# SRM VALLIAMMAI ENGINEERING COLLEGE

**(An Autonomous Institution)**

SRM Nagar, Kattankulathur – 603 203

**DEPARTMENT OF**
**ELECTRONICS AND INSTRUMENTATION ENGINEERING**

## QUESTION BANK

**VI SEMESTER**

## 1904007–DATA STRUCTURES

**Regulation – 2019**

**Academic Year 2024 – 2025 (EVEN)**

Prepared by

**Dr. R.IssanRaj, Assistant Professor (Sr.G)/EIE**

**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 - applications of lists – Polynomial Manipulation.

**PART -A**

| S.No | Questions | BT level | Competence | COS |
|---|---|---|---|---|
| **1.** | Define Abstract data type. | BTL1 | Remember | CO1 |
| **2.** | Illustrate the advantages of ADT. | BTL 2 | Understand | CO1 |
| **3.** | Give a C routine to deallocate the entire linked list. | BTL 2 | Understand | CO1 |
| **4.** | What are the applications of lists? | BTL1 | Remember | CO1 |
| **5.** | List the major operations in linear data structure. | BTL1 | Remember | CO1 |
| **6.** | Write a Pseudo code to create a list of integers using arrays and to search for an element from the list. | BTL 2 | Understand | CO1 |
| **7.** | Express an algorithms to insert an element from a linked list. | BTL 2 | Understand | CO1 |
| **8.** | Point out the four major operations in linear data structures. | BTL1 | Remember | CO1 |
| **9.** | Compare arrays and linked lists. | | | CO1 |
| **10.** | Write a function to find the position of the given element in the linked list. | BTL 2 | Understand | CO1 |
| **11.** | Summarize the disadvantages of linked list over array? | BTL 2 | Understand | CO1 |
| **12.** | Define linked list. | BTL1 | Remember | CO1 |
| **13.** | What are the advantages of linked list over array? | BTL1 | Remember | CO1 |
| **14.** | Differentiate between linear linked list and circular linked list. | BTL 2 | Understand | CO1 |
| **15.** | Assess the use of Header node in a linked list. | BTL 2 | Understand | CO1 |
| **16.** | Discuss the operations can be done with set ADT? | BTL 2 | Understand | CO1 |
| **17.** | List any three applications of linked list. | BTL1 | Remember | CO1 |
| **18.** | What data structure is used to implement recursion? Why? | BTL1 | Remember | CO1 |
| **19.** | Demonstrate the differences between singly and doubly linked lists. | BTL 2 | Understand | CO1 |
| **20.** | Analyze and write the array representation of a polynomial: $p(x) = 4x^3+6x^2+7x+9$ | BTL 2 | Understand | CO1 |
| **21.** | Explain the term Data structure. | BTL1 | Remember | CO1 |
| **22.** | What do you meant by linear data structure?.Give Examples. | | | CO1 |
| **23.** | State the advantages of abstract data type | BTL1 | Remember | CO1 |
| **24.** | Why is linked list used for polynomial arithmetic? | BTL 2 | Understand | CO1 |

**PART -B**

| S.No | Questions | BT level | Competence | COS |
|---|---|---|---|---|
| **1.** | **(i)** Describe about the classification of Data structures with necessary examples. **(9)**<br>**(ii)** Define the various operations on Data structures. **(4)** | BTL 3 | Apply | CO1 |
| **2.** | **(i)** Recall about the abstract data type. **(4)**<br>**(ii)** Describe the list ADT with examples. **(9)** | BTL 4 | Analyze | CO1 |
| **3.** | **(i)** List the limitations of array based implementation of ADT.**(3)** | BTL 4 | Analyze | CO1 |

