

VALLIAMMAI ENGINEERING COLLEGE

SRM Nagar , Kattankulathur-603203

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

QUESTION BANK

SUBJECT : CS8391- DATA STRUCTURES

SEM/YEAR :III/II

| Q. No | Questions | BT Level | Competence |
|-----------------|---|----------|------------|
| 1 | Define ADT. Give any two examples. | BTL1 | Remember |
| 2 | Distinguish between linear and nonlinear data structures. | BTL2 | Understand |
| 3 | Compare calloc() and realloc() function and mention its application in linkedlist. | BTL4 | Analyze |
| 4 | Describe the differences between singly and doubly linked lists. | BTL1 | Remember |
| 5 | List out the areas in which data structures are applied extensively. | BTL1 | Remember |
| 6 | Define nonlinear data structure. | BTL1 | Remember |
| 7 | Compare singly linked list with circular linked list. | BTL5 | Evaluate |
| 8 | State the advantage of ADT. | BTL1 | Remember |
| 9 | List out the advantage of circular linked list. | BTL1 | Remember |
| 10 | Binary search cannot be performed on a linked list. Examine . | BTL3 | Apply |
| 11 | Discuss the advantages and disadvantages of linked lists and arrays. | BTL2 | Understand |
| 12 | Give an example for linked list application. | BTL2 | Understand |
| 13 | Specify the use of Header node in a linked list. | BTL6 | Create |
| 14 | Illustrate the use of linked list with an example. | BTL3 | Apply |
| 15 | Show the ways in which list ADT can be implemented. | BTL3 | Apply |
| 16 | Differentiate arrays and linked lists. | BTL2 | Understand |
| 17 | Analyze and write a find routine in array implementation. | BTL4 | Analyze |
| 18 | Analyze and write the array representation of a polynomial: $p(x) = 4x^3 + 6x^2 + 7x + 9$ | BTL4 | Analyze |
| 19 | Should arrays or linked lists be used for the following types of applications? Support your justification . | BTL5 | Evaluate |
| | 1. Many search operations in sorted list. | | |
| | 2. Many search operations in Unsorted list. | | |
| 20 | Develop an algorithm for insertion operation in a singly linked list. | BTL6 | Create |
| PART – B | | | |
| 1 | Describe the following: | BTL1 | Remember |
| | i. Applications of lists. (5) | | |
| | ii. Polynomial manipulation. (8) | | |
| 2 | i. What is a linked list? (2) | BTL1 | Remember |
| | ii. Describe the suitable routine segments for any four operations. (11) | | |
| 3 | List an algorithm to perform the following operations in a doubly linked list. | BTL1 | Remember |
| | i. Insert a node at the end of the list. (6) | | |
| | ii. Delete the last node in the list. (7) | | |
| 4 | i. Discuss the insertion and deletion procedures for cursor based linked | BTL2 | Understand |

