SRM VALLIAMMAI ENGINEERING COLLEGE

(An Autonomous Institution)

SRM Nagar, Kattankulathur – 603 203

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

QUESTION BANK

III SEMESTER

1922302 – PROGRAMMING AND DATA STRUCTURES

Regulation – 2019

Academic Year 2022 – 2023 (ODD SEMESTER)

Prepared by

Ms.R.LAKSHMI, Assistant Professor (Sel. G) / AI & DS

SRM VALLIAMMAI ENGINEERING COLLEGE

(An Autonomous Institution)

SRM Nagar, Kattankulathur – 603 203.

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

QUESTION BANK

SUBJECT : 1922302 – PROGRAMMING AND DATA STRUCTURES

SEM / YEAR : III Sem / II Year

UNIT I - OBJECT ORIENTED APPROACHES OF PYTHON			
Basic concepts of Object Oriented Programming – From Procedural to Object approach –Properties - Classes – Methods – Attributes - Objects – Inheritance – Overloading –Overriding – Data Hiding - Exceptions – Processing Files			
PART – A			
1	What Is Object-Oriented Programming?	BTL 2	Understanding
2	Differentiate Object-Oriented Programming (OOP) vs Procedure Oriented Programming (POP)	BTL 4	Analyzing
3	What is a class?	BTL 1	Remembering
4	Define Objects and object instantiation	BTL 1	Remembering
5	What is Class constructor	BTL 2	Understanding
6	How many statements are inside the constructor method	BTL 2	Understanding
7	Explain class methods with example	BTL 4	Analyzing
8	How do you create more than one object of a class	BTL 2	Understanding
9	Create a class to pass the wrong number of arguments	BTL 1	Remembering
10	Define inheritance in python class	BTL 3	Applying
11	Define encapsulation	BTL 6	Creating
12	What is the difference between multiple and multilevel inheritances	BTL 5	Evaluating
13	What is meant by data abstraction	BTL 6	Applying
14	How do you create an object and class using python	BTL 1	Remembering
15	Is Python 100 percent object oriented?	BTL 3	Creating
16	List the functions used to check a relationships of two classes and instances	BTL 1	Remembering
17	Can we override parent class methods if so state the reason.	BTL 1	Remembering
18	Does object's attributes will be visible outside the class definition.	BTL 4	Analyzing
19	What is Python Exception?	BTL 3	Applying
20	Explain Built-In Class Attributes	BTL 5	Evaluating

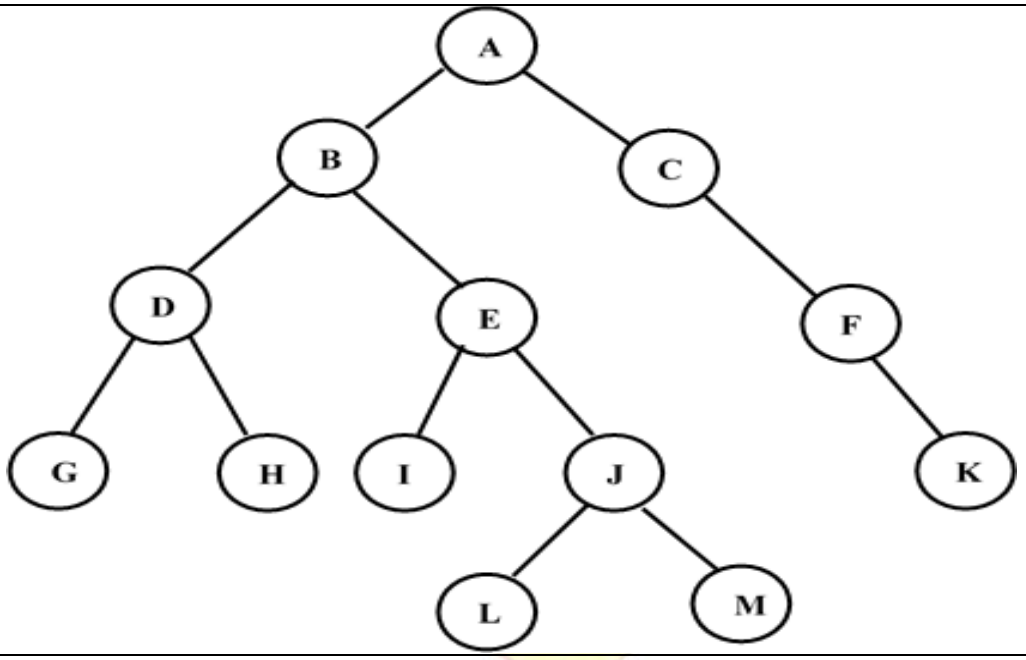
21	Identify the types of arguments in python	BTL 2	Understanding
22	Summarize some built in modules in python	BTL 5	Evaluating
23	Point out the operators which python support.	BTL 4	Analyzing
24	Discover the difference between intermediate mode and script mode?	BTL 3	Applying
PART – B			
1	List the conditions must be met for overriding a function with example.(13)	BTL 5	Evaluating
2	Explain in detail about OOP Terminology.(13)	BTL 1	Remembering
3	How do you access the object attributes explain with example.(13)	BTL 1	Remembering
4	What is Inheritance in Python? List the Types of Inheritance in Python.(13)	BTL 1	Remembering
5	Explain how A subclass may change the functionality of a Python method in the superclass.(13)	BTL 2	Understanding
6	How do you Declare a Python Constructor explain with example.(13)	BTL2	Understanding
7	(i)How many methods are there in Python? Explain with example.(7) (ii)How a function differs from a method.(6)	BTL 4	Analyzing
8	How do you call a method from one class to another in Python?(13)	BTL 2	Understanding
9	What are objects in Python? Explain with example.(13)	BTL 6	Creating
10	(i)What is file i/o in Python?(7) (ii)How to use file i/o in Python?(6)	BTL 3	Applying
11	(i)How do you write output to a file in Python?(7) (ii)Why do we need files in Python?(6)	BTL 1	Remembering
12	What is Python exception handling? Explain with example.(13)	BTL 4	Analyzing
13	Does Python support Access control? Justify(13)	BTL 4	Analyzing
14	What is the purpose of try and catch in Python exception handling?(13)	BTL 2	Understanding
15	Interpret the various features of the Object Oriented Programming Language.(13)	BTL 3	Applying
16	Summarize the key elements of Object Oriented Programming and briefly explain it.(13)	BTL 5	Evaluating
17	Illustrate with the help of a program how object oriented programming overcomes the shortcomings of procedure oriented programming.(13)	BTL 3	Applying
PART C			
1	(i)Why exception handling is important in Python?(10) (ii)How to handle multiple exceptions in Python.(5)	BTL 5	Evaluating
2	(i)How does Python handle value exception?(10) (ii)Describe In-built Python Exception.(5)	BTL 6	Creating
3	Write a module called Circle which contains circle class with data attributes such as radius and method getarea(). Explain with sample data.(15)	BTL 6	Creating
4	Create a program to define point class which models 2D point with x and y coordinates using overload operator '+' and '*' by overriding the method _add_() and _mul_().(15)	BTL 5	Analyzing
5	Given two one dimensional arrays A and B which are sorted in ascending order. Recommend Object Oriented and Python program to merge them into a single sorted array, see that is contains every item from array A and B, in ascending order.(15)	BTL 5	Evaluating