| | | | | |
|---|---|---|---|---|
| | **(ii)** Describe the advantages of using linked list implementation with examples. **(10)** | | | |
| **4.** | Demonstrate about the linked list implementation for traversing and searching for a value in a linked list with their algorithm. | BTL 3 | Apply | CO1 |
| **5.** | Explain the steps involved in the following insertion operations in a singly linked list<br>**(i)** Insert the node in the start and end. **(7)**<br>**(ii)** The new node is inserted after a given node. **(6)** | BTL 4 | Analyze | CO1 |
| **6.** | Discuss the steps involved in the following deletion operations in a singly linked list<br>**(i)** Delete the node in the start and end. **(7)**<br>**(ii)** The new node is deleted after a given node. **(6)** | BTL 3 | Apply | CO1 |
| **7.** | Analyze the program to find the maximum and minimum elements from a singly linked list. | BTL 4 | Analyze | CO1 |
| **8.** | Demonstrate about the algorithms to insert and delete elements from a circular linked list. Consider all cases. | BTL 3 | Apply | CO1 |
| **9.** | **(i)** Summarize the basic operations of linked list. **(8)**<br>**(ii)** Describe how to insert an element in circular linked list. **(5)** | BTL 4 | Analyze | CO1 |
| **10.** | **(i)** List the various operations on array? Write a procedure to insert an element in the middle of the array. **(7)**<br>**(ii)** Describe procedure to deleting the last node from a circular linked list. **(6)** | BTL 3 | Apply | CO1 |
| **11.** | Express the program that uses functions to perform the following operation on doubly linked list<br>**(i)** Creation **(5)**<br>**(ii)** Insertion **(4)**<br>**(iii)** Deletion **(4)** | BTL 3 | Apply | CO1 |
| **12.** | Analyze about the insertion and deletion operations in a circularly doubly linked list with suitable ADT's and examples. | BTL 4 | Analyze | CO1 |
| **13.** | **(i)** Evaluate the polynomial representation for $6x^3 + 9x^2 + 7x + 1$ using linked list. Write procedure to add and multiply two polynomial and explain with suitable examples. **(7)**<br>**(ii)** Summarize 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)** | BTL 3 | Apply | CO1 |
| **14.** | **(i)** Prepare a functions to multiply two polynomials using linked list implementation. **(7)**<br>**(ii)** Develop the program in C to delete a node the minimum value from a singly linked list. **(6)** | BTL 4 | Analyze | CO1 |
| **15.** | Demonstrate the addition of Two polynomials using singly linked list with necessary diagram. | BTL 3 | Apply | CO1 |
| **16.** | Summarize about the circularly linked list with necessary diagrams. | BTL 4 | Analyze | CO1 |
| **17.** | (i) Summarize about the linked list and memory management. **(5)** | BTL 4 | Analyze | CO1 |
| | (ii) Briefly discuss about Dynamic Memory Management and also in any one high level language **.** **(8)** | | | |
| | | | | |

| PART -C | | | | |
|---|---|---|---|---|
| **S.No** | **Questions** | **BT level** | **Competence** | **COS** |
| **1.** | Develop the C program for the following linked list operation<br>**(i)** Displaying the content of the linked list **(5)**<br>**(ii)** Counting the number of nodes in a linked list **(5)**<br>**(iii)** Reversing the linked list. **(5)** | BTL 3 | Apply | CO1 |
| **2.** | Create the algorithm to insert a new node and to delete a node from a circular header linked list with necessary examples. | BTL4 | Analyze | CO1 |
| **3.** | Develop the program in linked list to do the following<br>**(i)** To edit the content of a particular node with a given value. **(8)**<br>**(ii)** To merge two linked lists. **(7)** | BTL4 | Analyze | CO1 |
| **4.** | **(i)** Write a function to add two polynomials represented by a linked representation. Apply the function for the following input **(8)**<br>$A = 3x^{14} + 2x^{18} + 1$ and $B = 8x^{12} + 3x^{10} + 3x^{8} + 10x^{6}$<br>**(ii)** Write a function to delete the node 'n' from the given doubly linked list.<br>$p \leftrightarrow q \leftrightarrow r \leftrightarrow n \leftrightarrow s \leftrightarrow t \leftrightarrow z \leftrightarrow$ **(7)** | BTL 3 | Apply | CO1 |
| **5.** | Design the implementation of polynomial operation using singly linked list. | BTL 5 | Evaluate | CO1 |

**UNIT-II**

**LINEAR DATASTRUCTURES - STACKS, QUEUES**

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

## PART -A

| S.No | Questions | BT level | Competence | COS |
|---|---|---|---|---|
| **1.** | Give the applications of stacks? | BTL2 | Understand | CO2 |
| **2.** | Point out the advantage of representing stack using a linked list than array. | BTL1 | Remember | CO2 |
| **3.** | Write an ADT for insertion in stack. | BTL1 | Remember | CO2 |
| **4.** | Develop an algorithm for inserting a new element into the stack. | BTL2 | Understand | CO2 |
| **5.** | Point out the rules followed during the infix to postfix conversions. | BTL2 | Understand | CO2 |
| **6.** | Describe how the following "infix" expression is evaluated with the help of the help of Stack: 5 * (6 + 2) - 12 / 4. | BTL1 | Remember | CO2 |
| **7.** | Write the following infix expression into post fix (A+B)*(C+B)*(E/F). | BTL2 | Understand | CO2 |
| **8.** | Discover the postfix and prefix forms of the expression: A + B* (C – D) / (P – R). | BTL2 | Understand | CO2 |
| **9.** | List the applications of Stack and Queue. | BTL1 | Remember | CO2 |
| **10.** | Compare Stack and an Array. | BTL2 | Understand | CO2 |
| **11.** | Discuss about queue? List its advantages. | BTL2 | Understand | CO2 |
| **12.** | Write a routine to check whether the queue is full or empty. | BTL2 | Understand | CO2 |
| **13.** | What are the applications of queue? | BTL1 | Remember | CO2 |
| **14.** | Write a routine to display the contents of queue. | BTL2 | Understand | CO2 |