| | | | | |
|-----------------|---|------|------|------------|
| | lists. | (7) | | |
| | ii. Give an algorithm for the deletion and reverse operations on doubly linked list. | (6) | | |
| 5 | i. Give the algorithm to perform insertion on a doubly linked list. | (7) | BTL2 | Understand |
| | ii. Give the algorithm to perform deletion on a doubly linked list. | (6) | | |
| 6 | Write an algorithm to demonstrate a polynomial using a linked list for | | | |
| | i. Addition and Subtraction. | (7) | BTL3 | Analyze |
| | ii. Multiplication operations. | (6) | | |
| 7 | Analyze and write algorithm for Circular Linked list for the following operations using structure pointer. | | BTL4 | Analyze |
| | i. Create & Insert | (6) | | |
| | ii. Delete & Display. | (7) | | |
| 8 | Explain the application of linked list in detail. | | BTL4 | Apply |
| | i. Radix sort. | (7) | | |
| | ii. Multi list. | (6) | | |
| 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) | BTL5 | Evaluate |
| 10 | Develop a program to add the values of the nodes of a linked list and then calculate the mean. | (13) | BTL6 | Create |
| 11 | Describe the various operations of the list ADT with examples. | (13) | BTL1 | Remember |
| 12 | i) Illustrate the polynomial representation for $6x^3 + 9x^2 + 7x + 1$ using linked list. Write procedure to add and multiply two polynomials and explain with suitable example. | (7) | BTL3 | Analyze |
| | ii) What are the ways to insert a node in linked list? Write an algorithm for inserting a node before a given node in a linked list. | (6) | | |
| 13 | Explain the steps involved in the following insertion operations in a singly linked list. | | BTL4 | Apply |
| | i. Insert the node in the start and End. | (7) | | |
| | ii. Insert the node in the middle of the List | (6) | | |
| 14 | Discuss an algorithm for linked list implementation of list. | (13) | BTL2 | Understand |
| PART – C | | | | |
| 1 | Create an algorithm to add two polynomials using linked list. | (15) | BTL6 | Create |
| 2 | Explain an algorithm to split a linked list into two sub lists containing odd and even ordered elements in them respectively. | (15) | BTL5 | Evaluate |
| 3 | Explain an algorithm to merge two sorted linked lists into a single sorted list. | (15) | BTL5 | Evaluate |
| 4 | Design algorithm for various operations performed on circular linked list. Extend the algorithm defined in the previous question for the doubly linked circular list. | (15) | BTL6 | Create |

UNIT II - LINEAR DATA STRUCTURES – STACKS, QUEUES

Stack ADT – Evaluating arithmetic expressions- other applications- Queue ADT – circular queue implementation – Double ended Queues – applications of queues

PART – A

| Q. No | Questions | BT Level | Competence |
|-------|---|----------|------------|
| 1 | Point out the advantage of representing stack using a linked list than array. | BTL4 | Analyze |
| 2 | Point out the rules followed during the infix to postfix conversions. | BTL4 | Analyze |
| 3 | Compare the working of stack and queue data structure. | BTL5 | Evaluate |
| 4 | Develop an algorithm for inserting a new element into the stack. | BTL6 | Create |
| 5 | What are priority queues? What are the ways to implement priority queue? | BTL1 | Remember |
| 6 | List any four applications of stack. | BTL1 | Remember |
| 7 | Given the prefix for an expression. Write its postfix: -*+abc/ef-g/hi | BTL2 | Understand |
| 8 | Describe how the following "infix" expression is evaluated with the help of the help of Stack : $5 * (6 + 2) - 12 / 4$ | BTL2 | Understand |
| 9 | Give the postfix and prefix forms of the expression: $A + B * (C - D) / (P - R)$ | BTL2 | Understand |
| 10 | Define double ended queue. | BTL1 | Remember |
| 11 | List the applications of a queue. | BTL1 | Remember |
| 12 | What are the applications of priority queue? | BTL1 | Remember |
| 13 | What is circular queue? | BTL1 | Remember |
| 14 | Circular queue is better than standard linear queue, Why? | BTL2 | Understand |
| 15 | Classify the different types of queues. | BTL3 | Apply |
| 16 | Illustrate the difference between a queues and linked lists with an example. | BTL3 | Apply |
| 17 | Complete a routine to display the contents of queue. | BTL3 | Apply |
| 18 | Analyze and write a routine to check whether the queue is full or empty. | BTL4 | Analyze |
| 19 | For railway reservation the queue data structure is preferred – Justify . | BTL5 | Evaluate |
| 20 | Develop an algorithm for deleting an element in a double ended queue. | BTL6 | Create |

