SRM VALLIAMMAI ENGINEERING COLLEGE
(An Autonomous Institution)

SRM Nagar, Kattankulathur – 603 203

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

QUESTION BANK

III SEMESTER

1904301 - PROGRAMMING & DATA STRUCTURES IN C

Regulation – 2019

Academic Year 2022 – 2023(odd Semester)

Prepared by

Dr. M. Mayuranathan, *Assistant Professor (Sr.G)*
Ms.R.Anitha *Assistant Professor (O.G)*
Mr.K.Shanmugam, *Assistant Professor (O.G)*

SRM VALLIAMMAI ENGINEERING COLLEGE

(An Autonomous Institution)

SRM Nagar, Kattankulathur – 603 203.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

QUESTION BANK

SUBJECT : 1904301 - PROGRAMMING & DATA STRUCTURES IN C

SEM / YEAR: III SEM / II Year

UNIT I - LINEAR DATA STRUCTURES – LIST

Introduction to structure-Abstract Data Types (ADTs) – List ADT – array-based implementation – linked list implementation —singly linked lists- circularly linked lists- doubly-linked lists – applications of lists – Polynomial Manipulation.

PART - A

Q.No	Questions	BT Level	Competence
1.	Define ADT. Give any two examples.	BTL1	Remembering
2.	Define list. Mention any two operations that are performed on a list.	BTL1	Remembering
3.	List out the areas in which data structures are applied extensively.	BTL1	Remembering
4.	Define data structure with example.	BTL1	Remembering
5.	What is circular linked list?	BTL1	Remembering
6.	List out the advantage of circular linked list.	BTL1	Remembering
7.	Distinguish between linear and nonlinear data structures.	BTL2	Understanding
8.	Interpret the advantages of linked lists over arrays.	BTL2	Understanding
9.	Differentiate arrays and linked lists.	BTL2	Understanding
10.	Give an example for linked list application.	BTL2	Understanding
11.	Examine a doubly linked list with neat diagram.	BTL3	Applying
12.	Illustrate the basic operations carried out in a linked list.	BTL3	Applying
13.	Show the ways in which list ADT can be implemented.	BTL3	Applying
14.	Compare calloc() and realloc() function and mention its application in linked list.	BTL4	Analyzing
15.	Analyze and write a routine to find position of given element in singly linked list.	BTL4	Analyzing
16.	Analyze and write the linked list representation of a polynomial: $p(x)=4x^3+6x^2+7x+9$	BTL4	Analyzing
17.	Should arrays or linked lists be used for the following types of applications? Support your justification. 1. Many search operations in sorted list. 2. Many search operations in unsorted list.	BTL5	Evaluating
18.	Compare between linear linked list and Circular linked list.	BTL5	Evaluating
19.	Design a routine to delete an element in a linked list.	BTL6	Creating