| | | | | |
|---|---|---|---|---|
| **15.** | Circular queue is better than standard linear queue, Why? | BTL2 | Understand | CO2 |
| **16.** | What is priority queue? | BTL1 | Remember | CO2 |
| **17.** | Define double ended queue. | BTL1 | Remember | CO2 |
| **18.** | What is a deque? What are the two ways in which a deque can be implemented? | BTL2 | Understand | CO2 |
| **19.** | Compare the working of stack and queue data structure. | BTL1 | Remember | CO2 |
| **20.** | Write an algorithm for deleting an element in a double ended queue. | BTL2 | Understand | CO2 |
| **21.** | What do you understand by polish notation? | BTL2 | Understand | CO2 |
| **22.** | Write any two applications of stack. | BTL1 | Remember | CO2 |
| **23.** | List the characteristics of stacks. | BTL1 | Remember | CO2 |
| **24.** | Write the steps to reverse the contents of the list with the help of stack data structure. | BTL1 | Remember | CO2 |
| | **PART -B** | | | |
| **S.No** | **Questions** | **BT level** | **Competence** | **COS** |
| **1.** | **(i)** Give the ADT operations for array implementations of a stack. **(9)**<br>**(ii)** Summarize about the concept of multiple stacks. **(4)** | BTL 3 | Apply | CO2 |
| **2.** | **(i)** Describe algorithm to display to contents of a stack with an example. **(6)**<br>**(ii)** Recall the algorithms for inserting and deleting values from a stack. **(7)** | BTL 4 | Analyze | CO2 |
| **3.** | **(i)** Write the procedure for stack operations. **(7)**<br>**(ii)** Show that the Recursion is implicit application of STACK ADT with necessary examples. **(6)** | BTL 4 | Analyze | CO2 |
| **4.** | Describe the Tower of Hanoi implementation of stack application with necessary diagram, algorithm and examples. | BTL 4 | Analyze | CO2 |
| **5.** | Write the following operation in stack with its program<br>**(i)** Push<br>**(ii)** Pop<br>**(iii)** To display Stack top element<br>**(iv)** To swap the top two element **(3+3+3+4)** | BTL 3 | Apply | CO2 |
| **6.** | Analyze the algorithms to check if the given parenthesized arithmetic expression contains balanced parenthesis and to convert such expression to postfix form and evaluate it. Illustrate with example. | BTL 4 | Analyze | CO2 |
| **7.** | **(i)** Assess the procedure to convert the infix expression to post fix expression and steps involved in evaluating the postfix expression. **(7)**<br>**(ii)** Deduce the expression A-(B/C+(D%E*F)/G)*H to postfix form and also Evaluate the given post fix expression 934*8+4/-. **(6)** | BTL 3 | Apply | CO2 |
| **8.** | **(i)** Demonstrate about the conversion of Infix to post fix expression in stack. **(5)**<br>**(ii)** Illustrate the ADT operations for array implementations of a queue **(8)** | BTL 3 | Apply | CO2 |
| **9.** | **(i)** Describe the ADT operations for a linear queue using linked list implementation. **(9)** | BTL 4 | Analyze | CO2 |

| | | | | |
|---|---|---|---|---|
| | **(ii)** List the application of stacks. **(4)** | | | |
| **10.** | **(i)** Describe the operations of queue with example. **(7)**<br>**(ii)** Develop algorithms for inserting and deleting values from a queue **(6)** | BTL 3 | Apply | CO2 |
| **11.** | **(i)** Point out the advantages of circular queue over linear queue. Write the functions for Insertion in circular queue. **(7)**<br>**(ii)** Explain how to delete an element in a circular queue. **(6)** | BTL 4 | Analyze | CO2 |
| **12.** | Write a program to implement all the queue operations using array implementation of ADT. | BTL 3 | Apply | CO2 |
| **13.** | Summarize about the priority queue and discuss about the linked representation and linked list representation of priority queue with example. | BTL 4 | Analyze | CO2 |
| **14.** | Explain about the following with necessary examples<br>**(i)** Deques **(7)**<br>**(ii)** Multiple queues **(6)** | BTL 4 | Analyze | CO2 |
| **15.** | Summarize about the stack ADT and primitive stack operations with necessary example. | BTL 4 | Analyze | CO2 |
| **16.** | Create a program to perform the conversion of infix expression to post fix expression. | BTL 3 | Apply | CO2 |
| **17.** | Write an algorithm to convert a postfix expression into an infix expression. Consider following arithmetic expression in postfix notation:<br>752+*415-/-<br>**(i)** Find the value of the expression **(7)**<br>**(ii)** Find the equivalent prefix form of above expression. **(6)** | BTL 3 | Apply | CO2 |