PART – B

| | | | |
|---|---|------|----------|
| 1 | Describe with an example how to evaluate arithmetic expressions using stacks. (13) | BTL1 | Remember |
| 2 | i. Explain array based implementation of stacks. (7) | BTL4 | Apply |
| | ii. Explain linked list implementation of stacks. (6) | | |
| 3 | i. Describe about stack ADT in detail. (7) | BTL1 | Remember |
| | ii. Explain any one application of stack. (6) | | |
| 4 | Explain the following expressions with an example. | | |
| | i. Prefix and infix . (7) | BTL4 | Apply |
| | ii. Postfix. (6) | | |
| 5 | i. Write an algorithm to convert an infix expression to a postfix expression. Trace the algorithm to convert the infix expression $(a+b)*c/d+e/f$ to a postfix expression. (8) | BTL5 | Evaluate |

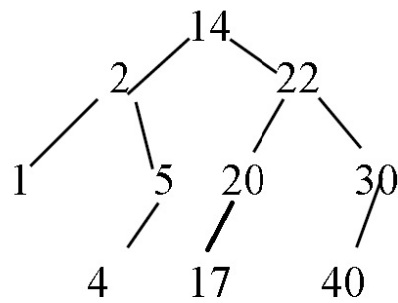
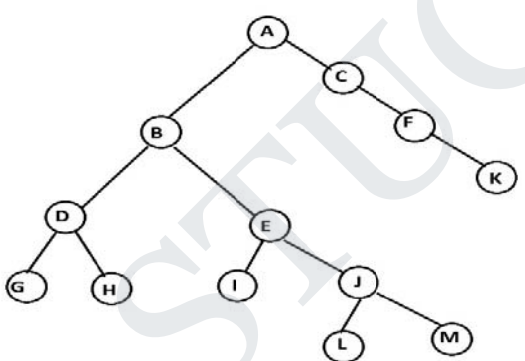
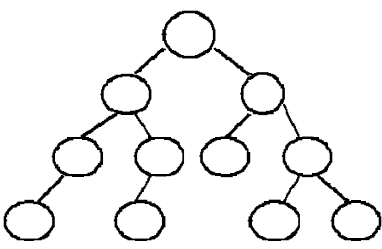
| | | | |
|-----------------|--|------|------------|
| | ii. Justify the need for Infix and Postfix expression. (5) | | |
| 6 | i. Give an algorithm for push and pop operations on stack using a linkedlist. (7) | BTL2 | Understand |
| | ii. Discuss about addition and deletion operations performed on a circular queue with necessary algorithms. (6) | | |
| 7 | i. Describe the process of postfix expression evaluation with an example. (7) | | |
| | ii. Describe the process of conversion from infix expression to postfix expression using stack. (6) | BTL2 | Understand |
| 8 | i. Write an algorithm that checks if expression is correctly parenthesized using stack and illustrate with an example. (7) | BTL3 | Analyze |
| | ii. Write the function to examine whether the stack is full() or empty() (6) | | |
| 9 | i. Describe about queue ADT in detail. (7) | BTL1 | Remember |
| | ii. Explain any one application of queue with suitable example. (6) | | |
| 10 | Briefly describe the operations of queue with examples. (13) | BTL1 | Remember |
| 11 | Analyze and write an algorithm to implement queue functions using arrays. (13) | BTL4 | Analyze |
| 12 | Develop an algorithm to perform the four operations in a double ended queue that is implemented as an array. (13) | BTL6 | Create |
| 13 | Discuss circular queue and its implementation. (13) | BTL2 | Understand |
| 14 | Illustrate the enqueue and dequeue operations on double ended queues. (13) | BTL3 | Analyze |
| PART – C | | | |
| 1 | Develop and Show the simulation using stack for evaluation of the following expression : $12 + 3 * 14 - (5 * 16) + 7$ (15) | BTL6 | Create |
| 2 | Explain an algorithm to implement the circular queue using arrays. List the applications of Queues. (15) | BTL5 | Evaluate |
| 3 | Assess the difference between double ended queue and circular queue. Show the simulation using stack for the following expression to convert infix to postfix: $p * q = (r-s / t)$. (15) | BTL5 | Evaluate |
| 4 | Develop an algorithm to explain Priority Queue, dequeue and the applications of queues. (15) | BTL6 | Create |

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.