20.	Develop an algorithm for insertion operation in a singly linked list.	BTL6	Creating
21.	Describe the advantages of ADT.	BTL2	Understanding
22.	Show the disadvantages of linked list over arrays.	BTL3	Applying
23.	Explain about linked list.	BTL4	Analyzing
24.	Summarize the areas in which data structures are applied extensively.	BTL5	Evaluating
PART - B			
1.	(i). Describe in detail about Polynomial manipulation in linked list. (6) (ii). Write procedure to Add two polynomial and explain with suitable example. (7)	BTL1	Remembering
2.	What is a linked list? Describe the suitable routine segments for any four operations. (13)	BTL1	Remembering
3.	(i). Examine the algorithms to implement the doubly linked list. (7) (ii). Examine with example to perform all the operations on the created list. (6)	BTL1	Remembering
4.	(i). Identify the array implementation of list. (6) (ii). Explain all its operations with code. (7)	BTL1	Remembering
5.	Discuss the creation of a doubly linked list and write routine to insert an element in doubly linked list. (13)	BTL2	Understanding
6.	(i). What are the ways to insert a node in linked list? (4) (ii). Write an algorithm for inserting a node before a given node in a linked list. (9)	BTL2	Understanding
7.	Differentiate single linked list and doubly linked list with an example. (13)	BTL2	Understanding
8.	Explain the application of linked list in dynamic storage in detail. (13)	BTL3	Applying
9.	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
10.	(i). Analyze and write procedure for circular linked list. (4) (ii). Analyze and write code for create, insert, delete, display operations using circular linked list. (9)	BT L4	Analyzing
11.	Explain the various operations of the list ADT with examples. (13)	BTL4	Analyzing
12.	Analyze the doubly linked list and circular linked list. (9) Mention its advantages and disadvantages. (4)	BTL4	Analyzing
13.	(i). Explain the steps involved in insertion into a singly linked list. (6) (ii). How nodes are inserted after a specific node? (7)	BTL5	Evaluating
14.	Develop a C program for linked list implementation of list. (13)	BTL6	Creating
15.	(i). Express the polynomial representation for $6x^3+9x^2+7x+1$ using linked list. (6) (ii). Write procedure to multiply two polynomial and explain with suitable example. (7)	BTL2	Understanding
16.	(i).Illustrate are the various operations of an array? (6) Write a procedure to insert, delete and search an element in to an array. (7)	BTL3	Applying

17.	Explain the steps involved in deletion into a singly linked list. (13)	BTL5	Evaluating

PART – C

1.	Recommend an algorithm to add two polynomials represented by linked representation. Apply the function for the following input. $A=3x^2+2x^1+1, B=8x^2+3x^1+10$ (15)	BTL5	Evaluating
2.	Compose an algorithm to i) Reverse the elements of a single linked lists. (5) ii) Count the number of nodes in a given singly linked list. (5) iii) Searching the element from linked list. (5)	BTL6	Creating
3.	Given a list 100, 200, 300, 400, generalize the steps and write routine to delete a node from the beginning of the linked list, deletion of last node in a list and deletion of middle node in a list. (15)	BTL5	Evaluating
4.	i) Develop 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) Write a non-recursive procedure to reverse a singly linked list. (5)	BTL6	Creating
5.	Explain about the different types of data structure? Discuss about the different data structure in detail with an example.	BTL5	Evaluating

UNIT II - LINEAR DATA STRUCTURES – STACKS, QUEUES

Stack ADT – Operations – Applications – Evaluating arithmetic expressions- Conversion of Infix to postfix expression – Queue ADT – Operations – Circular Queue – Priority Queue – Deque –applications of queues.

PART - A

Q.No	Questions	BT Level	Competence
1.	Define stack and specify the operations	BTL1	Remembering
2.	List any four applications of stack?	BTL1	Remembering
3.	Given the prefix for an expression. Write its postfix: ++A*BCD and +*AB*CD	BTL2	Understanding
4.	Given a infix expression convert it into postfix expression using stack a+b*(c^d-e)^(f+g*h)-i	BTL2	Understanding
5.	Discuss the postfix and prefix forms of the expression? A+B*(C-D)/(P-R)	BTL2	Understanding
6.	Illustrate the purpose of top and pop?	BTL3	Applying
7.	What causes underflow of stack? How it could be avoided?	BTL4	Analyzing
8.	How to implement stack using linked list.	BTL4	Analyzing
9.	Summarize the rules followed during the infix to postfix Conversions.	BTL5	Evaluating
10.	Generalize a routine to return top element in a stack using linked list.	BTL6	Creating
11.	Define double ended queue.	BTL1	Remembering
12.	List the applications of a queue.	BTL1	Remembering
13.	What are priority queues? What are the ways to implement priority queue?	BTL1	Remembering
14.	What is circular queue?	BTL1	Remembering
15.	Circular queue is better than standard linear queue, Why?	BTL2	Understanding

