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)	
Credit Units	
Course Author	
Version No	
Date	

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, dynamic programming and map reduce) 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
C07	Introduce reduction method to prove a problem's complexity class.

Text Book(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	Data Structures, Algorithms and Applications in Java, Sartaj Sahni, Second Ed, 2005, Universities Press
R2	Introduction to Algorithms, TH Cormen, CE Leiserson, RL Rivest, C Stein, Third Ed, 2009, PHI
R3	Handbook of Data Structures and Applications, Dinesh P. Mehta and Sartaj Sahni. Second Ed. 2018 CRC Press

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. Counting Primitive Operations 1.1.4. Notion of best case, average case and worst case 1.2. Characterizing Run Time 1.2.1. Use of asymptotic notation 1.2.2. Big-Oh Notation, Little-Oh, Omega and Theta Notations 1.3. Correctness of Algorithms 1.4. Analyzing Recursive Algorithms 1.4.1. Recurrence relations 1.4.2. Specifying runtime of recursive algorithms 1.4.3. Solving recurrence equations 1.5. Case Study: Analyzing Algorithms	T1: 1.1, 1.2, 1.4
M2	**Elementary Data Structures** 2.1. Stacks 2.1.1 Stack ADT and Implementation 2.1.2. Applications 2.2. Queues 2.2.1. Queue ADT and Implementation 2.2.2. Applications 2.2.3. Amortized Analysis – Stack, Queue operations 2.3. List 2.3.1. Notion of position in lists 2.3.2. List ADT and Implementation 2.4 Sets 2.4.1 Set ADT and Implementation	T1: 2.1, 1.5, 2.2, 4.2
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 (Vector Based and Linked) 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	T1: 2.3, 2.4, 6.1, 6.2, 6.3, 6.4 R3: 31 http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf

	3.3.2. Representations (Vector Based and Linked) 3.3.3. Insertion and deletion of elements 3.3.4. Heap implementation of priority queue 3.3.5. Heap sort 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) 3.4.5. Directed Graph and Reachability 3.4.5.1. Applications :- Page Rank Algorithm 3.4.6. Dynamic Graphs	
M4	**Dictionaries [as Hash Tables and Search Trees]** 4.1. Unordered Dictionary 4.1.1. ADT Specification 4.1.2. Applications 4.2. Hash Tables 4.2.1. Notion of Hashing and Collision (with a simple vector based hash table) 4.2.2. Hash Functions 4.2.2.1. Properties 4.2.2.2. Simple hash functions 4.2.3. Methods for Collision Handling 4.2.3.1. Separate Chaining 4.2.3.2. Notion of Load Factor 4.2.3.3. Rehashing 4.2.3.4. Open Addressing [Linear & Quadratic Probing, Double Hash] 4.3. Ordered Dictionary 4.3.1. ADT Specification 4.3.2. Applications 4.4. Bloom Filters 4.4.1. Motivation 4.4.2. Properties and Operations 4.4.3. Applications 4.5. Binary Search Tree 4.5.1. Motivation with the task of Searching and Binary Search Algorithm 4.5.2. Properties of BST 4.5.3. Searching an element in BST 4.5.4. Insertion and Removal of Elements 4.5.5 Performing Rank and Range queries with BST 4.5.5. Performance 4.6 k-d Trees 4.6.1 Motivation (need of k-d Trees) 4.6.2 Data Representation in k-d Trees 4.6.3 Performing ranking, range queries 4.6.4 Performance	T1: 2.5, 3.1 R3: 10 T1: 12.3.2
M5	**Algorithm Design Techniques**	T1: 5.1, 5.2, 5.3, 4.1, 4.3,

	5.1. Greedy Method 5.1.1. Design Principles and Strategy 5.1.2. Fractional Knapsack Problem 5.1.3. Task Scheduling Problem 5.1.4. Minimum Spanning Tree 5.1.5. Shortest Path Problem - Djikstra's Algorithm 5.2. Divide and Conquer 5.2.1. Design Principles and Strategy 5.2.2. Analyzing Divide and Conquer Algorithms 5.2.3. Integer Multiplication Problem 5.2.4. Sorting Problem 5.2.4.1. Merge Sort Algorithm 5.2.4.2. Quick Sort Algorithm 5.2.5. Searching Problem and Binary Search Algorithm [Ref to 4.4.1] 5.3. Dynamic Programming 5.3.1.1. Design Principles and Strategy 5.3.1.2. Matrix Chain Product Problem 5.3.1.3. 0/1 Knapsack Problem 5.3.1.4. All-pairs Shortest Path Problem 5.4. Map Reduce Paradigm 5.4.1. Fork-Join 5.4.2. Map-Reduce 5.4.3. Applications	7.1, 7.2, 7.3 http://gee.cs.oswego.edu/dl/papers/fj.pdf https://dl.acm.org/citation.cfm?id=1327492 http://www.macs.hw.ac.uk/cs/techreps/docs/files/HW-MACS-TR-0096.pdf
M6	**Complexity Classes** 6.1. Definition of P and NP classes and examples 6.2. Understanding NP-Completeness 6.2.1. NP-Hardness 6.2.2. Polynomial time reducibility 6.2.3. Cook-Levin theorem 6.2.4. Problems in NP-Complete and using polynomial time reductions 6.2.4.1. CNF-SAT, 3-SAT 6.2.4.2. Vertex Cover 6.2.4.3. Clique and Set-Cover 6.2.4.4. Subset-Sum and Knapsack 6.2.4.5. Hamiltonian Cycle and TSP 6.3. Approximation Algorithms for harder problems 6.3.1. Solving vertex Cover Problem 6.4. Back Tracking for harder problems 6.4.1. Solving CNF-SAT Problem	T1: 13.1, 13.2, 13.3, 13.4, 13.5