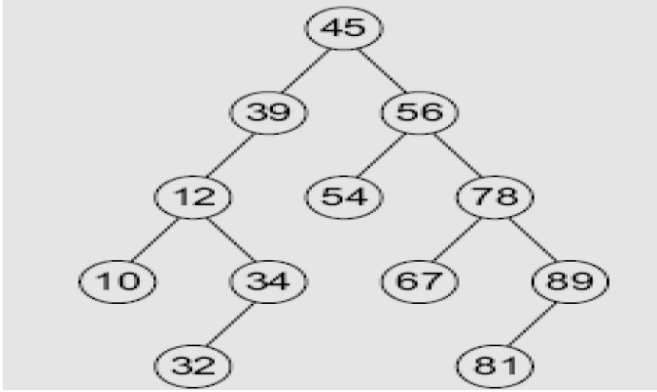
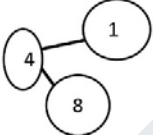

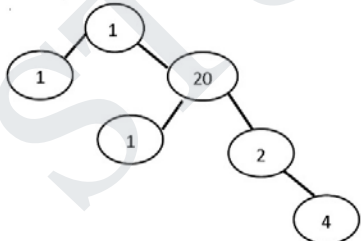
PART-A

| Q. No | Questions | BT Level | Competence |
|-------|--|----------|------------|
| 1 | If the depth of the binary tree is k, the maximum number of nodes in the binary tree is $2^k - 1$. Justify | BTL5 | Evaluate |

| | | | |
|----|--|------|------------|
| 2 | <p>For the given binary search tree, if we remove the root and replace it with something from left subtree. What will be the value of the new root? Justify your answer.</p>  | BTL5 | Evaluate |
| 3 | <p>Define a fully binary tree. Give an example.</p> | BTL1 | Remember |
| 4 | <p>Create an expression tree for the expression $a*(b+c)+((d+e*f)*g)$</p> | BTL6 | Create |
| 5 | <p>How does the AVL tree differ from binary search tree?</p> | BTL4 | Analyze |
| 6 | <p>What are the various rotations in AVL trees?</p> | BTL1 | Understand |
| 7 | <p>List the applications of trees.</p> | BTL1 | Remember |
| 8 | <p>What are threaded binary trees? Give its advantages</p> | BTL2 | Understand |
| 9 | <p>Define balance factor of AVL Tree.</p> | BTL1 | Remember |
| 10 | <p>How do we calculate the balance factor for each node in an AVL tree?</p> | BTL2 | Understand |
| 11 | <p>Simulate the result of inserting 3,1,4,6,2,8,9 into an initially empty AVLTree.</p> | BTL6 | Create |
| 12 | <p>Discuss with respect to following tree:</p> <ol style="list-style-type: none"> List the siblings for node E. Compute the height  | BTL2 | Understand |
| 13 | <p>Number the following binary tree to traverse it in</p> <ol style="list-style-type: none"> Preorder Inorder  | BTL2 | Understand |
| 14 | <p>Explain why binary search cannot be performed on a linked list.</p> | BTL4 | Analyze |
| 15 | <p>How do you calculate the depth of a B-Tree?</p> | BTL3 | Apply |
| 16 | <p>List out the various operations that can be performed on B-trees</p> | BTL1 | Remember |

| PART C | | | |
|--------|--|------|----------|
| 17 | List out the properties of B+ -Trees | BTL1 | Remember |
| 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 | BTL3 | Apply |
| 19 | Analyze the properties of binary heap. | BTL4 | Analyze |
| 20 | Define a heap and show how it can be used to represent a priority queue. | BTL3 | Apply |