16.	Classify the different types of queues.	BTL3	Applying
17.	Show a routine to perform enqueue operation in a queue.	BTL3	Applying
18.	Differentiate between double ended queue and circular queue.	BTL4	Analyzing
19.	For railway reservation the queue data structure is preferred –Justify.	BTL5	Evaluating
20.	Develop an algorithm for deleting an element in a double ended queue.	BTL6	Creating
21.	Express the various Operations performed on the Stack.	BTL2	Understanding
22.	Examine the usage of stack in recursive algorithm implementation.	BTL3	Applying
23.	Analyze how do you test for an empty Queue?	BTL4	Analyzing
24.	Explain about Dequeue operation.	BTL5	Evaluating
PART – B			
1.	Describe about stack ADT using array in detail. (13)	BTL1	Remembering
2.	(i).Give an algorithm for push and pop operations on stack using a linked list with an example. (8) (ii). Describe the function to examine whether the stack is full() or empty(). (5)	BTL1	Remembering
3.	(i)Write algorithm to check if the parenthesized arithmetic expression contains balanced parenthesis. (6) (ii). Convert such expression to postfix form and evaluate it. Illustrate with example. (7)	BTL2	Understanding
4.	(i). Give an algorithm to convert an infix expression to a postfix expression using stack. (6) (ii). Convert infix to postfix expression with suitable example. (7)	BTL2	Understanding
5.	Illustrate the application of stack with an example. (13)	BTL3	Applying
6.	Show the procedure to convert the infix expression to postfix expression and steps involved in evaluating the postfix expression. Convert the expression A-(B/C+(D%E*F)/G)*H to postfix form. Evaluate the given postfix expression 9 3 4 * 8 + 4 / -. (13)	BTL3	Applying
7.	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) 43*67+5-+ (4)	BTL5	Evaluating
8.	Briefly describe the operations of queue with example. (13)	BTL1	Remembering
9.	Describe about implementation of queue ADT using linked list with suitable example. (13)	BTL1	Remembering
10.	(i) Discuss about the implementation of queue functions using arrays. (6) (ii). Explain about the implementation with suitable example with code. (7)	BTL2	Understanding
11.	Explain application of queue with suitable example. (13)	BTL4	Analyzing
12.	(i). What are circular queues. (2) (ii). Explain the procedure to insert an element to circular queue using array implementation. (5) (iii). Explain the procedure to delete an element from a circular queue using array implementation. (6)	BTL4	Analyzing

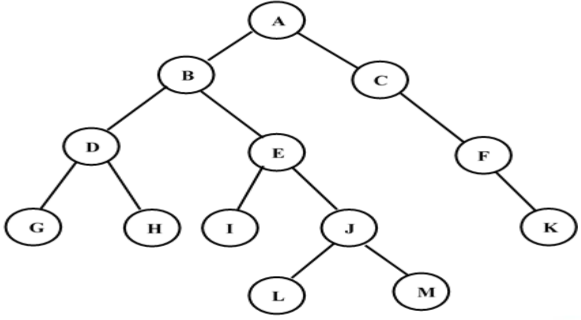
13.	Analyze the implementation of priority queue. (13)	BTL4	Analyzing
14.	(i). Prepare an algorithm to perform the operations in a double ended queue. (6) (ii). Prepare an suitable code to perform the operations in a double ended queue. (7)	BTL6	Creating
15.	(i). Give an algorithm to reverse a list of given numbers using stack. (6) (ii). Give a program to reverse a list of given numbers using stack.(7)	BTL2	Understanding
16.	Demonstrate a program to calculate the factorial and finding a GCD of a given number using recursion. (13)	BTL3	Applying
17.	Evaluate and write a program to create a linear queue of 10 values. (13)	BTL5	Evaluating
PART – C			
1.	Develop a C program for linked list implementation of stack. (15)	BTL6	Creating
2.	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)	BTL5	Evaluating
3.	A circular queue has a size of 5 and has 3 elements 10, 20 and 40 where F=2 and R=4.After inserting 50 and 60,what is the value of F and R. Trying to insert 30 at this stage what happens? Delete 2 elements from the queue and insert 70, 80 & 90.Assess the sequence of steps with necessary diagrams with the value of F & R. (15)	BTL5	Evaluating
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.	Design a program to convert the expression “a+b” into “ab+” (15)	BTL6	Creating

