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
Version No	2.0
Date	01/10/2020

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 elementary techniques to analyze algorithms and characterize their running time and space as complexity measures
CO2	Introduce linear and non-linear data structures and criteria to select appropriate data structure(s) for a given application
CO3	Study - in detail – generic / classic dictionary data structures and their implementations (Lists, Binary Search Trees, and Hash Tables), along with select specialized data structures (such as Heaps, Bloom Filters, kd-trees, and Sketches).
CO4	Study - in detail - the elementary algorithm design approaches (Greedy, Divide-and-Conquer, and Dynamic Programming), their application in different contexts (Searching and Sorting, Graphs Optimization etc.) and their impact on efficiency
CO5	Introduce complexity classes, P, EXP, and NP, the notion of NP-Completeness along with ways of classifying problems (including reduction) and ways to handle NP-complete problems

Learning Outcomes:

No	Learning Outcomes
LO1	Define, design, and implement various elementary and advanced data structures,
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 and Linear and Non-Linear Data Structures
LO4	Explain with a practical example, each of the algorithm design strategies (greedy, divide-and-conquer, dynamic programming)
LO5	Translate real-world requirements to known data structures and algorithms with suitable implementation
LO6	Explain the significance of NP-completeness
LO7	Classify specific problems into complexity classes P and NP and prove hardness of problems

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	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

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 & Solving Recurrences by Substitution Method 1.3.3. Master Theorem	T1: 1.1, 1.2 T1:1.1.4 R1: 4.3,
M2	**Elementary Data Structures** 2.1. Stacks ADT and Queue ADT , Implementation and Applications 2.2. Amortized Analysis – Stack, Queue operations- Aggregate Method 2.3. 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	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. Probabilistic Look-up and cost involved, Bloom Filters, Applications 4.4. Binary Search Tree 4.5.1. BST Operations 4.5.2. Applications 4.5. Rank and Range Queries, Performance 4.6 Multi-dimensional data - Examples and Applications. k-d Trees, Representation, Region based and point based queries, Operations and Implementation. Analysis 4.7 External Memory Data Structures: Cost of disk access(es). IO Complexity. B-Tree. B+Trees, R-trees. Complexity Analysis	R2:11 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 - Dijkstra's Algorithm 5.1.5. Task Scheduling Problem [Assigned Reading] 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 [Assigned Reading] 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 [Assigned Reading] 5.3.5 Directed Graph and Reachability-Floyd-Warshall's Transitive Closure	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 complexity classes and example problems 6.2. Understanding NP-Completeness 6.2.1. Lower Bounding 6.2.2. Polynomial time reducibility and its transitivity 6.2.3 NP-Hardness and Cook-Levin theorem 6.2.4. Problems that are NP-Complete 6.2.5 Using polynomial time reductions to prove hardness Examples CNF-SAT, 3-SAT, Hamiltonian Cycle and TSP	T1: 13

	6.3. Approximation Algorithms for NP-hard problems Example: Vertex Cover Problem 6.4. BackTracking Example: CNF-SAT Problem	

Part B: Session Plan

Academic Term	**2020-2021 First Semester**
Course Title	**DATA STRUCTURES AND ALGORITHMS DESIGN**
Course No	**DSECLZG519**
Lead Instructor	**S.P.Vimal**

SESSION CONTENTS

Session (#)	**List of Topic Title (from content structure in Course Handout)**	**Text/Ref Book**
1	Introduction to the course - what to expect ? Algorithms - Specifying algorithms - Why analyzing algorithms - Specifying Algorithm - Pseudo Code - RAM Model - Counting primitive operations - best , average worst case - examples	R1: 1.1 T1: 1.1.1, 1.1.2, 1.1.3
2	Asymptotic Notations [big O, Big Omega, Big Theta] - Defining - Using - Comparing Algorithms - importance - Examples Example of a recursive program - Characterizing its run time using recurrence relation - solve it with substitution method - introduce masters method and show how its run time is arrived at - 3 examples of uses of masters method [Webinar #1] Exercises on Algorithm Analysis	T1:
3	**Elementary Data Structures** Stacks ADT and Queue ADT , Implementation and Applications List ADT , Implementation and Applications [Webinar #2] Amortized Analysis – Stack, Queue operations- Aggregate Method	R1: 10.1, 10.2, 10.3, T1: 2.2 Webinar Ref: R1: Ch 16 Link [for ref]
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 **Heaps :** Definition and Properties	T1: 2.3
5	**Heaps**	

