Course Synopsis: This course contains the concepts of different types of data structures and concepts of algorithms and their analysis.

Course Objective: This course aims to provide sufficient theoretical and practical knowledge of data structure and algorithms required to build efficient programs.

Course Content:

Unit 1: Background and Concept of Data Structures (2hrs)	**Teaching Methodology**	**Teaching Hour**
Concepts of Data Types, Data Structure, Abstract Data Type and their uses Background for Data Structure, Definition and use of ADT, Array as an ADT, Structure, Pointer	Lectures	2 hrs
Unit 2: Algorithms (2hrs)		
Introduction to Algorithm and their properties, Concepts of Analysis of algorithm with asymptotic notations (Big Oh) and their properties, time and space complexities	Lectures	2 hrs
Unit 3: Stack (4hrs)		
Definition and Primitive Operations, Stack as an ADT	Lecture/ Lab	2 hrs
Stack Applications: Evaluation of Infix, Postfix and Prefix expressions, converting from infix to prefix and postfix.		2 hrs
Unit 4: Queue (3hrs)		
Definition, Queue as an ADT and Primitive Operations of Linear and Circular Queue. Application and advantages of Linear, Circular Queue, and Priority Queue (Ascending and Descending Priority Queue)	Lecture/ Lab	3 hrs
Unit 5: Recursion (2hrs)		

Definition and Principle of Recursion, Application of Recursion, Recursion removal using stack, example of recursion for TOH Factorial, Fibonacci Sequences, GCD, efficiency of above recursive algorithms	Lecture/ Lab	2 hrs
Unit 6: List (9hrs)		
List concepts, Definition and List as ADT, Static and Dynamic List Structure and implementation, Types of linked list, Operations on Linked List	Lecture/Lab	2hrs
Singly linked list, Circular Linked List, Doubly Linked List, Doubly Circular Linked List, Inserting, traversing and deleting nodes at beginning, end and specified positions in these linked lists		6 hrs
Linked implementation of a stack and queue in singly linked list		1 hr
Unit 7: Tree (7hrs)		
Definition and basic terminologies of tree, Binary Tree: Introduction, Types of Binary Tree, Level and depth, height balance tree(AVL)	Lecture/ Lab	2 hrs
Operations in Binary Search Tree (BST): Insertion, Deletion, Searching		3 hrs
Tree Traversal: Pre-order traversal , In-order traversal (sorted list of Nodes), Post-order traversal, Applications of Binary Tree (Huff man tree, expression tree)		2 hrs
Unit 8: Sorting (6hrs)		
Introduction and types of sorting Algorithm and implementation of Bubble Sort, Insertion Sort, Selection Sort, Quick Sort, Merge Sort Comparison and Efficiency of sorting algorithms	Lecture/ Lab	6 hrs
Unit 9: Searching (5hrs)		
Introduction Sequential Search, Binary Search and Tree Search Comparison and Efficiency of Searching	Lecture/Lab	3 hrs
Hashing: hash function, hash table and collision resolution techniques		2 hrs
Unit 10: Graph (5hrs)		
Definition, Representation of Graph, Types of Graph	Lecture/ Lab	1 hr
Graph Traversal: Depth First Search, Breadth First Search Spanning Tree, Prim's Algorithm, Kruskal's algorithm and Round Robin Algorithm		2 hrs

Shortest Path Algorithm, Greedy and Dijkstra's Algorithm		2 hrs

Laboratory Works:

Data Structure and Algorithm is highly practical oriented course. Each unit should include plenty of programming practices. Laboratory work should include implementation of Stack, Queue, Lists, Tree, Graphs, and Recursive functions as well as implementation of Sorting Algorithms and Searching Algorithms. Laboratory exercises can be implemented in high level programming languages like C or C++.

Some important contents that should be included in lab exercises are as follows:

Lab Exercises	**Units**
• Write a program to implement array as an ADT. • Writing programs to implement stack operations • Writing programs using stack to convert infix expression to postfix/prefix expression • Write a program to evaluate postfix expression using stack	Unit 1,3
• Writing programs to implement primitive operation in linear and circular queue.	Unit 4
• Writing recursive programs to implement factorial, Fibonacci sequence, GCD, and Tower of Hanoi algorithms	Unit 5
• Writing programs with dynamic memory allocation and de-allocation • Writing programs for operation of linear linked list • Linked list implementation of stack and queue	Unit 6
• Writing programs to implement Binary Search Trees basic operations	Unit 7
• Writing programs to implement sorting algorithms; bubble, insertion, selection, merge and quick sort	Unit 8
• Writing programs to implement: sequential, binary search and hashing	Unit 9

• Writing programs to implement searching, spanning tree and shortest path algorithms in graph	Unit 10

Text Book:

1. Data structure using C and C++, Langsam, Augenstein, Tenenbaum

References Books:

1. Horowitz and Sahni, Fundamentals of Data Structures
2. Aho, Hopcroft and Ullman, Data Structure and Algorithms

Model Question

Full Marks: 60 Pass Marks: 24

Course Title: Data Structure and Algorithms (BIT201) Time: 3 Hours

Section A

Long Answer Questions

Attempt any 2 questions. [2*10=20]

1. Differentiate stack with queue? Trace an algorithm for converting infix expression to postfix for the following infix expression.
   (A+B)*(C$(D-E)+F)-G
2. What are the advantages and disadvantages of linked list over an array? Discuss algorithms for inserting a node at front position of the linked list and deleting its last item in singly linked list.
3. Define sorting problem. Trace quick sort algorithm for the following given list of data and also discuss about its time complexity.

   78 45 23 89 65 12 90 33

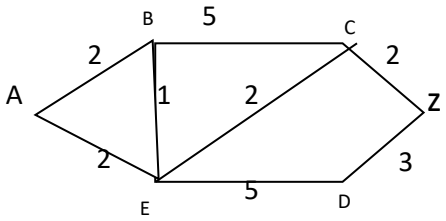
Section B

Short Answer Questions

Attempt any eight questions. [8 × 5 = 40]

4. Define ADT? Explain the benefits of using ADT?
5. Why circular queue is advantageous over linear queue? Write algorithm for enqueue and isfull operation for circular queue.
6. Define recursive algorithm. Write recursive TOH algorithm?
7. Is hashing better than binary search algorithm? Give reasons. Define any two collision resolution techniques.
8. What is a Binary Search Tree? Write an algorithm for searching an item in a binary search tree.
9. What are the different traversing methods in a binary tree? Explain with a clear example.

10. What is a circular linked list? How can you traverse all nodes in a singly linked list?
11. Trace Prim's Algorithm to find minimum spanning tree for the following graph.



12. Write short notes on: (2 × 2.5 = 5)
    a. Analysis of Algorithm
    b. Representation of Graph