BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Birla Institute of Technology & Science, Pilani
Work Integrated Learning Programmes Division
Second Semester 2024-2025

Digital Learning Handout

Part A: Content Design

Course Title	DATA STRUCTURES AND ALGORITHMS DESIGN
Course No(s)	SE ZG519/SS ZG519
Credit Units	5
Credit Model	X-X-X (XX Hours of Class-room Instruction + XX Hours of Case-studies/Tutorials/Laboratories + XX Hours of Student Preparation)
Instructors	**Prof. Rajib Ranjan Maiti (LEAD), Febin Vahab**
Version No:	
Date:	16/01/2025

Course Description:

Commented [1]: Course description should be as per Senate approved document

The course includes the lesson to design, implement and apply a set of basic and advanced data structures, including trees, graphs and bloom filters. The course also includes important algorithm design techniques, like greedy, dynamic programming, map reduce etc. and their applications to develop solutions for sorting, searching, graph searching, networking and number theory. The course also includes the techniques to measure the performance of algorithm in terms of their time complexity.

Course Objectives

No	Course Objective
CO1	Equip students with the ability to analyse the performance and correctness of algorithms using theoretical foundations and asymptotic notations, preparing them for advanced study and professional application in algorithm design.
CO2	Enable students to proficiently analyse and solve recursive algorithms through the understanding and application of recurrence relations, the Master Theorem, and various methods for solving recurrences.
CO3	Ensure that students gain a deep understanding of both elementary and advanced data structures, including their implementation, applications, and optimization techniques, to solve complex computational problems effectively.
CO4	Provide students with the knowledge and skills to implement and utilize advanced data structures such as hash tables, AVL trees, k-d trees, and Bloom filters, focusing on collision handling, rehashing, and other optimization techniques.
CO5	Train students in employing various algorithm design techniques such as greedy methods, divide and conquer, and dynamic programming to develop efficient solutions for a wide range of computational problems.

CO6	Foster a comprehensive understanding of computational complexity theory, including the classification of problems into P and NP classes, the concept of NP-completeness, and the application of polynomial-time reducibility to significant computational problems.

Text Book(s):

T1	Algorithms Design: Foundations, Analysis and Internet Examples Michael T. Goodrich, Roberto Tamassia, 2006, Wiley (Students Edition)

Reference Book(s) & other resources:

R1	Introduction to Algorithms, TH Cormen, CE Leiserson, RL Rivest, C Stein, Third Ed, 2009, PHI
R2	Data Structures, Algorithms and Applications in Java, Sartaj Sahni, Second Ed, 2005, Universities Press

Commented [2]: With edition and year of publication

Commented [3]: latest edition of book

Learning Outcomes: Students will be able to

LO1	Demonstrate the ability to analyse the efficiency of algorithms using asymptotic notation and characterize their run-time complexities. Understand and apply concepts such as best case, average case, and worst case scenarios, as well as the correctness of algorithms.
LO2	Develop the skills to analyse recursive algorithms using recurrence relations and the Master Theorem. Solve recurrence relations using methods such as substitution and recursion trees to specify the runtime of recursive algorithms.
LO3	Implement and utilize various data structures including stacks, queues, lists, trees (binary trees, AVL trees, k-d trees), heaps, and graphs. Analyse and optimize the operations associated with these data structures for different applications.
LO4	Understand and implement advanced data structures such as hash tables, including collision handling methods, and Bloom filters. Apply these data structures to solve complex problems efficiently.
LO5	Employ algorithm design techniques such as the greedy method, divide and conquer, and dynamic programming to solve various computational problems. Implement algorithms for problems like the knapsack problem, shortest path problems, and task scheduling.
LO6	Understand the fundamental concepts of computational complexity, including P and NP classes, NP-completeness, and polynomial-time reducibility. Apply these concepts to problems such as CNF SAT and the Clique problem, and comprehend the implications of the Cook-Levin theorem.

Part B: Learning Plan

Contact Session	List of Topic Title	Sub-Topics	Reference

1	**Analyzing Algorithms Theoretical Foundation**	1.1 Algorithms and it's Specification 1.2 Random Access Machine Model 1.3 Notion of best case, average case and worst case 1.4. Notion of Algorithm Correctness	T1: 1.1, 1.2
2	**Characterizing Run Time**	**2.1. Characterizing Run Time** 2.1.1. Use of asymptotic notation 2.1.2. Big-Oh, Omega and Theta Notations **2.2. Analyzing Recursive Algorithms** 2.2.1. Recurrence relations 2.2.2. Specifying runtime of recursive algorithms 2.2.3. Master Theorem	T1:1.1.4 R1: 4.3,4.4,4.5
3	**Elementary Data Structures**	3.1. Stacks ADT, Implementation and Applications 3.2. Queues ADT, Implementation and Applications 3.3. Amortized Analysis -Stack, Queue operations-Aggregate Method 3.4. List ADT , Implementation and Applications	R1:10.1 R1:17.1 R1:10.2
4	**Non-Linear Data Structures**	**4.1. Trees** 4.1.1. Terms and Definition 4.1.2. Tree ADT 4.1.3. Applications **4.2. Binary Trees** 4.2.1. Properties 4.2.2. Representations (Array Based and Linked Structure) 4.4.3. Binary Tree traversal (In Order, Pre Order, Post Order) 4.3.4. Applications	T1: 2.3
5	**Heaps**	5.1. Definition and Properties 5.2. Representations (Array Based and Linked) 5.3. Insertion and deletion of elements 5.4. Heap sort 5.5. Priority Queue	R2:6
6	**Graphs**	6.1. Terms and Definitions 6.2. Properties 6.3. Representations (Edge List, Adjacency list, Adjacency Matrix) 6.4. Graph Traversals (Depth First and Breadth First Search) 6.5. Applications	R1: 22.1, 22.2,22.3