## PART C

| S.No | Questions | BT level | Competence | COS |
|---|---|---|---|---|
| **1.** | **(i)** Develop a program to perform Push, Pop, and Peek operation on a stack. **(11)**<br>**(ii)** Develop a program to calculate the factorial of a given number using recursive function. **(4)** | BTL 4 | Analyze | CO2 |
| **2.** | Design a program to implement all the stack operations using array implementation of ADT. | BTL 3 | Apply | CO2 |
| **3.** | Develop a program to implement all the stack operations using linked list implementation of ADT. | BTL 3 | Apply | CO2 |
| **4.** | Design a program to implement all the queue operations using linked implementation of ADT | BTL 4 | Analyze | CO2 |
| **5.** | Develop the program for the implementation of circular queue using arrays. | BTL 3 | Apply | CO2 |

**UNIT-III**

**NON LINEAR DATA STRUCTURES TREES- GRAPHS.**

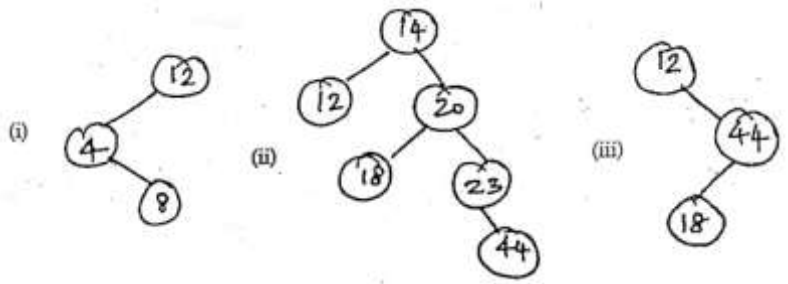
Binary Trees – Binary tree representation and traversals – Application of trees: – Graph and its representations – Graph Traversals – Connected components.

## PART A

| S.No | Questions | BT level | Competence | COS |
|---|---|---|---|---|
| **1.** | What is a threaded binary tree? | BTL1 | Remember | CO3 |
| **2.** | List few applications of tree. | BTL1 | Remember | CO3 |
| **3.** | Write the result of inserting 3,1,4,6,2,8,9 into an initially empty AVL tree. | BTL 2 | Understand | CO3 |
| **4.** | Deduce the need for tree representation? | BTL 2 | Understand | CO3 |
| **5.** | How do you calculate the depth of a B-tree? | BTL 2 | Understand | CO3 |
| **6.** | Can you define tree in terms of graph? Analyze. | BTL1 | Remember | CO3 |
| **7.** | What do you mean by splay tree? | | | CO3 |
| **8.** | Give an example for expression tree. | BTL1 | Remember | CO3 |
| **9.** | Illustrate the height balanced tree:"AVL". | BTL 2 | Understand | CO3 |
| **10.** | Write the expression tree for ((b+c)*a)+((d+e*f)+g). | BTL 2 | Understand | CO3 |
| **11.** | Infer the advantages of threaded binary tree? | BTL 2 | Understand | CO3 |
| **12.** | List the operations performed in splay trees? | BTL1 | Remember | CO3 |
| **13.** | Differentiate strongly connected and weakly connected graph | BTL 2 | Understand | CO3 |
| **14.** | How do you represent a graph using linked list? Give example. | BTL 2 | Understand | CO3 |
| **15.** | Illustrate the algorithm for pre-order traversal? | BTL 2 | Understand | CO3 |
| **16.** | Define in degree of Graph. | BTL1 | Remember | CO3 |
| **17.** | Interpret about undirected graph. | BTL 2 | Understand | CO3 |
| **18.** | Write the two applications of graph. | BTL1 | Remember | CO3 |
| **19.** | The depth of complete binary tree is 8 and evaluate the number of nodes in leaf. | BTL 2 | Understand | CO3 |
| **20.** | Demonstrate about connected graph. | BTL 2 | Understand | CO3 |
| **21.** | Define complete binary tree? | BTL1 | Remember | CO3 |
| **22.** | List the applications of trees. | BTL1 | Remember | CO3 |
| **23.** | Illustrate about graph traversal | BTL 2 | Understand | CO3 |
| **24.** | Explain Connected components in Data structure | BTL1 | Remember | CO3 |