| PART B | | | |
|--------|---|------|------------|
| 1 | Write an algorithm for preorder, inorder and postorder traversal of a binary tree. (13) | BTL1 | Remember |
| 2 | Explain the following operations on a binary search tree with suitable algorithms | BTL4 | Analyze |
| | i. Find a node (6) | | |
| | ii. Find the minimum and maximum elements of binary search tree (7) | | |
| 3 | i. Write short notes on threaded binary tree (7) | BTL1 | Remember |
| | ii. Describe an iterative algorithm to traverse a tree in preorder (6) | | |
| 4. | Write an algorithm for inserting and deleting a node in a binary search tree. (13) | BTL1 | Remember |
| 5 | Discuss in detail the various methods in which a binary tree can be represented. Discuss the advantage and disadvantage of each method (13) | BTL2 | Understand |
| 6 | i. Explain the B+ tree and its properties with an Example (7) | BTL5 | Evaluate |
| | ii. What are the steps to convert general tree to binary tree? (6) | | |
| 7 | i. Construct B Tree to insert the following key elements (order of the tree is 3) 5, 2, 13, 3, 45, 72, 4, 6, 9, 22 (7) | BTL6 | Create |
| | ii. Draw a B Tree of order 6 (6) | | |
| 8 | i. Discuss how to insert an element in a AVL tree, Explain with algorithm. (7) | BTL2 | Understand |
| | ii. Explain how deletion can take place in AVL trees with suitable algorithms (6) | | |
| 9 | i. What are AVL trees? Describe the different rotations defined for AVL tree. (7) | BTL1 | Remember |
| | ii. Insert the following elements step by step in sequence into an empty AVL tree 15, 18, 20, 21, 28, 23, 30, 26 (6) | | |
| 10 | i. Point out the operations of B-tree using 2-3 tree. (7) | BTL4 | Analyze |
| | ii. Explain the operations of threaded binary tree. (6) | | |
| 11 | Discuss the different traversal techniques in binary tree with suitable algorithms and examples? (13) | BTL2 | Remember |
| 12 | Explain the construction of expression tree with example. (7) | BTL4 | Analyze |
| | Give the applications of trees (6) | | |
| 13 | i. Show the result of inserting 15, 17, 6, 19, 11, 10, 13, 20, 8, 14, 12 one at a time into an initially empty binary min heap. (7) | BTL3 | Apply |
| | ii. Show the result of performing three delete min operations in the final binary min heap obtained. (6) | | |
| 14 | i. Illustrate how delete operation is performed on binary heap? (7) | BTL3 | Apply |
| | ii. Write suitable operations for percolate up and percolate down operations in a binary heap. (6) | | |

| | | | |
|---|---|-------|----------|
| 1 | <p>Consider the binary search tree given below. (15)</p> <p>Find the result of in-order, pre-order, and post-order traversals.</p> <p>Show the deletion of the root node</p> <p>Insert 11, 22, 33, 44, 55, 66, and 77 in the tree</p>  | BTL 5 | Evaluate |
| 2 | <p>i. Compare B trees with B+ trees. (7)</p> <p>ii. Create a B+ tree of order 5 for the following data arriving in sequence: (8)</p> <p>90, 27, 7, 9, 18, 21, 3, 4, 16, 11, 21, 72</p> | BTL 6 | Create |
| 3 | <p>i. Draw B – Tree pf order m = 5 for the keys {K, O,S,V,MF,B,G,T,U,W} (5)</p> <p>ii. Delete the keys K and G in order. (5)</p> <p>iii. Justify the number of splits needed for inserts / delete with proper reasons. (5)</p> | BTL 5 | Evaluate |
| 4 | <p>Construct AVL tree for the followings after rotation. (4)</p> <p>i.  (4)</p> <p>ii.  (4)</p> <p>iii.  (7)</p> | BTL 6 | Create |