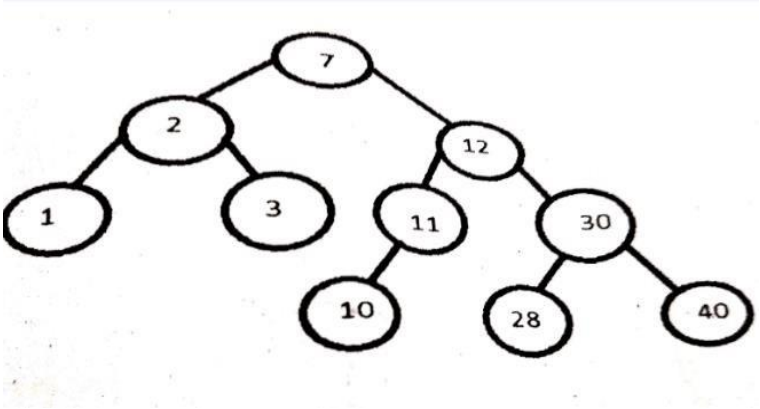
UNIT II - LINEAR DATA STRUCTURES			
Abstract Data Types (ADTs) – List ADT – array-based implementation – linked list implementation - singly linked lists- circularly linked lists- doubly-linked lists - Stack ADT –Operations - Applications - Evaluating arithmetic expressions - Queue ADT – Operations -Circular Queue.			
PART – A			
Q.No	**Questions**	**BT Level**	**Competence**
1	Define data structure with example.	BTL1	Remembering
2	Define ADT. Give any two examples.	BTL1	Remembering
3	Define list. Mention any two operations that are performed on a list.	BTL1	Remembering
4	Point out the basic operations carried out in a linked list.	BTL4	Analyzing
5	Compare calloc() and realloc() function and mention its application in linked list.	BTL4	Analyzing
6	Develop an algorithm for insertion operation in a singly linked list.	BTL6	Creating
7	Examine a doubly linked list with neat diagram.	BTL3	Applying
8	What is circular linked list?	BTL1	Remembering
9	Indicate the advantage of circular linked list.	BTL2	Understanding
10	Design a routine to delete an element in a linked list.	BTL6	Creating
11	Define stack and specify the operations	BTL1	Remembering
12	Identify any four applications of stack?	BTL2	Understanding
13	Illustrate the purpose of top and pop?	BTL3	Applying
14	What causes underflow of stack? How it could be avoided?	BTL4	Analyzing
15	How to implement stack using linked list.	BTL4	Analyzing
16	Recommend a routine to return top element in a stack using linked list.	BTL 5	Evaluating
17	What is circular queue?	BTL1	Remembering
18	Circular queue is better than standard linear queue, Why?	BTL2	Understanding
19	Classify the different types of queues.	BTL3	Applying
20	Show a routine to perform enqueue operation in a queue.	BTL3	Applying
21	Differentiate between linear linked list and Circular linked list.	BTL2	Understanding
22	Assess and write the linked list representation of a polynomial: p(x)=4x3 +6x2 +7x+9	BTL 5	Evaluating
23	Given the prefix for an expression. Estimate its postfix: ++A*BCD and +*AB*CD	BTL2	Understanding
24	Given a infix expression convert it into postfix expression using stack a+b*(c^d-e)^(f+g*h)-i	BTL 5	Evaluating

