ITA5002	**Problem Solving with Data Structures and Algorithms**	**L**	**T**	**P**	**J**	**C**
		3	**0**	**2**	**0**	**4**
Pre-requisite	**Nil**	**Syllabus version**				
		v.1.0				

Course Objectives:

1. Familiarize with basic techniques of algorithm analysis and master the implementation of linked data structures.
2. Familiarize with several sub-quadratic sorting algorithms.
3. Familiarize with graph algorithms

Expected Course Outcomes:

1. Calculate to find the time and space complexities of various algorithms.
2. Choose appropriate data structure as applied to specified problem definition.
3. Handle operations like searching, insertion, deletion and traversing mechanism on various data structures.
4. Use linear and non-linear data structures.
5. Solve problems using data structures.
6. Apply concepts learned in various domains

Student Learning Outcomes (SLO): **2, 9, 20**

Module:1	**Introduction to algorithm analysis**	**4 hours**

The Problem-solving Aspect, Analysis framework, Asymptotic notations, Growth rate of functions, Complexity analysis, Mathematical analysis of recursive and non-recursive algorithms.

Module:2	**Fundamental Data Structures – List, Stacks and Queues**	**7 hours**

List ADT, Singly linked lists, Doubly Linked lists and Circular Linked Lists – Stack ADT, Implementation of Stacks and applications. Queue ADT, Implementation of Queue and applications

Module:3	**Trees**	**7 hours**

Tree ADT, Binary tree, Search Tree ADT, Tree Traversals, AVL tree, Splay tree

Module:4	**Sorting and Searching**	**6 hours**

Insertion Sort, Selection, heap sort and Merge sort. Linear time sorting – bucket and radix sort. Linear search and binary search.

Module:5	**Graph algorithms**	**7 hours**

The Graph ADT, Representation of adjacency list and matrix, Graph traversals – Depth First Search and Breadth First Search implementation. Shortest path – weighted graphs – Dijkstra’s algorithm. Minimum spanning tee – Prim’s and Kruskal’s algorithm.

Module:6	**Algorithm Design Techniques**	**7 hours**

Greedy algorithms – Simple scheduling algorithms, Huffman code, Divide and Conquer – Running time of divide and conquer technique, Closest point problem and Selection problem. Backtracking technique

Module:7	**Dynamic Programming**	**5 hours**
Using a table Instead of recursion, Ordering matrix multiplication, Optimal binary search tree and All Pairs Shortest path.		
Module:8	**Contemporary issues:**	**2 hours**
Expert talk		
	Total Lecture hours:	**45 hours**
Text Book(s)		
1.	Mark Allen Weiss, Data Structure and Algorithm Analysis in C++, 2014, 4th Edition, Pearson Education Limited.	
Reference Books		
1.	AnanyLevitin, Introduction to design and analysis of algorithm, 2012, 3rd Edition, Addison – Wesley.	
2	Thomas H. Cormen, C.E. Leiserson, R L.Rivest and C. Stein, Introduction to Algorithms, Paper Back, 2010, 3rd Edition, MIT Press.	
List of Challenging Experiments (Indicative)		
1.	Write a program to implement a 3-stacks of size ‘m’ in an array of size ‘n’ with all the basic operations such as IsEmpty(i), Push(i), Pop(i), IsFull(i) where ‘i’ denotes the stack number (1,2,3), m n/3. Stacks are not overlapping each other. Leftmost stack facing the left direction and other two stacks are facing in the right direction.	2 hours
2.	Students of a Programming class arrive to submit assignments. Their register numbers are stored in a LIFO list in the order in which the assignments are submitted. Write a program using array to display the register number of the ten students who submitted first. Register number of the ten students who submitted first will be at the bottom of the LIFO list. Hence pop out the required number of elements from the top so as to retrieve and display the first 10 students.	2 hours
3.	To facilitate a thorough net surfing, any web browser has back and forward buttons that allow the user to move backward and forward through a series of web pages. To allow the user to move both forward and backward two stacks are employed. When the user presses the back button, the link to the current web page is stored on a separate stack for the forward button. As the user moves backward through a series of previous pages, the link to each page is moved in turn from the back to the forward stack. When the user presses the forward button, the action is the reverse of the back button. Now the item from the forward stack is popped, and becomes the current web page. The previous web page is pushed on the back stack. Simulate the functioning of these buttons using array implementation of Stack. Also provide options for displaying the contents of both the stacks	2 hours