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 | What is a graph and its types? | BTL 1 | Remember |

| | | | |
|----|--|-------|------------|
| 2 | Consider the graph given below. Create the adjacency matrix of it | BTL 6 | Create |
| | | | |
| 3 | Find out the in-degree and out-degree of each node in the given graph | BTL 3 | Apply |
| 4 | Create a complete undirected graph having five nodes | BTL 6 | Create |
| 5 | Given the following adjacency matrix, draw the weighted graph. | BTL 2 | Understand |
| 6 | When do you say a graph is bi-connected? | BTL 1 | Remember |
| 7 | Give the purpose of Dijkstra's algorithm. | BTL 2 | Understand |
| 8 | Differentiate cyclic and acyclic graph | BTL 4 | Analyze |
| 9 | Classify strongly connected and weakly connected graph. | BTL 3 | Apply |
| 10 | How to find all articulation points in a given graph? | BTL 3 | Apply |
| 11 | Define the length of the graph. | BTL 1 | Remember |
| 12 | Define minimum spanning tree. Give an example | BTL 2 | Understand |
| 13 | State the principle of Topological sorting. | BTL 1 | Remember |
| 14 | Explain procedure for Depth first search algorithm. | BTL 4 | Analyze |
| 15 | Analyze Bi-connectivity. | BTL 4 | Analyze |
| 16 | Prove that the number of edges in a complete graph of n vertices is $n(n-1)/2$ | BTL 5 | Evaluate |
| 17 | In a complete graph with n vertices, show that the number of spanning trees is at least $2^{n-1}-1$ | BTL 5 | Evaluate |
| 18 | What are Euler circuits? | BTL 1 | Remember |
| 19 | Give two applications of graphs. | BTL 2 | Understand |
| 20 | What is residual graph? | BTL 1 | Remember |

| PART B | | | |
|---------------|--|-------|----------|
| 1 | Describe in detail about the following representations of a graph. i. Adjacency Matrix (7) ii. Adjacency List (6) | BTL 1 | Remember |
| 2 | i. Consider the given directed acyclic graph D. Sort the nodes D by applying topological sort on 'D' (7) | BTL 3 | Apply |

| | | | | |
|----|---|------------|-------|------------|
| | <p>ii. Consider the graph given below and show its adjacency list in the memory</p> | (6) | BTL1 | Remember |
| 3 | <p>i. Explain the topological sorting of a graph G with example. ii. Quote the step wise procedure for topological sort.</p> | (7) (6) | BTL4 | Analyze |
| 4 | <p>Differentiate depth-first search and breadth-first search traversal of a graph with suitable examples.</p> | (13) | BTL4 | Analyze |
| 5 | <p>i. Explain with algorithm, How DFS be performed on a undirected graph. ii. Show the algorithm for finding connected components of an undirected graph using DFS, and derive the time complexity of the algorithm.</p> | (7) (6) | BTL4 | Analyze |
| 6 | <p>i. Discuss an algorithm for Breadth first Search on a graph ii. Give an example based on the algorithm.</p> | (7) (6) | BTL2 | Understand |
| 7 | <p>i. Illustrate Kruskal's algorithm to find the minimum spanning tree of a graph. ii. Trace the algorithm for the following graph</p> | (7) (6) | BTL 3 | Apply |
| 8 | <p>Compare any two applications of Graph and describe with your own example</p> | (13) | BTL2 | Analyze |
| 9 | <p>Describe any one of the shortest path algorithms with suitable example</p> | (13) | BTL1 | Remember |
| 10 | <p>Discuss the prim's algorithm for minimum spanning tree. Give an example.</p> | (13) | BTL2 | Understand |
| 11 | <p>i. Write a program to find an Euler circuit in a graph. ii. Trace the algorithm for the given graph.</p> | (7) (6) | BTL1 | Remember |
| 12 | <p>Develop an algorithm to compute the shortest path using Dijkstra's algorithm. Validate the algorithm with suitable example.</p> | (13) | BTL6 | Create |
| 13 | <p>Explain the depth first approach of finding articulation points in a connected graph with necessary algorithm.</p> | (13) | BTL4 | Analyze |
| 14 | <p>i. Write short notes on Bi-connectivity. ii. Express different types of graphs with example.</p> | (7) (6) | BTL2 | Understand |