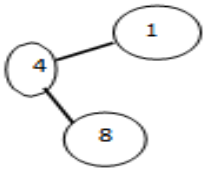
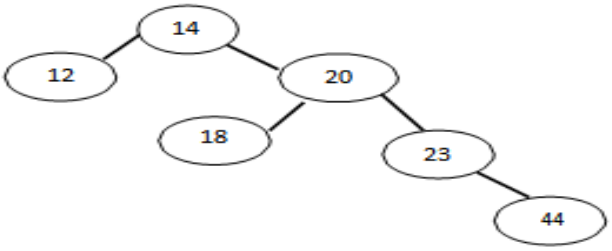
PART – B			
1	Identify the array implementation of list and show all its operations.(13)	BTL1	Remembering
2	What is a linked list? Describe the suitable routine segments for any four operations.(13)	BTL1	Remembering
3	Discuss the creation of a doubly linked list and write routine to insert an element in doubly linked list and delete an element in doubly linked list.(13)	BTL2	Understanding
4	Analyze and write procedure for circular linked list with create, insert, delete, display operations.(13)	BTL4	Analyzing
5	Analyze the doubly linked list and circular linked list. Mention its advantages and disadvantages.(13)	BTL4	Analyzing
6	Explain the steps involved in insertion and deletion into a singly linked list.(13)	BTL5	Evaluating
7	Develop a C program for linked list implementation of list.(13)	BTL6	Creating
8	Discuss about stack ADT using array in detail.(13)	BTL2	Understanding
9	(i)Identify an algorithm for push and pop operations on stack using a linked list with an example.(8) (ii)Explain the function to examine whether the stack is full() or empty().(5)	BTL2	Understanding
10	Describe about implementation of queue ADT using linked list with suitable example.(13)	BTL1	Remembering
11	Briefly describe the operations of queue with example.(13)	BTL1	Remembering
12	Explain how to evaluate the following arithmetic expressions using stacks. i) 6 2 3 + - 3 8 2 / + * 2 ^ 3 + (5) ii) 6 2 / 3 - 4 2 * + (4) iii) 4 3* 6 7 + 5 - + (4)	BTL5	Evaluating
13	Discuss and write a C program to implement queue functions using arrays.(13)	BTL2	Understanding
14	What are circular queues. Explain the procedure to insert an element to circular queue and delete an element from a circular queue using array implementation.(13)	BTL4	Analyzing
15	Demonstrate the application of linked list in detail. (13).	BTL3	Applying
16	Consider an array A[1: n] Given a position, write an algorithm to insert an element in the Array. If the position is empty, the element is inserted easily. If the position is already occupied the element should be inserted with the minimum number of shifts.(Note: The elements can shift to the left or to the right to make the minimum number of moves). (13)	BTL3	Applying
17	(i)Illustrate the procedure to convert the infix expression to postfix expression and steps involved in evaluating the postfix expression. (7) (ii)Convert the expression A-(B/C+(D%E*F)/G)*H to postfix form. Evaluate the given postfix expression 9 3 4 * 8 + 4/-. (6)	BTL3	Applying
PART C			
1	Compose an algorithm to i. Reverse the elements of a single linked list. (5) ii. count the number of nodes in a given singly linked list. (5) iii. Searching the element from linked list. (5)	BTL6	Creating
2	Recommend a program to merge two sorted linked list (P &Q-assume that they are available) to get a single sorted list S. Eg. P:1→2→45→56 Q:3→24→56→63→66 (10) ii) Choose and write a non-recursive procedure to reverse a singly linked list. (5)	BTL 5	Evaluating

