BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
WORK INTEGRATED LEARNING PROGRAMMES
COURSE HANDOUT

Part A: Content Design

Course Title	DATA STRUCTURES AND ALGORITHMS DESIGN
Course No(s)	DSECLZG519
Credit Units	**5**
Course Author	Febin. A.Vahab (01/10/2020)
Version No	2.0
Minor Edits	Parthasarathy PD (01/11/2023)

Course Description

The course covers design, implementation and applications of basic and advanced data structures including trees, graphs, bloom filters. The course also covers algorithm design techniques like greedy, dynamic, map reduce etc. using examples from sorting, searching, graph theory, networking and number theory. The complexity issues are also discussed further.

Course Objectives

No	Objective
CO1	Introduce mathematical and experimental techniques to analyze algorithms
CO2	Introduce linear and non-linear data structures and best practices to choose appropriate data structure for a given application
CO3	Teach various dictionary data structures (Lists, Trees, Heaps, Bloom filters) with illustrations on possible representation, various operations and their efficiency
CO4	Exposes students to various sorting and searching techniques
CO5	Discuss in detail various algorithm design approaches (Greedy method, divide and conquer and dynamic programming) with appropriate examples, methods to make correct design choice and the efficiency concerns
CO6	Introduce complexity classes , notion of NP-Completeness, ways of classifying problem into appropriate complexity class
CO7	Introduce reduction method to prove a problem's complexity class.

Learning Outcomes:

No	Learning Outcomes
LO1	Describe various fundamental and advanced data structures, their properties, algorithm design techniques and various means of evaluating algorithms
LO2	Demonstrate the ability to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in a particular context.
LO3	Solve problems using Algorithms for Linear and Non-Linear Data Structures
LO4	Explain with a practical example, each of the algorithm design strategies (greedy, divide-and-

	conquer, dynamic programming and map-reduce)
LO5	Use brute-force, greedy, divide-and-conquer, recursive backtracking, dynamic programming and map reduce techniques to solve a given algorithm design problem.
LO6	Relate the real-world problems to known data structures and algorithms leading to the recommend appropriate solutions in representation and implementation.
LO7	Explain the significance of NP-completeness
LO8	Classify problems into complexity classes P and NP and to prove hardness of problems

Textbook(s):

No	Author(s), Title, Edition, Publishing House
T1	Algorithms Design: Foundations, Analysis and Internet Examples Michael T. Goodrich, Roberto Tamassia, 2006, Wiley (Students Edition)

Reference Book(s) & other resources:

No	Author(s), Title, Edition, Publishing House
R1	Introduction to Algorithms, TH Cormen, CE Leiserson, RL Rivest, C Stein, Third Ed, 2009, PHI
R2	Computer Algorithms, Ellis Horowitz, Sartaj Sahni and Rajasekaran, 2nd Edition.
R3	Data Structures & Algorithms in Python , Michael T. Goodrich, Roberto Tamassia, Michael H Goldwasser, Wiley, 2013

CONTENT STRUCTURE

No	Title of the Module	References
M1	**Analyzing Algorithms** 1.1. Theoretical Foundation 1.1.1. Algorithms and it's Specification 1.1.2. Random Access Machine Model 1.1.3. Notion of best case, average case and worst case 1.1.4. Notion of Algorithm Correctness 1.2. Characterizing Run Time 1.2.1. Use of asymptotic notation 1.2.2. Big-Oh, Omega and Theta Notations 1.3. Analyzing Recursive Algorithms 1.3.1. Recurrence relations 1.3.2. Specifying runtime of recursive algorithms 1.3.3. Master Theorem 1.3.4 Solving Recurrences: Substitution Method, Recursion Tree Method	T1: 1.1, 1.2 T1:1.1.4 R1: 4.3,4.4,4.5
M2	**Elementary Data Structures** 2.1. Stacks ADT , Implementation and Applications 2.2. Queues ADT , Implementation and Applications 2.3. Amortized Analysis – Stack, Queue operations- Aggregate Method 2.4. List ADT , Implementation and Applications	R1:10.1 R1:17.1 R1:10.2