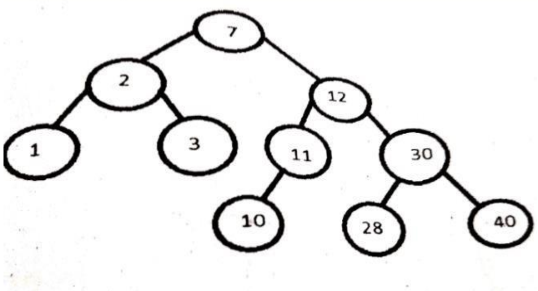
UNIT III - NON LINEAR DATA STRUCTURES – TREES

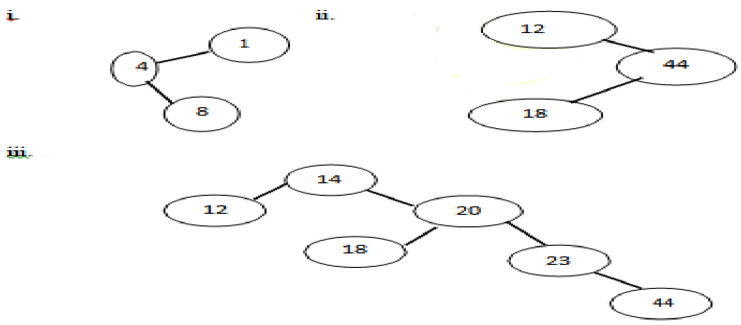
Tree ADT – tree traversals – Binary Tree ADT – expression trees – binary search tree ADT – Threaded Binary Trees-AVL Trees – B-Tree – B+ Tree-Heap

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.	For the given tree a. List the siblings for node E b. Compute the height.  A B C D E F G H I J K L M	BTL 1	Remembering
8.	What are threaded binary trees? Give its advantages.	BTL 1	Remembering
9.	Define balance factor of AVL Tree.	BTL 1	Remembering
10.	Simulate the result of inserting 2, 1, 4, 5, 9, 3, 6, 7 into an initially empty AVL Tree.	BTL 6	Creating
11.	Define an expression tree. Give an example for it.	BTL 2	Understanding
12.	Summarize tree traversal and mention the type of traversals?	BTL 2	Understanding
13.	Differentiate B tree and B+ tree	BTL 2	Understanding
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.	Discuss about B tree and B+ tree.	BTL 2	Understanding
22.	Show the operations performed in list?	BTL 3	Applying
23.	Point out the applications of trees.	BTL 4	Analyzing
24.	Explain how the polynomial is represented using linked lists?	BTL 5	Evaluating
	PART – B		
1.	(i). Write an algorithm for preorder, inorder and postorder traversal of a binary tree. (7) (i). Write suitable code for preorder, inorder and postorder traversal of a binary tree. (6)	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.	(i). Write an algorithm for inserting a node in a binary search tree. (6) (ii). Write an algorithm for deleting a node in a binary search tree. (7)	BTL 1	Remembering
4.	Describe the concept of threaded binary tree with example. (13)	BTL 1	Remembering

5.	(i). Discuss in detail the various methods in which a binary tree can be represented. (9) (ii). Discuss the advantage and disadvantage of each method. (4)	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.	Discuss about tree in brief with suitable example. (13)	BTL 2	Understanding
16.	Classify the various operations of the list ADT with examples (13)	BTL 3	Applying
17.	(i). Explain in detail about linked list? (6) (ii). Write a C program for linked list implementation of list. (7)	BTL 5	Evaluating
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,Postorer traversals (6)  	BTL 6	Creating