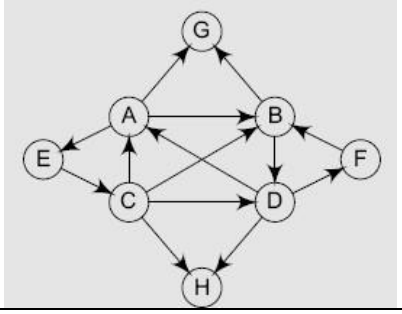
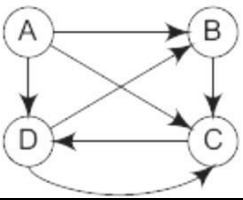
3	Develop a C program for linked list implementation of stack. (15)	BTL6	Creating
4	Generalize and develop a function to insert an element into a queue and delete an element from a queue, in which the queue is implemented as a linked list.(15)	BTL6	Creating
5	Convert the following infix expression into postfix form: i) (A-B)+C*D/E-C (5) ii)(A*B) +(C/D)-(D+E) (5) iii)14/7*3-4+9/2 (5)	BTL 5	Evaluating

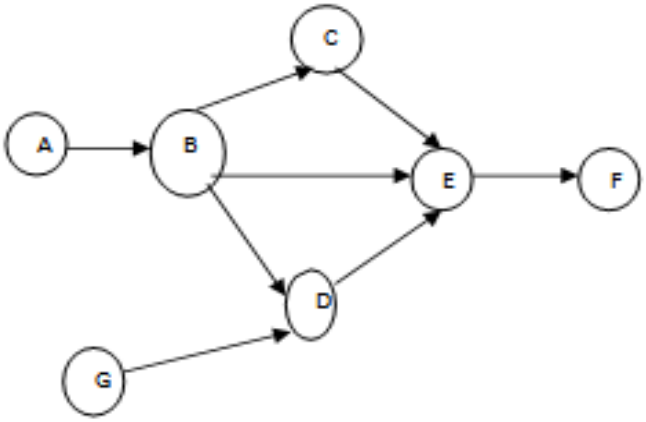
UNIT III – NON LINEAR DATA STRUCTURES – TREES			
Tree ADT – tree traversals - Binary Tree ADT – expression trees – applications of trees –binary search tree ADT –Threaded Binary Trees- AVL Trees – B-Tree - B+ Tree - Heap –Applications of heap			
PART – A			
Q.No	**Questions**	**BT Level**	**Competence**
1	If the depth of binary tree is k, the maximum number of nodes in the binary tree is 2^k-1. Prove.	BTL 5	Evaluating
2	Recommend the result of inserting 3, 1, 4, 6, 9, 2, 5, 7 into an initially empty binary search tree.	BTL 5	Evaluating
3	Define a binary tree. Give an example.	BTL 1	Remembering
4	Create an expression tree for the expression. ((a + ((b/c)*d))- e)	BTL 6	Creating
5	Differentiate AVL tree and Binary search tree.	BTL 4	Analyzing
6	Give the various types of rotations in AVL tree during the insertion of a node?	BTL 2	Understanding
7	What are threaded binary trees? Give its advantages.	BTL 1	Remembering
8	Define balance factor of AVL Tree.	BTL 1	Remembering
9	Simulate the result of inserting 2, 1, 4, 5, 9, 3, 6, 7 into an initially empty AVL Tree.	BTL 6	Creating
10	Define an expression tree. Give an example for it.	BTL 2	Understanding
11	Summarize tree traversal and mention the type of traversals?	BTL 2	Understanding
12	Differentiate B tree and B+ tree	BTL 2	Understanding
13	For the given tree i. List the siblings for node E ii. Compute the height.	BTL 1	Remembering

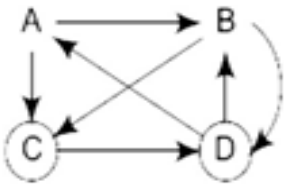
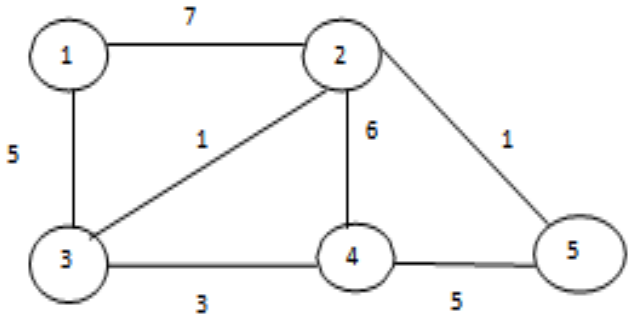
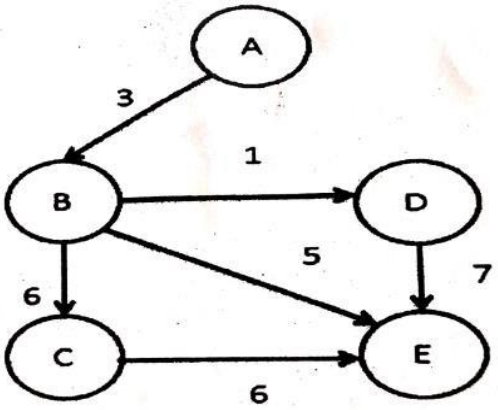
			