| PART C | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--|-------------|----------------|-----------|--------|-------------|-----|---|---|-----|---|---|-----|---|---|-----|---|---|-----|---|---|-----|---|---|-----|---|---|-----|---|---|-----|---|---|-----|---|---|
| 1 | Given the adjacency matrix of a graph, write a program to calculate the in-degree and the out-degree of a node N in the graph. | (15) | BTL 4 Analyze | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | Consider five cities: (1) New Delhi, (2) Mumbai, (3) Chennai, (4) Bangalore, and (5) Kolkata, and a list of flights that connect these cities as shown in the following table. Use the given information to construct a graph. | (15) | BTL 6 Create | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Flight No</th> <th>Origin</th> <th>Destination</th> </tr> </thead> <tbody> <tr><td>101</td><td>2</td><td>3</td></tr> <tr><td>102</td><td>3</td><td>2</td></tr> <tr><td>103</td><td>5</td><td>3</td></tr> <tr><td>104</td><td>3</td><td>4</td></tr> <tr><td>105</td><td>2</td><td>5</td></tr> <tr><td>106</td><td>5</td><td>2</td></tr> <tr><td>107</td><td>5</td><td>1</td></tr> <tr><td>108</td><td>1</td><td>4</td></tr> <tr><td>109</td><td>5</td><td>4</td></tr> <tr><td>110</td><td>4</td><td>5</td></tr> </tbody> </table> | | | | Flight No | Origin | Destination | 101 | 2 | 3 | 102 | 3 | 2 | 103 | 5 | 3 | 104 | 3 | 4 | 105 | 2 | 5 | 106 | 5 | 2 | 107 | 5 | 1 | 108 | 1 | 4 | 109 | 5 | 4 | 110 | 4 | 5 |
| Flight No | Origin | Destination | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 101 | 2 | 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 102 | 3 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 103 | 5 | 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 104 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 105 | 2 | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 106 | 5 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 107 | 5 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 108 | 1 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 109 | 5 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 110 | 4 | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | i. How can we efficiently check whether or not a graph is disconnected? ii. Describe an algorithm that uses a brute force approach to find all the articulation points in G in $O(V(V + E))$ time. | (7) (8) | BTL 5 Evaluate | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | i. Given a rooted tree, one desires to find the shortest path from the root to a given node v. Which algorithm would one use to find this shortest path? ii. Write a program to determine whether there is at least one path from the source to the destination. | (7) (8) | BTL 6 Create | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| UNIT V SORTING, SEARCHING AND HASH TECHNIQUES | | | |
|---|--|----------|------------|
| Sorting algorithms: Insertion sort - Selection sort - Shell sort - Bubble sort - Quick sort - Merge sort - Radix sort - Searching: Linear search - Binary Search Hashing: Hash Functions - Separate Chaining - Open Addressing - Rehashing - Extendible Hashing | | | |
| PART - A | | | |
| Q. No | Questions | BT Level | Competence |
| 1 | What are the advantage and disadvantage of separate chaining and linear probing? | BTL-1 | Remember |
| 2 | Define extendible hashing. | BTL-1 | Remember |
| 3 | Give the fastest searching algorithm. | BTL-2 | Understand |
| 4 | What is meant by internal and external sorting? Give any two examples for each type. | BTL-1 | Remember |
| 5 | Describe the complexity of bubble sort. | BTL-1 | Remember |
| 6 | Name the applications of linear and binary search techniques. | BTL-1 | Remember |
| 7 | Give the time complexities of bubble sort and quick sort. | BTL-2 | Understand |
| 8 | Predict the fastest sorting algorithm, justify. | BTL-2 | Understand |