## PART B

| S.No | Questions | BT level | Competence | COS |
|---|---|---|---|---|
| **1.** | **(i)** Summarize the array representation of a binary tree with suitable example. **(7)**<br>**(ii)** Give the program to delete all the leaf nodes of a binary tree. **(6)** | BTL 3 | Apply | CO3 |
| **2.** | **(i)** Describe about tournament trees with an example. **(5)**<br>**(ii)** Examine how binary tree is created from a general tree. **(8)** | BTL 4 | Analyze | CO3 |
| **3.** | Demonstrate the various tree traversal of a binary tree with algorithm and examples. | BTL 4 | Analyze | CO3 |
| **4.** | Estimate the AVL tree for the following after rotation. **(4+6+3)** | BTL 3 | Apply | CO3 |

| | | | | |
|---|---|---|---|---|
| |  | | | |
| **5.** | Deduce the B tree to insert the following key elements (consider order of the B tree is 3).<br>55,4,44,3,6,7,9,45,46,56,57 | BTL 3 | Apply | CO3 |
| **6.** | **(i)** Explain how to implement binary search tree. **(9)**<br>**(ii)** Infer about threaded binary tree? Explain its use. **(4)** | BTL 4 | Analyze | CO3 |
| **7.** | **(i)** Describe how deletion can take place in AVL trees with suitable algorithms. **(8)**<br>**(ii)** Recall about Huffman's tree with an example. **(5)** | BTL 4 | Analyze | CO3 |
| **8.** | Analyze about the insertion and deletion operations of B-tree with simulate abstract data types. | BTL 4 | Analyze | CO3 |
| **9.** | Define the following graph terminology<br>**(i)** Adjacent node<br>**(ii)** Degree of a node<br>**(iii)** Regular graph<br>**(iv)** Connected graph<br>**(v)** Complete graph<br>**(vi)** Weighted graph **(2+2+2+2+2+3)** | BTL 4 | Analyze | CO3 |
| **10.** | Explain the following representation of graph<br>**(i)** Adjacency Matrix representation **(7)**<br>**(ii)** Adjacency list representation **(6)** | BTL 4 | Analyze | CO3 |
| **11.** | **(i)** Describe about the Breadth –First Search algorithm for graph traversal with an example and give its features. **(7)**<br>**(ii)** Describe about the Adjacency Multi list representation of graph with an example. **(6)** | BTL 3 | Apply | CO3 |
| **12.** | **(i)** Demonstrate Pseudo code for prim's algorithm. Also give an example to construct a minimum spanning tree. **(9)**<br>**(ii)** Illustrate the applications of tree. **(4)** | BTL 3 | Apply | CO3 |
| **13.** | Summarize the Depth –First Search algorithm for graph traversal with an example and give its applications. | | | CO3 |
| **14.** | **(i)** Develop a program to find an Euler circuit in a graph. Trace the algorithm with example. **(5)**<br>**(ii)** Write the Pseudo code for Dijkstra's shortest path algorithm. Give suitable example to trace the algorithm. **(8)** | BTL 3 | Apply | CO3 |
| **15.** | Write the B tree index for the following data<br>12,24,13,5,7,65,54,32,7 | BTL 4 | Analyze | CO3 |
| **16.** | Write the program for Depth –First Search algorithm for graph traversal. | BTL 4 | Analyze | CO3 |
| **17.** | Demonstrate about the tree traversal with necessary examples | BTL 3 | Apply | CO3 |
| | | | | |

| PART C | | | | |
|---|---|---|---|---|
| **S.No** | **Questions** | **BT level** | **Competence** | **COS** |
| **1.** | **(i)** Create a binary search tree using the following data elements:<br>45,39,56,12,34,78,32,10,89,54,67,81 **(8)**<br>**(ii)** Develop the algorithm to search for a given value in a binary search tree. **(7)** | BTL 3 | Apply | CO3 |
| **2.** | Develop a program to implement Binary Tree and perform in order, pre order and post order traversals. | BTL 4 | Analyze | CO3 |
| **3.** | Design B tree to insert the following key elements (consider order of the B tree is 3).<br>37,7,24,3,5,7,8,35,77,56,61 | BTL 3 | Apply | CO3 |
| **4.** | Apply Kruskal's algorithm on the graph if<br>F={{A},{B},{C},{D},{E},{F}}<br>MST={}<br>Q={(A,D),(E,F),(C,E),(E,D),(C,D),(D,F)(A,C),(A,B),(B,C)} | BTL 3 | Apply | CO3 |
| **5.** | Develop the program to demonstrate the tree concept. | BTL 4 | Analyze | CO3 |