14	Point out the properties of B+ tree.	BTL 4	Analyzing
15	Illustrate the benefits of B+ tree.	BTL 3	Applying
16	List out the various operations that can be performed on B-trees	BTL 1	Remembering
17	Identify the structural properties of B-Trees.	BTL 1	Remembering
18	Illustrate the steps in the construction of a heap of records with the following key values: 12, 33, 67, 8, 7, 80, 5, 23.	BTL 3	Applying
19	Analyze the properties of binary heap.	BTL 4	Analyzing
20	Define a heap and show how it can be used to represent a priority queue.	BTL 3	Applying
21	Interpret the meaning of Depth and height of a tree.	BTL 3	Applying
22	Analyze and Convert the following into an expression tree: (a+b)*(c/d)-e/f.	BTL 4	Analyzing
23	Predict the advantages of threaded binary trees over ordinary binary trees?	BTL 2	Understanding
24	Summarize the merit and demerit of linked list representation of binary trees.	BTL 5	Evaluating
	PART – B		
1	Write an algorithm for preorder, inorder and postorder traversal of a binary tree. (13)	BTL 1	Remembering
2	Explain the following operations on a binary search tree with suitable algorithm i) Find a node(7) ii) Find minimum and maximum elements of BST.(6)	BTL 4	Analyzing
3	Write an algorithm for inserting and deleting a node in a binary search tree.(13)	BTL 1	Remembering
4	Describe the concept of threaded binary tree with example.(13)	BTL 1	Remembering
5	Discuss in detail the various methods in which a binary tree can be represented. Discuss the advantage and disadvantage of each method.(13)	BTL 2	Understanding
6	i) Create a binary search tree using the following data elements 45, 39, 56, 12, 34, 78, 32, 10, 89, 54, 67, 81 (7) ii) Explain the steps to convert general tree to binary tree? (6)	BTL 5	Evaluating

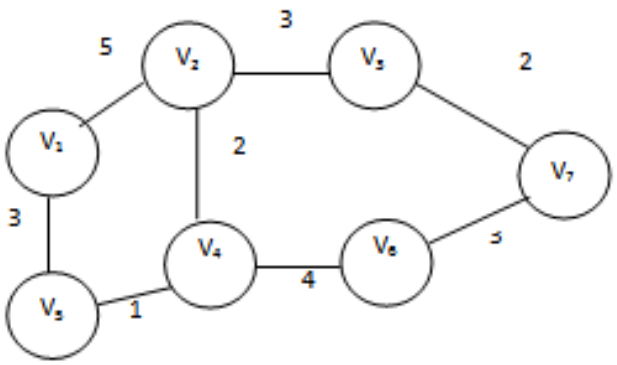
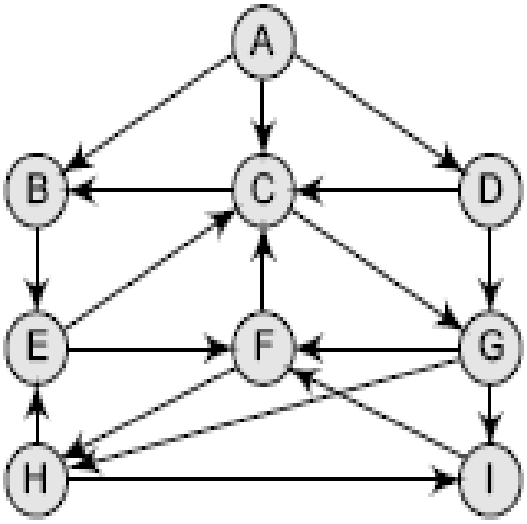
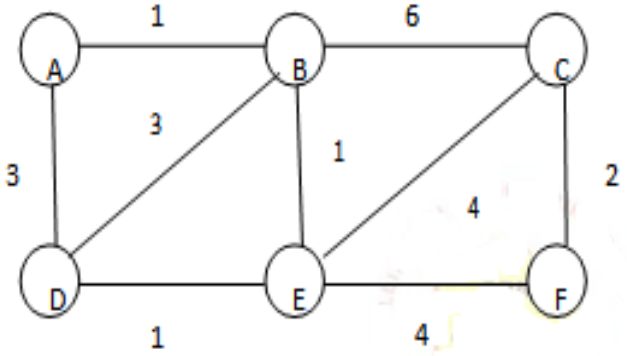
7	i) Construct B Tree of order m=5 for the following keys 1, 12, 8, 2, 25, 5, 14, 28, 17, 7, 52, 16, 48, 68, 3, 26, 29, 53, 55, 45 (8) ii) Delete the keys 8 and 55.State the rules for deletion. (5)	BTL 6	Creating
8	i) Discuss how to insert an element in a AVL tree and explain with algorithm.(7) ii) Explain how deletion can take place in AVL trees with suitable algorithm.(6)	BTL 2	Understanding
9	i)What are AVL trees? Describe the different rotations defined for AVL tree.(7) ii) Insert the following elements step by step in sequence into an empty AVL tree 63, 9, 19, 27, 18, 108, 99, 81. (6)	BTL 1	Remembering
10	Analyze the operations of B-tree using 2-3 tree with example.(13)	BTL 4	Analyzing
11	Discuss about B+ tree in brief with suitable example.(13)	BTL 2	Understanding
12	Explain the construction of expression tree with example. Give the applications of trees.(13)	BTL 4	Analyzing
13	Illustrate the construction of binomial heaps and its operations with a suitable example. (13)	BTL 3	Applying
14	i) Illustrate how the delete operation is performed on binary heap? (7) ii) Write suitable operations for percolate up and percolate down operations in a binary heap. (6)	BTL 3	Applying
15	Define Tree. Explain the tree traversals with algorithms and examples.(13)	BTL 2	Understanding
16	Construct an expression tree for the expression (a + b * c) +((d * e + 1) * g). Evaluate and Give the outputs when you apply preorder, inorder and postorder traversals.(13)	BTL 5	Evaluating
17	Interprert the algorithms used to perform single and double rotation on AVL tree.(13)	BTL 3	Applying
	PART C		
1	i) Construct and evaluate a B tree of order 5 by inserting the following: 3, 14, 7, 1, 8, 5, 11, 17, 13, 6, 23, 12, 20, 26, 4, 16, 18, 24, 25 and 19. (10) ii) Compare and assess B Tree and B+ Tree. (5)	BTL 5	Evaluating
2	i)Develop a routine for post order traversal. Is it possible to find minimum and maximum value in the binary search tree using traversals? Discuss. (5) ii)Display the given tree using Inorder, Preorder, Postorder traversals (6)  iii)Delete 11 and 10 from the above binary search tree and display the tree after each deletion. (4)	BTL 6	Creating
3	i) Draw B-Tree of order 5 for the keys {K, O, S, V, M, F, B, G, T, U, W} (5) ii) Delete the keys K and G in order. (5) iii) Justify the number of splits needed for inserts / delete with proper reasons. (5)	BTL 5	Evaluating
4	**Construct** AVL tree for the followings after rotation. (5+5+5)	BTL 6	Creating