M3	**Non-Linear Data Structures** 3.1. Trees 3.1.1. Terms and Definition 3.1.2. Tree ADT 3.1.3. Applications 3.2. Binary Trees 3.2.1. Properties 3.2.2. Representations (Array Based and Linked Structure) 3.2.3. Binary Tree traversal (In Order, Pre Order, Post Order) 3.2.4. Applications 3.3. Heaps 3.3.1. Definition and Properties 3.3.2. Representations (Array Based and Linked) 3.3.3. Insertion and deletion of elements 3.3.4. Heap sort 3.3.5. Priority Queue 3.4. Graphs 3.4.1. Terms and Definitions 3.4.2. Properties 3.4.3. Representations (Edge List, Adjacency list, Adjacency Matrix) 3.4.4. Graph Traversals (Depth First and Breadth First Search) 3.5.5. Applications 3.5. Directed Graph and Reachability- Floyd-Warshall's Transitive Closure	T1: 2.3 R2:6 R1: 22.1, 22.2,22.3 R1:25.2
M4	**Dictionaries** 4.1. Dictionary ADT , Applications 4.2. Hash Tables 4.2.1. Notion of Hashing and Collision 4.2.2. Methods for Collision Handling 4.2.2.1. Separate Chaining 4.2.2.2. Notion of Load Factor 4.2.2.3. Rehashing 4.2.2.4. Open Addressing [Linear & Quadratic Probing, Double Hash] 4.2.2.5. Applications 4.3. Universal Hashing 4.4. Introduction to Bloom Filters, Applications 4.5. Binary Search Tree 4.5.1. BST Operations 4.5.2. Applications 4.6. AVL trees 4.7. Rank and Range Queries, Performance 4.8 k-d Trees 4.6.1 Representation 4.6.2 Insertion, Deletion and Complexity	R2:11 Bloom Filter R1: 12 T1:3.1 T1:3.2 T1:12.1 T1:12.3.2

M5	**Algorithm Design Techniques** 5.1. Greedy Method 5.1.1. Design Principles and Strategy 5.1.2. Fractional Knapsack Problem 5.1.3. Minimum Spanning Tree 5.1.4. Shortest Path Problem - Djikstra's Algorithm 5.1.5. Task Scheduling Problem 5.2. Divide and Conquer 5.2.1. Design Principles and Strategy 5.2.2. Integer Multiplication Problem 5.2.3. Merge Sort 5.2.4. QuickSort 5.3. Dynamic Programming 5.3.1. Design Principles and Strategy 5.3.2. Matrix Chain Product Problem 5.3.3. All-pairs Shortest Path Problem 5.3.4. 0/1 Knapsack Problem	T1: 5.1, 7.3,7.1.1 T1: 5.2.2, 4.1,4.3 T1: 5.3,7.2
M6	**Complexity Classes** 6.1. Definition of P and NP classes and examples 6.2. Understanding NP-Completeness: CNF SAT 6.3. Cook-Levin theorem 6.4. Polynomial time Reducibility: 6.4.1 CNF SAT 6.4.2 Clique	T1: 13

Part B: Session Plan

Academic Term	**2023-2024 First Semester**
Course Title	**DATA STRUCTURES AND ALGORITHMS DESIGN**
Course No	**DSECCZG519**
Lead Instructor	Prof Parthasarathy

SESSION CONTENTS

Session (#)	List of Topic Title (from content structure in Course Handout)	Text/Ref Book
1	**Analyzing Algorithms** **Theoretical Foundation** • Algorithms and it's Specification • Random Access Machine Model • Notion of best case, average case and worst case • Notion of Algorithm Correctness	T1: 1.1, 1.2
2	**Analyzing Algorithms (Continued…)** **Characterizing Run Time** • Use of asymptotic notation • Big-Oh, Omega and Theta Notations **Analyzing Recursive Algorithms** • Recurrence relations • Specifying runtime of recursive algorithms • Master Theorem	T1:1.1.4 R1: 4.3,4.4,4.5
3	**Elementary Data Structures** • Stacks ADT , Implementation and Applications • Queues ADT , Implementation and Applications • Amortized Analysis -Stack, Queue operations-Aggregate Method • List ADT , Implementation and Applications	R1:10.1 R1:17.1 R1:10.2

