Second Semester 2023-2024

Date:09th January 2024

In addition to part-I (General Handout for all courses appended to the time table) this portion gives further specific details regarding the course.

Course No.	: **BITS F232**
Course Title	: **Foundations of Data Structures and Algorithms**
Instructor-in-Charge	: Sameera Muhamed Salam
Instructors	: Prof Tathagata Ray

Scope and Objective of the Course:

A data structure is a collection of large amounts of data values, the relationships among them, and the functions or operations that can be applied on them. In order to be effective, data has to be organized in a manner that adds to the effectiveness of an algorithm, and data structures such as stacks, queues, linked lists, heaps, trees, and graphs provide different capabilities to organize and manage large amounts of data. While developing a program or an application, many developers find themselves more interested in the type of algorithm used rather than the type of data structure implemented. However, the choice of data structure used for a particular algorithm is always of paramount importance. For example, B-trees have unique abilities to organize indexes and hence are well suited for implementation of databases; Linked lists are well suited for backtracking algorithms like, accessing previous and next pages in a web browser; Tries are well suited for implementing approximate matching algorithms like, spell checking software or predicting text in dictionary lookups on Mobile phones; Graphs are well suited for path optimization algorithms (like in Google maps) or searching in a Social graph (like Facebook). As computers have become faster and faster, the problems they must solve have become larger and more complex, requiring development of more complex programs. This course will also teach students good programming and algorithm analysis skills so that they can develop such programs with a greater degree of efficiency.

The primary objectives of the course are as under:

- Apply various basic data structures such as stacks, queues, linked lists, trees etc. to solve complex programming problems. Understand basic techniques of algorithm analysis.
- Design and implement advanced data structures like graphs, balanced search trees, hash tables, priority queues etc. Apply graph and string algorithms to solve real world problems like finding shortest paths on huge maps or detecting plagiarism percentage.
- Apply basic algorithmic techniques such as brute-force, greedy algorithms, divide and conquer, dynamic programming etc. to solve complex programming problems and examine their efficiency.

At the end of the course, you should understand common data structures and algorithms, be able to develop new data abstractions (interfaces) .

Textbooks:

T1. Cormen TH, Leiserson CE, Rivest RL, and C Stein. **Introduction to Algorithms.** MIT Press Second Edition (India reprint: Prentice-Hall 2009).

Reference books

R1. Micheal T Goodrich and Roberto Tamassia. **Algorithm Design: Foundations, Analysis and Internet examples.** (John Wiley &Sons, Inc., 2002).

R2. Jon Kleinberg and Eva Tardos. **Algorithm Design.** Pearson Education. (2007).

R3. Sanjoy Das Gupta, Christos Papadimitriou, Umesh Vazirani, **Algorithms.** Tata McGraw-Hill Publishers.

Course Plan:

Lecture No.	Learning objectives	Topics to be covered	Chapter in the Text Book
1-2	Tounderstand the role of DS and Algorithms in Computing.	Course Introduction & Motivation.	T1-1
3 - 6	To understand analysis of algorithms	Growth of Functions & Asymptotic Notation , Simple Case Studies: Binary search & Bubble Sort	T1-2,3,4 R1-1
7-8	To understand Divide and Conquer Algorithmic Technique	Using arrays , Divide & Conquer, Merge Sort & Quick Sort (Analysing lower bounds)	T1-7 R2 – 5
9-10	To understand Linear time sorting algorithms	Linear time Sorting Algorithms – Radix Sort and Bucket Sort (Analysing lower bounds)	T1 – 2, 6, 8 R1- 4
11 – 16	To understand basic datastructures, their implementations, Complexity, Efficiency & Applications	Insertion and removal from a Linked list, generic single linked list, doubly linked lists, circular linked lists, Stack ADT, Queue ADT, Double ended Queue , Vectors, Lists & Sequences	T1-10 R1-2
17-18	To understand Tree Data Structure	Tree ADT, Binary Tree, Types of Binary tree,	R1 - 2
19 - 22	To understand Advanced data structures like Priority queues, Heaps, Hash tables, Maps, Skip	Priority Queue ADT, Heaps, Applications of heap: Insertion Sort, Selection Sort & Heap Sort	
23 - 24		MAP ADT, Dictionories and Hash Tables, Separate Chaining vs. Open Addressing, Probing, Rehashing.	T1 – 11 R1- 2

25 - 30	lists, Dictionaries, Search Trees.	Binary Search Tree, Balanced Binary SearchTrees - Red-Black Trees, Skip list (Implementation, Complexity & Efficiency)	T1 – 12, 13 R1- 3
31 - 36	To understand String Manipulation and Dynamic Programming Algorithmic Technique	Trie Data Structure, Pattern Matching Algorithms, LCS using Dynamic Programming	T1 – 4,15, 32 R1- 9
37 - 38	To understand Graph Data Structure and Greedy Algorithmic technique	Graphs ADT& Graph Algorithms: Representation schemes, Traversals:DFS and BFS	T1 – 22, R1 - 6
39 - 42		Greedy Algorithms: Shortest path and MST (Dijkstra, Kruskal, and Prim-Jarnik algorithms.)	T1- 23, 24 R3 – 4,5

Evaluation Scheme:

Component	Duration	Weightage (%)	Date & Time	Nature of Component
Mid Test	90 minutes	25%	15/03 - 4.00 - 5.30PM	Closed Book
Lecture-Continous Evaluation	10 Minutes	10 %	One per week	Closed Book
Lab–Continuous Evaluation (Assignments)	Every assignment will be evaluated.	10%	TBA	Open Book
Lab Final Test	One Hour	15%	TBA	Open Book
Comprehensive	3 hours	40%	17/05 AN	Closed Book

Note: Minimum 40% of the evaluation to be completed by midsem grading.

Chamber Consultation Hour: Monday (5 PM to 6 PM) Room No: H107

Notices: All notices pertaining to this course will be displayed on the CSIS Notice Board/CMS.

Make-up Policy: Prior Permission is mustand Make-up shall be granted only in genuine cases based on individual’s need, circumstances. The recommendation from chief warden is necessary to request for a make-up.

Academic Honesty and Integrity Policy: Academic honesty and integrity are to be maintained by all the students throughout the semester and no type of academic dishonesty is acceptable.

INSTRUCTOR-IN-CHARGE

innovate achieve lead