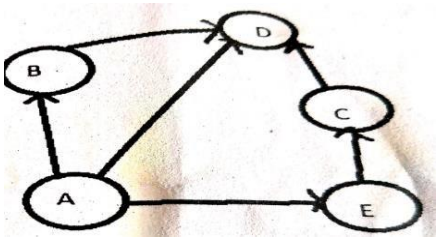
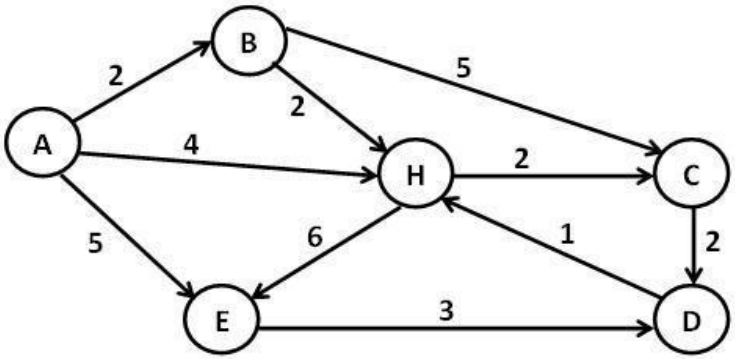
	i.  ii.  iii.  		
5	Create a binary search tree for the following numbers start from an empty binary search tree. 45,26,10,60,70,30,40 Delete keys 10,60 and 45 one after the other and show the trees at each stage.(15)	BTL 6	Creating

UNIT IV - NON LINEAR DATA STRUCTURES – GRAPHS			
Definition – Representation of Graph – Types of graph - Breadth-first traversal - Depth-first traversal – Topological Sort – Bi-connectivity – Cut vertex – Euler circuits – Applications of graphs			
PART – A			
Q.No	**Questions**	**BT Level**	**Competence**
1	Define graph.	BTL 1	Remembering
2	Consider the graph given below. Create the adjacency matrix of it  	BTL 6	Creating
3	Find out the in-degree and out-degree of each node in the given graph  	BTL 3	Applying
4	Create an undirected graph and its adjacency matrix for the following specification of a graph G. V(G)=1,2,3,4 E(G) ={ (1,2),(1,3),(3,3),3,4),(4,1)}	BTL 6	Creating
5	Differentiate BFS and DFS.	BTL 2	Understanding
6	What is meant by bi-connected graph?	BTL 1	Remembering
7	Give the purpose of Dijikstra's algorithm.	BTL 2	Understanding
8	Differentiate cyclic and acyclic graph	BTL 4	Analyzing