Learning Outcomes:

No	Learning Outcomes
LO1	Describe various fundamental and advanced data structures, their properties, algorithm design techniques and various means of evaluating algorithms
L02	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

Part B: Contact Session Plan

Academic Term	
Course Title	DATA STRUCTURES AND ALGORITHMS DESIGN
Course No	
Lead Instructor	

Course Contents

Contact Hours(#) Or Contact Sessions(#)	List of Topic Title (from content structure in Course Handout)	Text/Ref Book/external resource
1	Algorithms and it's Specification, Random Access Machine Model, Counting Primitive Operations, Notion of best case, average case and worst case. Use of asymptotic notation, Big-Oh, Little-Oh, Omega and Theta Notations. Correctness of Algorithms.	T1: 1.1, 1.2
2	Analyzing Recursive Algorithms: Recurrence relations, Specifying runtime of recursive algorithms, Solving recurrence equations. Case Study: Analysing Algorithms Stack ADT and Implementation. Queues: Queue ADT and Implementation, Applications	T1: 1.4, 2.1
3	Amortized Analysis – Stack, Queue operations Lists- Notion of position in lists, List ADT and Implementation. Sets- Set ADT and Implementation	T1: 1.5, 2.2, 4.2, 2.3

	Trees: Terms and Definition, Tree ADT, Applications	
4	Binary Trees : Terms and Definition, Properties, Properties, Representations (Vector Based and Linked), Binary Tree traversal (In Order, Pre Order, Post Order), Applications Heaps - Definition and Properties, Representations (Vector Based and Linked), Insertion and deletion of elements	T1: 2.3, 2.4
5	Heap implementation of priority queue, Heap sort Graphs - Terms and Definitions, Properties, Representations (Edge List, Adjacency list, Adjacency Matrix), Graph Traversals (Depth First and Breadth First Search)	T1: 2.4, 6.1, 6.2, 6.3
6	Directed Graph and Reachability, Applications :- Page Rank Algorithm Dynamic Graphs - Concepts, applications	T1: 6.4, http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf , R3: 31
7	Unordered Dictionary :ADT, Applications Hash Tables: Notion of Hashing and Collision (with a simple vector based hash table)Hash Functions: Properties, Simple hash functions Methods for Collision Handling: Separate Chaining, Notion of Load Factor, Rehashing, Open Addressing [Linear; Quadratic Probing, Double Hash]	T1: 2.5
8	Ordered Dictionary: ADT, Applications Bloom Filters - Motivation, Properties and Operations Types of Bloom Filters, Applications	T1: 3.1 R3: 10
9	Binary Search Tree - Motivation with the task of Searching and Binary Search Algorithm, Properties of BST, Searching an element in BST Insertion and Removal of Elements, Rank and Range Queries, Performance k-d Trees, Representation, Region based and point based queries, Performance	T1: 3.1 T1: 12.3.2
10	Greedy Method - Design Principles and Strategy, Fractional Knapsack Problem, Task Scheduling Problem Minimum Spanning Tree, Shortest Path Problem - Djikstra's Algorithm	T1: 5.1, 7.1, 7.3

11	Divide and Conquer - Design Principles and Strategy, Analyzing Divide and Conquer Algorithms, Integer Multiplication Problem Sorting Problem - Merge Sort Algorithm, Quick Sort Algorithm. Searching Problem and Binary Search Algorithm [Ref to 4.4.1]	T1: 5.2, 4.1, 4.3
12	Dynamic Programming - Design Principles and Strategy, Matrix Chain Product Problem, 0/1 Knapsack Problem, All-pairs Shortest Path Problem	T1: 5.3, 7.2
13	Map Reduce Paradigm - Fork-Join, Map-Reduce, Comparison, Applications	http://gee.cs.oswego.edu/dl/papers/fj.pdf https://dl.acm.org/citation.cfm?id=1327492 http://www.macs.hw.ac.uk/cs/techreps/docs/files/HW-MACS-TR-0096.pdf
14	Definition of P and NP classes and examples, Understanding NP-Completeness: CNF-SAT Cook-Levin theorem Polynomial time reducibility: CNF-SAT and 3-SAT, Vertex Cover	T1: 13.1, 13.2, 13.3
15	Polynomial time reducibility: Clique and Set-Cover, Subset-Sum and Knapsack A brief overview on: Approximation Algorithms for harder problems - Solving vertex Cover Problem. Back Tracking for harder problems -Solving CNF-SAT Problem	T1: 13.3, 13.4
16	Review Session	

 # The above contact hours and topics can be adapted for non-specific and specific WILP programs depending on the requirements and class interests.

Lab Details

Title	Access URL
Lab Setup Instructions	To be Determined
Lab Capsules	To be Determined
Additional References	To be Determined

Select Topics and Case Studies from business for experiential learning

Topic No.	Select Topics in Syllabus for experiential learning	Access URL
1	Modules 3, 4, 5 and 6 of this handout has case studies focussing on (1) evaluation and selection of data structures (2) design, analysis and implementation of algorithms.	TBD

Evaluation Scheme

Legend: EC = Evaluation Component

No	Name	Type	Duration	Weight	Day, Date, Session, Time

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

Important Information

Syllabus for Mid-Semester Test (Closed Book): Topics in Weeks 1-7

Syllabus for Comprehensive Exam (Open Book): All topics given in plan of study

Evaluation Guidelines:

1. EC-1 consists of either two Assignments or three Quizzes. Announcements regarding the same will be made in a timely manner.
2. For Closed Book tests: No books or reference material of any kind will be permitted. Laptops/Mobiles of any kind are not allowed. Exchange of any material is not allowed.
3. For Open Book exams: Use of prescribed and reference text books, 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.
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. 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 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 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.