	● Representations (Array Based and Linked) ● Insertion and deletion of elements ● Heap sort ● Priority Queue	T1: 2.4 R1: 6.5
6	**Graphs** ● Terms and Definitions ● Properties ● Representations (Edge List, Adjacency list, Adjacency Matrix) ● Graph Traversals (Depth First and Breadth First Search) ● Applications	T1: 6.1, 6.2, 6.3
7	**Dictionaries** ● Dictionary ADT , Applications ● Hash Tables ● Notion of Hashing and Collision **Methods for Collision Handling** ● Separate Chaining ● Notion of Load Factor ● Rehashing ● Open Addressing [Linear & Quadratic Probing, Double Hash] ● Applications	T1: 2.5.1, 2.5.2,2.5.3, 2.5.4, 2.5.5
8	Probabilistic Look-up and cost involved, Bloom Filters, Applications [Ref. Lecture Materials] [Webinar #3] Binary Search Tree[BST Operations, Applications] Rank and Range Queries, Performance [Ref. Lecture Materials]	T1: 3.1
9	Multi-dimensional data - Examples and Applications. k-d Trees, Representation, Region based and point based queries, Operations and Implementation. Analysis [Ref. Lecture Materials] R-trees [Ref. Lecture Materials]	
10 [1]	External Memory Data Structures: Cost of disk access(es). IO Complexity. B-Tree. B+Trees, Complexity Analysis	T1:14.1.2
10 [2]	Greedy Method Design Principles and Strategy, Fractional Knapsack Problem,	T1: 5.1 , 7.3
11.	Minimum Spanning Tree Shortest Path Problem , Dijkstra's Algorithm Task Scheduling Problem	T1: 7.3 ,7.1.1
12	Divide and Conquer: Merge Sort Integer Multiplication Problem Quicksort	T1: 4.1.1, 4.1.2 T1: 5.2.2
13	Dynamic Programming: Design Principles and Strategy Matrix Chain Product Problem [Webinar #4] Exercises on Algorithm Design Techniques	T1: 5.3.1, 5.2.2,

14	Dynamic Programming: All-pairs Shortest Path Problem 0/1 Knapsack Problem Directed Graph and Reachability-Floyd-Warshall's Transitive Closure	T1: 4.1,7.2, 5.3.3
15-16	**Complexity Classes** Definition of P and NP complexity classes and example problems Understanding NP-Completeness [Lower Bounding, Polynomial time reducibility and its transitivity, NP-Hardness and Cook-Levin theorem, Problems that are NP-Complete, Using polynomial time reductions to prove hardness, Examples CNF-SAT, 3-SAT, Hamiltonian Cycle and TSP] Approximation Algorithms for NP-hard problems - Example: Vertex Cover Problem BackTracking - Example: CNF-SAT Problem	T1: Ch 13

TUTORIAL SESSION CONTENTS

# There should be 4 tutorial sessions planned ,each with a duration of 1.5 hours.

Webinar(#)	Topic
1	Algorithm Analysis - Cases - Examples
2	Amortized Analysis – Stack, Queue operations- Aggregate Method - External
3	Blooms Filters and Applications in the Data Science Context.
4	Algorithm Design Exercises

Select Topics and Case Studies from business for experiential learning

Topic No.	Select Topics in Syllabus for experiential learning	Access URL
TBD	TBD	TBD

Select Topics and Case Studies from business for experiential learning

Topic No.	Select Topics in Syllabus for experiential learning	Access URL
TBD	TBD	TBD

Evaluation Scheme

Legend: EC = Evaluation Component

No	Name	Type	Duration	Weight	Day, Date, Session, Time
EC-1	Assignment-1 [12%]	TBA through Canvas Notice		30%	TBA through Canvas Notice
	Assignment-2 [13%]				
	Quiz [05%]				
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 1-7
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.