9	Classify strongly connected and weakly connected graph.	BTL 3	Applying
10	Illustrate an articulation point with example.	BTL 3	Applying
11	What are the representations of the graph?	BTL 1	Remembering
12	Define minimum spanning tree. Give an example	BTL 2	Understanding
13	State the principle of Topological sorting.	BTL 1	Remembering
14	Explain procedure for Depth first search algorithm.	BTL 4	Analyzing
15	Analyze the different ways of representing a graph.	BTL 4	Analyzing
16	Prove that the number of edges in a complete graph of n vertices in n(n-1)/2	BTL 5	Evaluating
17	Assess the minimum number spanning tree possible for a complete graph with n vertices.	BTL 5	Evaluating
18	What are Euler circuits?	BTL 1	Remembering
19	Give two applications of graphs.	BTL 2	Understanding
20	What is residual graph?	BTL 1	Remembering
21	Interpret the meaning of visiting and traversing in graph	BTL 2	Understanding
22	Illustrate about simple graph and weighted graph.	BTL 3	Applying
23	Assess the use of BFS?	BTL 5	Evaluating
24	Analyze the BFS algorithm.	BTL 4	Analyzing
	PART – B		
1	Describe in detail about the following representations of a graph. i) Adjacency Matrix (7) ii) Adjacency List (6)	BTL 1	Remembering
2	i) Consider the given directed acyclic graph D. Sort the nodes D by applying topological sort on 'D'. (7)  ii. Consider the graph given below and show its adjacency list in the memory. (6)	BTL 3	Applying

			
3	Examine topological sorting of a graph G with suitable example. (13)	BTL 1	Remembering
4	Differentiate depth-first search and breadth-first search traversal of a graph with suitable examples. (13)	BTL 4	Analyzing
5	i) Explain with algorithm, How DFS be performed on a undirected graph.(7) ii) Show the algorithm for finding connected components of an undirected graph using DFS and derive the time complexity of the algorithm. (6)	BTL4	Analyzing
6	i) Discuss an algorithm for Breadth first Search on a graph.(7) ii) Give an example based on the algorithm. (6)	BTL 2	Understanding
7	i) Illustrate Kruskal's algorithm to find the minimum spanning tree of a graph. (7) ii) Trace the algorithm for the following graph. (6) 	BTL 3	Applying
8	Compare any two applications of Graph with your own example.(13)	BTL 1	Remembering
9	Discuss the prim's algorithm for minimum spanning tree. Give an example.(13)	BTL 2	Understanding
10	Describe an appropriate algorithm to find the shortest path from 'A' to every other node of A for the given graph. (13) 	BTL 1	Remembering
11	i) Write a program to find an Euler circuit in a graph.(7) ii)Trace the algorithm for the given graph.(6)	BTL 5	Evaluating

			
12	Develop an algorithm to compute the shortest path using Dijkstra's algorithm. Validate the algorithm with suitable example. (13)	BTL 6	Creating
13	Explain the depth first approach of finding articulation points in a connected graph with necessary algorithm. (13)	BTL 4	Analyzing
14	i) Write short notes on Bi-connectivity. (7) ii) Express different types of graphs with example.(6)	BTL 2	Understanding
15	Discuss the shortest path algorithm with suitable example.(13)	BTL 2	Understanding
16	Summarize the various application of graphs.(13)	BTL 5	Evaluating
17	Demonstrate the breadth first search algorithm with an example.(13)	BTL 3	Applying
	PART C		
1	Consider the graph G given below. The adjacency list of G is also given. Assume that G represents the daily flights between different cities and we want to fly from city A to I with minimum stops. One alternative is to use a breadth-first search of G starting at node A. (15)  **Adjacency lists** A: B, C, D B: E C: B, G D: C, G E: C, F F: C, H G: F, H, I H: E, I I: F	BTL 5	Evaluating
2	i) Formulate the minimum spanning tree for the following graph. (8) 	BTL 6	Creating

	ii) For the following graph, In what order are the vertices visited using BFS and DFS starting from vertex A? Where a choice exists, use alphabetical order.(7) 		
3	Using Dijkstra's algorithm to find the shortest path from the source node A. (15) 	BTL 6	Creating
4	i) Explain weakly connected graph and strongly connected graph with example. (7) ii) State the various graph traversal algorithm. Explain each in detail. (8)	BTL 5	Evaluating
5	Summarize the algorithm to determine the biconnected components in the graph and assess it with an example.(15)	BTL 5	Evaluating

UNIT V - SEARCHING, SORTING AND HASHING TECHNIQUES			
Searching- Linear Search - Binary Search. Sorting - Bubble sort - Selection sort - Insertion sort - Shell sort – Radix sort. Hashing- Hash Functions – Separate Chaining – Open Addressing – Rehashing – Extendible Hashing.			
PART – A			
Q.No	**Questions**	**BT Level**	**Competence**
1	What is hashing?	BTL1	Remembering
2	Define extendible hashing and give its significance.	BTL1	Remembering
3	What is meant by internal and external sorting? Give any two examples for each type.	BTL1	Remembering
4	List the different types of searching	BTL1	Remembering
5	Define rehashing.	BTL1	Remembering
6	Identify the advantage of shell sort over insertion sort.	BTL1	Remembering
7	How many passes does the of insertion sort algorithm do to sort a list of 5 elements? What happens in its i^{th} pass?	BTL2	Understanding
8	Give the types of collision resolution.	BTL2	Understanding
9	Interpret the fastest searching algorithm and give reason.	BTL2	Understanding