	whenever required.	
4.	Most of the bugs in scientific and engineering applications are due to improper usage of precedence order in arithmetic expressions. Thus it is necessary to use an appropriate notation that would evaluate the expression without taking into account the precedence order and parenthesis. a) Write a program to convert the given arithmetic expression into i) Reverse Polish notational ii) Polish notation	2 hours
5.	In a theme park, the Roller-Coaster ride is started only when a good number of riders line up in the counter (say 20 members). When the ride proceeds with these 20 members, a new set of riders will line up in the counter. This keeps continuing. Implement the above scenario of lining up and processing using arrays with Queue ADT.	2 hours
6.	When burning a DVD it is essential that the laser beam burning pits onto the surface is constantly fed with data, otherwise the DVD fails. Most leading DVD burn applications make use of a circular buffer to stream data from the hard disk onto the DVD. The first part, the 'writing process' fills up a circular buffer with data, then the 'burning process' begins to read from the buffer as the laser beam burns pits onto the surface of the DVD. If the buffer starts to become empty, the application should continue filling up the emptied space in the buffer with new data from the disk. Implement this scenario using Circular Queue.	2 hours
7.	**Assume FLAMES** game that tests for relationship has to be implemented using a dynamic structure. The letters in the FLAMES stand for Friends, Love, Affection, Marriage, Enmity and Sister. Initially store the individual letters of the word 'flames' in the nodes of the dynamic structure. Given the count of the number of uncommon letters in the two names 'n', write a program to delete every nth node in it, till it is left with a single node. If the end of the dynamic structure is reached while counting, resume the counting from the beginning. Display the letter that still remains and the corresponding relationship.	2 hours
8.	Assume in the Regional Passport Office, a multitude of applicants arrive each day for passport renewal. A list is maintained in the database to store the renewed passports arranged in the increased order of passport ID. The list already would contain their cords renewed till the previous day. Apply Insertion sorting technique to place the current day's records in the list. Later the office personnel wish to sorting the records based on the date of renewal so as to know the count of renewals done each day. Taking into consideration the fact that each record has several fields (around 25 fields), follow Selection sorting logic to implement the same.	2 hours
9.	Write a program to implement Bubble sort, Heap sort and Quick sorting techniques to arrange the following sequence of elements in descending order. 9, -4, 5, 8, -3, 7, 0, 4, 1, 2. Display the count of number of	2 hours

	comparisons and swaps made in each method. Apply the same sorting techniques for sorting a large data set [Randomly generate 5000 integers within the range -5 0000 to 50000 to build the data set. From your observation and analysis, determine the best sorting technique for working with large numbers.	
10.	Write a program to implement Radix Sort on 1D array of Faculty structure (contains faculty name, faculty_ ID, subject_ codes, class_ names), with key as faculty_ ID. And count the number of swap performed.	2 hours
11.	Given a text file T, write a program that will output the longest sentence in the text file.	2 hours
12.	Write a program to implement Binary search on 1D array of Employee structure (contains employee_ name, emp_ no, emp_ salary), with key as emp_ no. And count the number of comparison happened.	2 hours
13.	Write a program for Binary Search Tree to implement following operations: a. Insertion b. Deletion i. Delete node with only child ii. Delete node with both children c. Finding an element d. Finding Min element e. Finding Max element f. Left child of the given node g. Right child of the given node h. Finding the number of nodes, leaves nodes, full nodes, ancestors, descendants.	2 hours
14.	Write a program for AVL Tree to implement the insertion operations: (For nodes as integers). Test the program for all cases (LL, RR, RL, LR rotation)	2 hours
15.	Write a program to match the string PATTERN for the given string TEXT and return the index of the leftmost character of the PATTERN if its exists in the string TEXT and return -1 otherwise.	
16.	Given a graph G = (V, E) and |V| = n and |E| = m, where V is the set of vertices and E is the set of edges. Write a program that will output the parent nodes of each nodes in each of the following traversal mechanisms: a. Depth First Traversal, b. Breadth First Traversal.	2 hours
17.	Let $G = (V, E)$ be a given graph with $\|V\| = n$and $\|E\| = m$, where V is the set of vertices and E is the set of edges. Write a program to find the shortest path in G, given a source node s and destination node t.	
	Total Laboratory Hours	30 hours
Mode of Evaluation: CAT / Assignment / Quiz / FAT / Project / Seminar		

Recommended by Board of Studies	05-03-2016		
Approved by Academic Council	40th	Date	18-03-2016