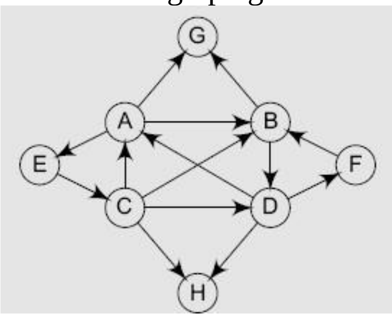
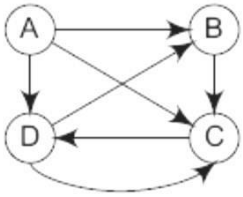
	iii) Delete 11 and 10 from the above binary search tree and display the tree after each deletion. (4)		
3.	i) Draw B-Tree of order m = 5 for the keys{K, O,S,V,M,F,B,G,T,U,W} (5) ii) Delete the keys K and G in order. (5) iii) Justify the number of splits needed for inserts / delete with proper reasons. (5)	BTL 5	Evaluating
4.	**Construct** AVL tree for the followings after rotation. (5+5+5)  i. 4 1 8 ii. 12 44 18 iii. 14 12 20 18 23 44	BTL 6	Creating
5.	(i)Summarize about binary tree and its classification with suitable example. (10) (ii)Explain briefly about In order, post order, pre order with suitable example. (5)	BTL 5	Evaluating

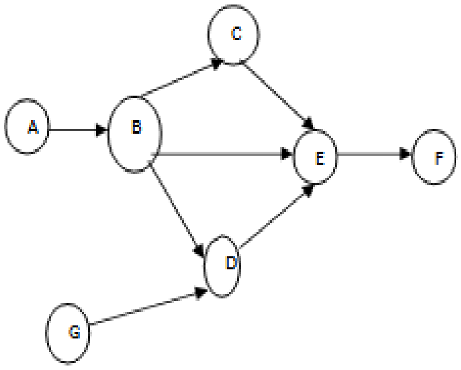
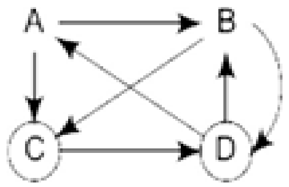
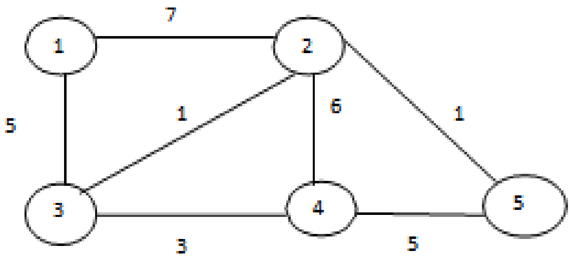
UNIT IV - NON LINEAR DATA STRUCTURES - GRAPHS

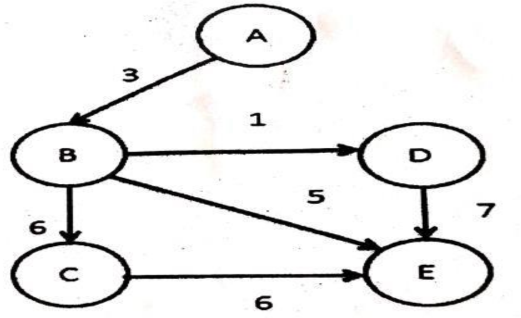
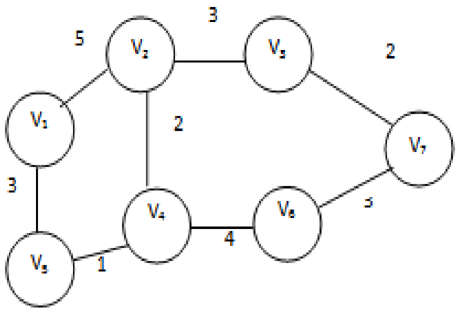
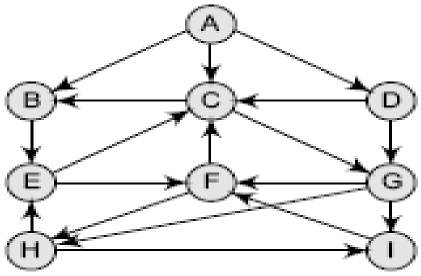
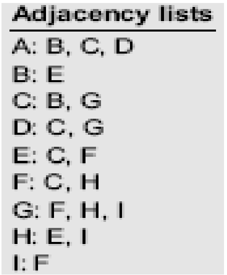
Definition – Representation of Graph – Types of graph – Breadth-first traversal – Depth-first traversal – Topological Sort – Dijkstra's algorithm-minimum spanning tree-Bi-connectivity.