10	Distinguish between linear and binary search technique.	BTL2	Understanding
11	Classify the different sorting methods.	BTL3	Applying
12	Apply insertion sort and sort the following elements 3, 1, 4, 1, 5, 9, 2, 6, 5	BTL3	Applying
13	Which hashing technique is best and illustrate with an example?	BTL3	Applying
14	Analyze why do we need a hash table as a data structure as compared to any other data structure?	BTL4	Analyzing
15	Point out the advantages of using open addressing.	BTL4	Analyzing
16	Compare the advantage and disadvantage of separate chaining and linear probing?	BTL4	Analyzing
17	Select the best sorting method out of the following - insertion sort, quick sort and merge sort and give justification.	BTL5	Evaluating
18	Summarize the open addressing hashing method with an example.	BTL5	Evaluating
19	Develop an algorithm for a shell sort.	BTL6	Creating
20	Prepare a simple C Program for a linear search.	BTL6	Creating
21	Interpret what operation does the insertion sort use to move numbers from the unsorted section to the sorted section of the list?	BTL3	Applying
22	Assess the output of selection sort after the 2nd iteration given the following sequence 16 3 46 9 28 14.	BTL5	Evaluating
23	Analyze when does the Bubble Sort Algorithm stop.	BTL4	Analyzing
24	Identify which sorting algorithm is easily adaptable to singly linked lists? Why?	BTL2	Understanding
	PART-B		
1	Describe about selection sort with suitable example.(13)	BTL1	Remembering
2	Examine the algorithm for Insertion sort and sort the following array: 39, 9, 45, 6318, 81, 108, 54, 72, 36 (13)	BTL1	Remembering
3	List the different types of hashing techniques? Explain them in detail with an example. (13)	BTL1	Remembering
4	Show the result of inserting the keys 2, 3, 5, 7, 11, 13, 15, 6, 4 into an initially empty extendible hashing data structure with M = 3. (13)	BTL1	Remembering
5	Write a C program to search a number with the given set of numbers using binary search. (13)	BTL2	Understanding
6	Interpret an algorithm to sort a set of 'N' numbers using bubble sort and demonstrate the sorting steps for the following set of numbers: 30, 52 29, 87, 63, 27, 19, 54. (13)	BTL2	Understanding
7	Discuss the various open addressing techniques in hashing with an example.(13)	BTL2	Understanding
8	(i) Sort the given integers and Show the intermediate results using shell sort: 39, 9, 81, 45, 90, 27, 72, 18. (7) (ii) Write an algorithm to sort an integer array using shell sort. (6)	BTL3	Applying
9	Illustrate with example the open addressing and chaining methods of collision resolution techniques in hashing. (13)	BTL3	Applying
10	Compare working of binary search and linear search technique with example.(13)	BTL4	Analyzing
11	Analyze extendible hashing in brief.(13)	BTL4	Analyzing

12	Explain in detail about separate chaining.(13)	BTL4	Analyzing
13	Formulate the rehashing technique with suitable example.(13)	BTL5	Evaluating
14	Prepare an algorithm to sort the elements using radix sort with example.(13)	BTL6	Creating
15	Illustrate about Re-hashing and Extendible hashing with example.(13)	BTL3	Applying
16	Write an algorithm to implement Bubble sort and assess it with suitable example.(13)	BTL5	Evaluating
17	Discuss the common collision resolution strategies used in closed hashing system.(13)	BTL2	Understanding
	PART C		
1	Mention the different Sorting methods and Explain about any 2 method in detailed Manner. (15)	BTL5	Evaluating
2	Sort the sequence 96, 31, 27,42,76,61,10,4 using shell sort and radix sort and explain the algorithm for shell sort. (15)	BTL6	Creating
3	Given input {4371, 1323, 6173, 4199, 4344, 9679, 1989} and a hash function h(x) =x mod 10. Prepare the resulting for the following: i) Open hash table. (3) ii) Open addressing hash table using linear probing. (4) iii) Open addressing hah table using quadratic probing. (4) iv) Open addressing hash table with second hash h2(x) =7- (xmod7). (4)	BTL6	Creating
4	i) Write and explain non-recursive algorithm for binary search. (8) ii) Using binary search, search the number 26 from the list of numbers and give the steps. 10, 7, 17, 26, 32, 92 (7)	BTL5	Evaluating
5	Write an algorithm for sorting the elements using shell sort. Evaluate the contents of the array after it has gone through a one increment pass of the shell sort. The increment factor is k=3. 23, 3, 7, 13, 89, 66, 6, 44, 18, 90, 98, 57 (15)	BTL5	Evaluating