4	**Non-Linear Data Structures** **Trees** • Terms and Definition • Tree ADT • Applications **Binary Trees** • Properties • Representations (Array Based and Linked Structure) • Binary Tree traversal (In Order, Pre Order, Post Order) • Applications	T1: 2.3
5	**Heaps** • Definition and Properties • Representations (Array Based and Linked) • Insertion and deletion of elements • Heap sort • Priority Queue	R2:6
6	**Graphs** • Terms and Definitions • Properties • Representations (Edge List, Adjacency list, Adjacency Matrix) • Graph Traversals (Depth First and Breadth First Search) • Applications [Self-Study]	R1: 22.1, 22.2,22.3
7	**Binary Search Tree** • BST Operations • Applications **Dictionaries** • Dictionary ADT , Applications • Hash Tables • Notion of Hashing and Collision	R1 and T1
8	**Methods for Collision Handling** • Separate Chaining • Notion of Load Factor • Rehashing • Open Addressing [Linear &Quadratic Probing, Double Hash] • Applications [Self-Study]	R2 and T1
MID – SEMESTER EXAM COVERING CS1 to CS8 topics (30Marks)		
9	**Esoteric Data Structures:** **Introduction to Bloom Filters, Applications** **k-d Trees** • Representation • Insertion, Deletion and Complexity	Class Notes
10	**AVL Trees** • Rotations, Insertion, Deletion.	T1:3.1,3.2
11	**Algorithm Design Techniques** **Greedy Method** • Design Principles and Strategy • Fractional Knapsack Problem • Minimum Spanning Tree - Kruskal	T1:12.1 T1:12.3.2
12	**Greedy Method (Continued…)** • Minimum Spanning Tree - Prims • Shortest Path Problem - Djikstra's Algorithm **Divide and Conquer** • Design Principles and Strategy • Merge Sort	T1: 5.1

13	• Integer Multiplication Problem • Quick Sort **Dynamic Programming** • Design Principles and Strategy • Fibonacci Series	T1: 7.3,7.1.1
14	• Matrix Chain Product Problem • Directed Graph & Reachability – Warshall's Algorithm • All-pairs Shortest Path Problem – Floyd's Algorithm	T1: 5.2.2, 4.1
15	• 0/1 Knapsack **Complexity Classes** • Definition of P and NP classes and examples • Understanding NP-Completeness: CNF SAT	T1: 5.3
16	**Complexity Classes (Continued…)** • Cook-Levin theorem Polynomial time Reducibility: • CNF SAT • Clique	T1: 7.2 T1: 13

Webinar: Topics and date/time will be informed by the instructor via Canvas.

Evaluation Scheme

Legend: EC = Evaluation Component

No	Name	Type	Duration	Weight	Day, Date, Session, Time
EC-1	Assignment-1 (12%)	Will be informed		30%	Will be informed in Canvas
	Assignment-2 (13%)				
	Quiz (Best of 2 – 5%)				
EC-2	Mid Term	**Closed Book**		30%	
EC-3	Comprehensive Exam	**Open Book**		40%	

Note *- Evaluation components can be tailored depending on the proposed model.*

Important Information

Syllabus for Mid-Semester Test (Closed Book): Topics in Weeks 1-8
Syllabus for Comprehensive Exam (Open Book): All topics given in plan of study
Evaluation Guidelines:

1. EC-1 consists of two Assignments (25% weightage) and 2 Quizzes (best score of quiz will be considered – 5%). Announcements regarding the same will be made in a timely manner.
2. For Open Book exams: Use of prescribed and reference textbooks, in original (not photocopies) is permitted. Class notes/slides as reference material in filed or bound form is permitted. However, loose sheets of paper will not be allowed. Use of calculators is permitted in all exams. Laptops/Mobiles of any kind are not allowed. Exchange of any material is not allowed.
3. If a student is unable to appear for the Regular Test/Exam due to genuine exigencies, the student should follow the procedure to apply for the Make-Up Test/Exam. The genuineness of the reason for absence in the Regular Exam shall be assessed prior to giving permission to appear for the Make-up Exam. Make-Up Test/Exam will be conducted only at selected exam centers on the dates to be announced later.

It shall be the responsibility of the individual student to be regular in maintaining the self-study schedule as given in the course handout, attend the lectures, and take all the prescribed evaluation components such as Assignment/Quiz, Mid-Semester Test and Comprehensive Exam according to the evaluation scheme provided in the handout.

********** ALL THE BEST **********