| | | | |
|----|--|-------|------------|
| 9 | Compare internal and external sorting. | BTL-4 | Analyze |
| 10 | Distinguish between linear and binary search technique. | BTL-2 | Understand |
| 11 | Classify the different sorting methods. | BTL-3 | Apply |
| 12 | Develop an algorithm for a quick sort. | BTL-6 | Create |
| 13 | Which hashing technique is best and illustrate with an example? | BTL-3 | Apply |
| 14 | Summarize the open addressing hashing method with an example. | BTL-5 | Evaluate |
| 15 | Point out the advantages of using quick sort. | BTL-4 | Analyze |
| 16 | Compare the working of linear and binary search techniques. | BTL-4 | Analyze |
| 17 | Select the best sorting method out of the following - insertion sort, quick sort and merge sort and give justification. | BTL-5 | Evaluate |
| 18 | Illustrate the time complexity of insertion sort with an example. | BTL-3 | Apply |
| 19 | Identify the advantage of shell sort over insertion sort. | BTL-1 | Remember |
| 20 | Develop a simple algorithm for a linear search. | BTL-6 | Create |

PART B

| | | | |
|----|--|-------|------------|
| 1 | Describe how the divide and conquer technique is implemented in binary search. (13) | BTL-1 | Remember |
| 2 | Describe the algorithm to sort the following array: 77, 33, 44, 11, 88, 22, 66, 55 i. Insertion sort (7) ii. Shell Sort (6) | BTL-1 | Remember |
| 3 | i. List the different types of hashing techniques? (7) ii. Explain them in detail with an Example. (6) | BTL-1 | Remember |
| 4 | i. Interpret 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. (7) ii. Discuss the running time of Divide-and-Conquer Merge sort algorithm. (6) | BTL-2 | Understand |
| 5 | i. Sort the sequence 3, 1, 4, 1, 5, 9, 2, 6, 5 using Insertion sort. (7) ii. Describe the routine for insertion sort. (6) | BTL-2 | Understand |
| 6 | Write an algorithm to sort a set of 'N' numbers using quick sort. Demonstrate the algorithm for the following set of numbers: 88,11,22,44,66,99,32,67,54,10. (13) | BTL-3 | Analyze |
| 7 | Explain the various collision resolution techniques in detail with an example. (13) | BTL-4 | Analyze |
| 8 | Compare the below different Sorting methods and discuss about each method in a very detailed Manner. i. Bucket Sort. (7) ii. Selection Sort. (6) | BTL-4 | Apply |
| 9 | i. Sort the given integers and Explain the intermediate results using shell sort: 35,12,14,9,15,45,32,95,40,5. (7) ii. Write and Explain a C code to sort an integer array. (6) | BTL-5 | Evaluate |
| 10 | i. Create an algorithm to perform a binary Search. (7) ii. Develop an algorithm for Merge sort with an example. (6) | BTL-6 | Create |
| 11 | i. Write short notes on Bubble Sort. (5) ii. Illustrate an algorithm to sort the elements using bubble sort. (8) | BTL-3 | Analyze |
| 12 | Describe the following collision resolution techniques in detail with an example. i. Separate chaining. (7) ii. Rehashing. (6) | BTL-1 | Remember |
| 13 | i. Explain different hashing technique. (5) ii. Explain the rehashing technique with suitable example. (8) | BTL-4 | Apply |
| 14 | Describe the open addressing and chaining methods of collision resolution (13) | BTL-2 | Understand |

| | | | |
|---------------|--|------|-------------------|
| | techniques in hashing. | | |
| PART C | | | |
| 1 | Develop an algorithm to search a number in a given set of numbers using binary search. Develop and algorithm to explain Extendible Hashing. | (15) | BTL-6 Create |
| 2 | Given input {4371, 1323, 6173, 4199, 4344, 9679, 1989} and a hash function $h(x) = x \pmod{10}$, show the resulting i) Open hash table ii) Closed hash table using linear probing iii) Closed hash table using quadratic probing | (15) | BTL-5 Evaluate |
| 3 | Explain an algorithm for Shell Sort and Merge Sort and explain with example. | (15) | BTL-5 Evaluate |
| 4 | Prepare a quick sort algorithm and explain with suitable example Give its worst case, average case and best case time complexities. | (15) | BTL-6 Create |

STUCOR APP