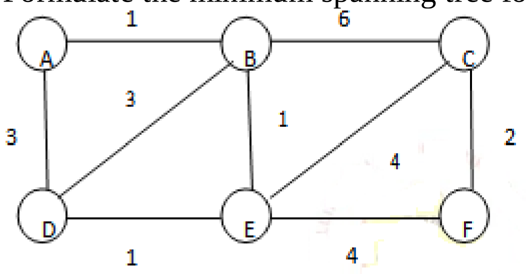
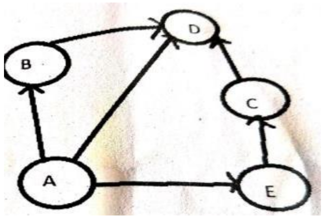
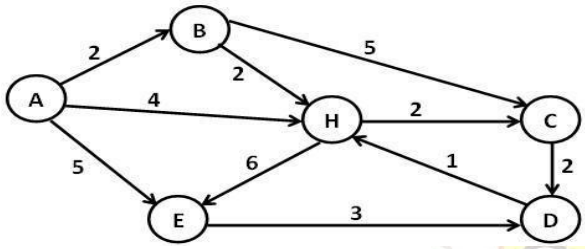
PART - A

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 representation 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.	Discuss topological sorting.	BTL 2	Understanding
22.	Explain BFS	BTL 5	Evaluating
23.	Explain indegree and out degree of a graph.	BTL 4	Analyzing
24.	Illustrate strongly connected graph.	BTL 3	Applying
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)	BTL 4	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.	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
10.	Discuss the prim's algorithm for minimum spanning tree. Give an example. (13)	BTL 2	Understanding
11.	i) Write a program to find an Euler circuit in a graph. (7) ii) Trace the algorithm for the given graph. (6)  3 5 V₂ V₃ 2 V₁ 2 V₇ 3 V₄ V₆ 3 4 V₅ 1	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.	Classify the various application of graphs. (13)	BTL 3	Applying
17.	Explain the breadth first search algorithm with an example. (13)	BTL 5	Evaluating
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)  A B C D E F 1 6 3 3 1 4 2 1 4 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)  	BTL 6	Creating
3.	Using Dijkstra's algorithm to find the shortest path from the source node A. (15)  B 2 5 2 A 4 H 2 C 5 6 1 2 E 3 D	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.	Explain the algorithm to determine the biconnected components in the graph (13)	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.

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.	Describe the advantages of selection sort.	BTL2	Understanding
22.	Show an example about collision in hashing	BTL3	Applying
23.	Explain the application of Hashing	BTL4	Analyzing
24.	Assess the hash values of keys 1234 and 5462.	BTL5	Evaluating
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 shellsort: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.	Interpret the numbers given below using radix sort. 345, 654, 924, 123, 567, 472, 555, 808, 911 (13)	BTL2	Understanding
16.	Examine an algorithm to search an element in an array using the linear search technique with an example. (13)	BTL3	Applying
17.	Explain about what is sorting and discuss in detail about its types. (13)	BTL5	Evaluating
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- (x mod 7). (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.	Test the numbers 7, 1, 4, 12, 67, 33, and 45 for bubble sort and write a program to implement bubble sort. How many swaps will be performed to sort these numbers using the bubble sort? (13)	BTL5	Evaluating