

Module Code and Title	ITS202 Algorithms and Data Structures
Programme	BSc in Information Technology
Credit Value	12
Module Tutor	
Module Coordinator	

General Objective

This module provides comprehensive knowledge on the attributes and implementation of fundamental data types, algorithms and data structures with an emphasis on time and space performance analysis.

Learning Outcomes

On completion of the module, students will be able to:

1. Define algorithms and data structures.
2. Determine time and memory complexity for a given algorithm.
3. Implement searching and sorting algorithms.
4. Utilize stack or queue structure to solve a given problem.
5. Apply graphs and trees to develop a programming solution.
6. Select the appropriate data structure to solve a given problem.
7. Use recursion appropriately.

Learning and Teaching Approach

Approach	Hours per week	Total credit hours
Lecture	4	60
Practical	3	45
Independent study	1	15
Total		120

Assessment Approach

Assessments will be carried out on a continuous basis through the following modes:

Assignment

Portion of Final Mark:10%

Students will be required to submit 2 assignments.

Assignment 1 must be submitted before the mid-semester examination related to the topics from unit I, unit II and unit III. Students are required to solve 3-5 problems to check knowledge and analytical skills related to data structure and sorting algorithms.

Assignment 2 must be submitted before the semester end examination, covering topics related to unit IV, V, VI and VII. Here, the students will attempt 3-5 problems to check knowledge and analytical skills related to searching, tree and graphs

Each of these assignments will be evaluated out of 10% and the average of its total will contribute to the final mark.

The assignments will be evaluated based on the following criteria (10%):

5%-Logic

2%-Documentation

2%-Completeness (The completeness of the questions within the assignment)

1%-Timely submission

Class Test

Portion of Final Mark: 15%

This is a closed book, written test conducted to assess the concepts for a duration of 1 hour. There will be two such tests. Each of these tests will be evaluated out of 15% and the average of its total will contribute to the final mark.

Class test 1 will be before the mid-semester examination and covers subject matters from unit I, unit II and unit III. It will test students' theoretical and analytical ability on data structure and sorting algorithms.

Class test 2 will be after the mid-semester examination, comprising topics from unit IV, V, VI and VII to evaluate student knowledge on searching, trees and graphs.

Practical Examination

Portion of Final Mark: 15%

This is a closed book test conducted within the class for the duration of 3 hours. There will be two such tests. It is hands-on testing of the programming skills.

Practical exam 1 will be a week or two before the mid-semester examination and covers subject matters from unit I and II. In this test, students will have to solve 3-5 small problems, where they have to write a trace table, design flow chart or basic program. This will assess a student's understanding of techniques and principles mentioned in the subject matter.

Practical exam 2 will be a week or two before the semester-end examination, comprising topics from unit IV, unit V and unit VI. In this test, students will have to solve 3 questions of increasing difficulty. This will test the student's fundamental knowledge related to functions, manipulation of tuples/lists/dictionaries and string operation.

Each of these tests will be evaluated out of 15% and the average of its total will contribute to the final mark.

The practical tests will be evaluated based on the following criteria (15%):

2%-Appropriate comments

3%-Program logic

5%-Subtask completeness

5%-Output correctness

Mid-semester Examination

Portion of Final Marks: 20%

This is a college wide examination conducted at the half-way into the semester. This examination is conducted to check student understanding on all topics till the half-way point in the subject matter.

Semester End Examination

Portion of Final Marks: 40%

This is a written examination conducted at the end of the semester to assess their conceptual understanding of the topics covered in this module

Overview of the assessment approaches and weighting

Areas of assessment	Quantity	Weighting
Assignment	2	10%
Class Test	2	15%
Practical Examination	2	15%
Mid-semester Examination	1	20%
Total Continuous Assessment (CA)		60%
Final Examination (FE)		40%

Prerequisites: MAT210 Discrete Mathematics, ITP102 Object Oriented Programming Fundamentals

Subject Matter

Unit I: Introduction

- 1.1 Introduction to computer algorithm and data structure, Iterative and recursive algorithm design approaches
- 1.2 Basic introduction to algorithmic paradigms like divide and conquer, recursion, greedy

- 1.3 Asymptotic Analysis and notations: Big oh, Big Theta and Big omega notations
- 1.4 Introduction to complexity analysis of algorithms: worst case, best case and average case

Unit II: Fundamental Data Structures

- 2.1 Representations for Arrays and linked list, types of linked list
- 2.2 Defining the Stack ADT using arrays and linked list, Application of Stacks
- 2.3 Defining the Queue ADT using arrays and linked list, Application of Queues

Unit III: Sorting Algorithms

- 3.1 Elementary sorts: Bubble sort, selection sort, insertion sort, shell sort and their performance analysis
- 3.2 Understanding Divide and Conquer approach: Merge Sort and its performance analysis, quicksort and its performance analysis
- 3.3 Heap sort and their performance analysis
- 3.4 Applications for sorting algorithms
- 3.5 Radix sort (LSD, MSD)

Unit IV: Searching Algorithms

- 4.1 Linear search algorithm and its performance analysis
- 4.2 Binary search algorithm and its performance analysis
- 4.3 Symbol table and their performance analysis
- 4.4 Applications of various searching algorithms

Unit V: Advanced Data Structures

- 5.1 Trees, Binary trees, types of Binary trees
- 5.2 Binary Search trees and its applications
- 5.3 Balanced binary search trees (Red-black trees/2-3 trees), AVL trees, B-trees
- 5.4 Hash tables, hashing with chaining, hashing with linear probing and their performance analysis
- 5.5 Trees and hashing Application.

Unit VI: Graphs

- 6.1 Introduction to Graphs and types of Graphs: Directed and Undirected graphs
- 6.2 Graph Traversals: Depth First Search and Breadth First Search
- 6.3 Shortest Path Problem: Dijkstra's algorithm, Greedy algorithm, Bellman For
- 6.4 Shortest path and Traveling salesman problem (TSP)

Unit VII: Spanning Trees

- 7.1 Minimum Spanning Trees concept and cut property
- 7.2 Greedy Minimum Spanning Trees algorithm
- 7.3 Edge-weighted graph
- 7.4 Prim's algorithm (Lazy Prim, Eager Prim)
- 7.5 Kruskal's algorithm

List of Practicals

1. Implementation of linked list data structure.
2. Implementation of push and pop methods in stack ADT using either array or linked list data structure.
3. Implementation of enqueue and dequeue methods in queue ADT using either array or linked list data structure.
4. Implementation of sorting algorithms (selection sort, insertion sort, merge sort, quick sort, heap and radix sort).
5. Implementation of searching algorithms (linear search, sequential search, binary search and binary search tree).

6. Implementation of tree data structure.
7. Implementation of hash table.
8. Implementation of directed and undirected graphs.
9. Implementation of undirected edge-weighted and directed edge-weighted graphs.

Essential Reading

Goodrich, T.M., Tamassia, R., & Goldwasser, H.M.(2014). *Data structures and algorithms in Java* (6th ed.). New Jersey, NJ: Wiley.

Sedgewick, R. & Wayne, K. (2011). *Algorithms* (4th ed.). New Jersey, NJ: Pearson Education.

Additional Reading

Narasimha, K. (2015). *Data structures and algorithms made easy in Java*. Hyderabad: CareerMonk Publications.

Goodrich, M., & Tamassia, R.(2010). *Data structures and algorithms in Java* (5th ed.). New Jersey, NJ :Wiley

Date: 10th Sep, 2019.