| **UNIT-IV**<br>**SORTING**<br>Selection sort-Insertion sort – Merge sort – Quick sort – Heap sort – Bubble sort- Shell sort – Radix sort. | | | | |
|---|---|---|---|---|
| **PART A** | | | | |
| **S.No** | **Questions** | **BT level** | **Competence** | **COS** |
| **1.** | Classify the different sorting methods. | BTL1 | Remember | CO4 |
| **2.** | What is topological sorting? | BTL 2 | Understand | CO4 |
| **3.** | Demonstrate how do you do a selection sort? | BTL 2 | Understand | CO4 |
| **4.** | Illustrate the time complexity of insertion sort with an example. | BTL 2 | Understand | CO4 |
| **5.** | Discuss about Selection sort algorithm. | BTL 2 | Understand | CO4 |
| **6.** | Give the best sorting method out of the following - insertion sort, quick sort and merge sort and give justification. | BTL1 | Remember | CO4 |
| **7.** | What is merge sort with example? | BTL 2 | Understand | CO4 |
| **8.** | Define Merge sort. | BTL1 | Remember | CO4 |
| **9.** | Write an algorithm for a quick sort. | BTL 2 | Understand | CO4 |
| **10.** | Point out the advantages of using quick sort. | BTL1 | Remember | CO4 |
| **11.** | How many type of heap are there? | | | CO4 |
| **12.** | Write about heap sort with example | BTL1 | Remember | CO4 |
| **13.** | How do you solve a bubble sort problem? | BTL 2 | Understand | CO4 |
| **14.** | Summarize about the time complexities of bubble sort and quick sort. | BTL 2 | Understand | CO4 |
| **15.** | Describe the complexity of bubble sort. | BTL1 | Remember | CO4 |
| **16.** | Illustrate the basic idea of shell sort? | BTL 2 | Understand | CO4 |
| **17.** | What is the other name of shell sort? Why called so? | BTL1 | Remember | CO4 |
| **18.** | Assess the advantage of shell sort over insertion sort. | BTL 2 | Understand | CO4 |
| **19.** | What is meant by Radix sort give example? | BTL1 | Remember | CO4 |
| **20.** | Write the fastest sorting algorithm, justify. | BTL 2 | Understand | CO4 |
| **21.** | Name the slowest and fastest sorting technique | BTL1 | Remember | CO4 |
| **22.** | What are the two stages in which heap sort is conducted? | BTL1 | Remember | CO4 |
| **23.** | Explain why binary search can not be performed using linked | BTL 2 | Understand | CO4 |