7	**Graphs (contd..) and Dictionaries**	7.1. Directed Graph and Reachability-Floyd-Warshall's Transitive Closure **7.2. Dictionaries** 7.2.1. Dictionary ADT, Applications 7.2.2. Hash Tables 7.2.3. Notion of Hashing and Collision **7.3. Methods for Collision Handling** 7.3.1. Separate Chaining 7.3.2. Notion of Load Factor 7.3.3. Rehashing	R1:25.2 R2:11
8	**Methods for Collision Handling (Continued…)**	8.1. Open Addressing [Linear &Quadratic Probing, Double Hash] 8.2. Applications Universal Hashing 8.3. Introduction to Bloom Filters, Applications	R2:11
9	**Binary Search Tree**	9.1. BST Operations Applications 9.2. AVL trees	T1:3.1,3.2
10	AVL Trees (Contd.) and K-D trees	10.1. Rank and Range Queries, Performance **10.2. k-d Trees** 10.2.1. Representation 10.3. Range and NN Queries	T1:12.1 T1:12.3.2
11	**Algorithm Design Techniques Greedy Method**	11.1. Design Principles and Strategy 11.2. Fractional Knapsack Problem	T1: 5.1
12	**Greedy Method (Continued…)**	12.1. Minimum Spanning Tree 12.2. Shortest Path Problem - Djikstra's Algorithm	T1: 7.3,7.1.1
13	**Divide and Conquer**	13.1. Design Principles and Strategy 13.2. Integer Multiplication Problem 13.3. Merge Sort	T1: 5.2.2, 4.1
14	**Dynamic Programming**	14.1. Design Principles and Strategy 14.2. Matrix Chain Product Problem	T1: 5.3
15	**Complexity Classes**	15.1. All-pairs Shortest Path Problem **15.2. Complexity Classes** 15.2.1. Definition of P and NP classes and examples	T1: 7.2 T1: 13.1
16	**Complexity Classes (Continued…)**	16.1. Understanding NP-Completeness: CNF SAT 16.1.1. Cook-Levin theorem 16.2. Polynomial time Reducibility: 16.2.1. CNF SAT	T1: 13

		16.3. Clique	

Experiential Learning Components:

Describe objective, outcome of Experiential Learning Component and the lab infrastructure needed (virtual, remote, open source etc..) number of lab exercises needed, etc.

1. Lab work: NA
2. Project work: NA
3. Case Study: NA
4. Simulation: NA
5. Work Integrated Learning Assignment: NA
6. Design work/ Field work: NA

Commented [4]: added common nomenclature as experiential learning component instead of lab. Added simulation and work integrated Learning Assignment

Commented [5]: Mention which component will be used in Experiential Learning. For rest of components mark it as NONE

Eg:

Objective of Experiential Learning Component:

Scope of Experiential Learning Component:

Lab Infrastructure:

List of Experiments: **NA**

Exp No.	Experiment Title	Reference to handout module/section
1.		
2.		

Commented [6]: If required add Instruction Schedule for the lab

Evaluation Scheme:

Legend: EC = Evaluation Component; AN = After Noon Session; FN = Fore Noon Session

Evaluation Component	Name (Quiz, Lab, Project, Mid-term exam, End semester exam, etc.)	Type (Open book, Closed book, Online, etc.)	Weight	Duration	Day, Date, Session, Time
EC – 1*	Quiz	Online	10%	1 week	February 17-27, 2025
	Assignment/Lab Assignment / Lab Exams	Online	20 %	10 days	April 1-10, 2025
EC - 2	Mid-Semester Test	Closed Book	30%	2 hours	22/03/2025 (FN)
EC - 3	Comprehensive Exam	Open Book	40%	2 ½ Hours	24/05/2025 (FN)

Commented [7]: If required describe the EC1 evaluation components and its weightage in detailed manner (including Lab components/simulation components) etc.

EC1* (20% - 30%): Quiz (optional): 5-10 %, Lab Assignment/Assignment: 20% - 30%
Syllabus for Mid-Semester Test (Closed Book): Topics in Contact session: 1 to 8
Syllabus for Comprehensive Exam (Open Book): All topics

Important Links and Information:

eLearn Portal: https://elearn.bits-pilani.ac.in
Students must visit the eLearn portal regularly and stay updated with the latest announcements and deadlines.
Contact Sessions: Students should attend the online lectures as per the schedule provided on the eLearn portal.
Evaluation Guidelines:

1. EC-1 consists of either two Assignments or three Quizzes. Students will attempt them through the course pages on the eLearn portal. Announcements will be made on the portal in a timely manner.
2. For Closed Book tests: No books or reference material of any kind will be permitted.
3. For Open Book exams: "open book" means text/ reference books (publisher copy only) and does not include any other learning material. No other learning material will be permitted during the open book examinations. For Detailed Guidelines refer to the attached document. EC3 Guidelines
4. 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, which will be made available on the eLearn portal. The Make-Up Test/Exam will be conducted only at selected exam centres 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 online lectures, and take all the prescribed evaluation components such as Assignments/Quizzes, Mid-Semester Tests and Comprehensive Exams according to the evaluation scheme provided in the handout.

****