| | | | | |
|---|---|---|---|---|
| | list? | | | |
| **24.** | List the sorting technique which uses logarithmic time complexity | BTL1 | Remember | CO4 |
| **PART B** | | | | |
| **S.No** | **Questions** | **BT level** | **Competence** | **COS** |
| **1.** | Summarize about the selection sort, its algorithm and also explain with an example. | BTL 3 | Apply | CO4 |
| **2.** | **(i)** Develop the sorting steps for the array given below using Selection sort<br>24,12,5,11,76,2,45,13 **(5)**<br>**(ii)** Develop the program to sort an array using selection sort algorithm. **(8)** | BTL 3 | Apply | CO4 |
| **3.** | **(i)** Give the program to sort an array using insertion sort algorithm. **(9)**<br>**(ii)** Discuss the advantages of Insertion sort. **(4)** | BTL 4 | Analyze | CO4 |
| **4.** | With an algorithm and example describe about the insertion sort. | BTL 4 | Analyze | CO4 |
| **5.** | **(i)** Infer a function to perform merge sort .Give example. **(6)**<br>**(ii)** Explain a routine for Insertion sort. Sort the following sequence using Insertion sort. **(7)**<br>3,10,4,2,8,6,5,1 | BTL 3 | Apply | CO4 |
| **6.** | Discriminate about the Merge sort, its algorithm and also explain with an example. | BTL 4 | Analyze | CO4 |
| **7.** | **(i)** Apply the merge sort algorithm to sort the following numbers. 4,7,2,9,1,8,3,15,12,6,14. **(7)**<br>**(ii)** Illustrate the quick sort algorithm. **(6)** | BTL 3 | Apply | CO4 |
| **8.** | Describe about the quick sort, its algorithm and also explain with an example. | BTL 4 | Analyze | CO4 |
| **9.** | **(i)** Sort the following numbers using the quick sort algorithm.<br>35,11,39,17,24,46 **(8)**<br>**(ii)** Point out the complexity of quick sort algorithm and also give its advantages and disadvantages. **(5)** | BTL 3 | Apply | CO4 |
| **10.** | Explain in detail about heap sort algorithm with an example. | BTL 4 | Analyze | CO4 |
| **11.** | Describe about the Bubble sort, its algorithm and also explain with an example. | BTL 4 | Analyze | CO4 |
| **12.** | **(i)** Apply Bubble sort algorithm to sort the following numbers.<br>4,7,2,9,1,8,3,15,12,6,14. **(6)**<br>**(ii)** Apply Shell sort algorithm to sort the following numbers.<br>63,19,7,90,81,36,54,45,72,27,22,9,41,59,33 **(7)** | BTL 3 | Apply | CO4 |
| **13.** | State and explain the shell sort. State and explain the algorithm for shell sort. Sort the elements using shell sort. | BTL 4 | Analyze | CO4 |
| **14.** | Summarize about the Radix sort, its algorithm and also explain with an example. | BTL 4 | Analyze | CO4 |
| **15.** | Give the program for performing implementation of insertion sort | BTL 4 | Analyze | CO4 |
| **16.** | Construct the algorithm and program for selection sort. | BTL 3 | Apply | CO4 |
| **17.** | Write the program for sorting the elements by radix sort. | BTL 4 | Analyze | CO4 |

| PART C | | | | |
|---|---|---|---|---|
| **S.No** | **Questions** | **BT level** | **Competence** | **COS** |
| **1.** | **(i)** Develop a program to enter 'n' numbers in an array. Redisplay the array with elements being sorted in ascending order. **(10)**<br>**(ii)** Develop the algorithm for bubble sort. **(5)** | BTL 3 | Apply | CO4 |
| **2.** | **(i)** Develop the program to sort an array using insertion algorithm. **(9)**<br>**(ii)** Sort the following numbers using the selection sort algorithm. 54,45,72,27,22,9,41, 63,19,7,90 **(6)** | BTL 4 | Analyze | CO4 |
| **3.** | Design the quick sort procedure to sort out the following elements and also give its algorithm 7,90,81,36,544,7,2,9,1,8 | BTL 3 | Apply | CO4 |
| **4.** | Design the radix sort procedure to sort out the following data and also give its algorithm 343,651,924,123,568,473,554,809,912. | BTL 4 | Analyze | CO4 |
| **5.** | Sort the following list of numbers using bubble sort technique 52,1,27,85,66,23,13,57 | BTL 4 | Analyze | CO4 |

**UNIT-V**
**SEARCHING AND INDEXING**
Linear Search–Binary Search-Hash tables–Overflow handling– Hash Index – B-Tree Indexing

| PART A | | | | |
|---|---|---|---|---|
| **S.No** | **Questions** | **BT level** | **Competence** | **COS** |
| **1.** | Discuss about linear search. | BTL 2 | Understand | CO5 |
| **2.** | Define binary search. | BTL1 | Remember | CO5 |
| **3.** | State the advantages of collision resolution strategies. | BTL1 | Remember | CO5 |
| **4.** | Analyze the average case complexity of the linear search algorithm. | BTL 2 | Understand | CO5 |
| **5.** | What are the advantage and disadvantage of separate chaining and linear probing? | BTL1 | Remember | CO5 |
| **6.** | Compare linear search and binary search. | BTL 2 | Understand | CO5 |
| **7.** | Illustrate the complexity of binary search. | BTL 2 | Understand | CO5 |
| **8.** | Write a simple algorithm for a linear search. | BTL 2 | Understand | CO5 |
| **9.** | Give the fastest searching algorithm. | BTL1 | Remember | CO5 |
| **10.** | Define hash function. | BTL1 | Remember | CO5 |
| **11.** | Analyze how does the division remainder method help in hashing method? | BTL 2 | Understand | CO5 |
| **12.** | Demonstrate the basic difference between static hashing and dynamic hashing? | BTL 2 | Understand | CO5 |
| **13.** | Define hash table? | BTL1 | Remember | CO5 |
| **14.** | Brief about Extendible hashing. | BTL 2 | Understand | CO5 |
| **15.** | Summarize briefly about Rehashing? | BTL1 | Remember | CO5 |
| **16.** | List out the different types of hashing functions? | BTL1 | Remember | CO5 |
| **17.** | Illustrate about overflow handling? | BTL 2 | Understand | CO5 |

| | | | | |
|---|---|---|---|---|
| **18.** | Summarize about Hash index. | BTL 2 | Understand | CO5 |
| **19.** | What is meant by B-Tree Indexing | BTL1 | Remember | CO5 |
| **20.** | Write the Pseudo code for fastest searching algorithm. | BTL 2 | Understand | CO5 |
| **21.** | What is overflow in hashing? | BTL1 | Remember | CO5 |
| **22.** | Interpret uniform hash function. | BTL 2 | Understand | CO5 |
| **23.** | Write the major problem in linear probing. | BTL1 | Remember | CO5 |
| **24.** | Point out the advantages of B tree indices. | BTL1 | Remember | CO5 |
| PART B | | | | |
| **S.No** | **Questions** | **BT level** | **Competence** | **COS** |
| **1.** | Summarize about the linear search with algorithm and an example. | BTL 4 | Analyze | CO5 |
| **2.** | With an algorithm describe about the Binary search and also examine with an example. | BTL 3 | Apply | CO5 |
| **3.** | Develop the program to demonstrate the binary search. | BTL 3 | Apply | CO5 |
| **4.** | **(i)** Differentiate linear search and binary search. State and explain the algorithms for both the search with example. **(7)**<br>**(ii)** Explain Rehashing and extendible hashing. **(6)** | BTL 4 | Analyze | CO5 |
| **5.** | **(i)** Examine the collision resolution methods in hashing. **(7)**<br>**(ii)** Describe about different types of hash functions. **(6)** | BTL 4 | Analyze | CO5 |
| **6.** | Deduce extendible hash structure to insert the following key elements<br>2,3,5,7,11,17,19,23,29,31. | BTL 3 | Apply | CO5 |
| **7.** | **(i)** When do you perform rehashing? Illustrate with example. **(8)**<br>**(ii)** Summarize about the real world applications of hashing **(5)** | BTL 4 | Analyze | CO5 |
| **8.** | Consider a hash table with 9 slots. The hash function is h(k)=k mod 9. The following keys are inserted in the order 5, 28, 19, 15, 20, 33, 12, 17, 10. Draw the contents of the hash table when the collisions are resolved by<br>**(i)** Chaining **(4)**<br>**(ii)** Linear probing **(4)**<br>**(iii)** Double hashing .The second hash function h2(x)=7-(x mod 7). **(5)** | BTL 3 | Apply | CO5 |
| **9.** | Given input {4371,1323,6173,4199,4344,9679,1989} and a hash function h(x)=x (mod10 ).Estimate the resulting<br>**(i)** Open hash table **(4)**<br>**(ii)** Closed hash table using linear probing **(4)**<br>**(iii)** Closed hash table using quadratic probing **(5)** | BTL 3 | Apply | CO5 |
| **10.** | Describe about the overflow handling in data structure with necessary example. | BTL 4 | Analyze | CO5 |
| **11.** | **(i)** What is meant by collision resolution by chaining? **(4)**<br>**(ii)** Describe about the operations on a chained hash table. **(9)** | BTL 4 | Analyze | CO5 |
| **12.** | Demonstrate about the different types of indexing used in data structure. | BTL 3 | Apply | CO5 |
| **13.** | Summarize about the following<br>**(i)** B tree indices **(7)**<br>**(ii)** Hashed Indices **(6)** | BTL 4 | Analyze | CO5 |
| **14.** | Explain about the B tree indices algorithm with an example. | BTL 3 | Apply | CO5 |

| | | | | |
|---|---|---|---|---|
| **15.** | Demonstrate about the different types of hash function with necessary examples | | | CO5 |
| **16.** | Summarize about the different collision resolution strategies with examples | BTL 4 | Analyze | CO5 |
| **17.** | Describe about the Extendible Hashing with necessary examples. | BTL 4 | Analyze | CO5 |
| | **PART C** | | | |
| **S.No** | **Questions** | **BT level** | **Competence** | **COS** |
| **1.** | Develop a program to demonstrate linear search and binary search with algorithm and an example. | BTL 4 | Analyze | CO5 |
| **2.** | Consider a hash table of size =10.Using double hashing, insert the keys 72,27,36,24,63,81,92 and 101 into the table. Take $h_1$=(k mod 10) $h_2$=(k mod 18). | BTL 4 | Analyze | CO5 |
| **3.** | Design the code to initialize, insert, delete and search a value in a chained hash table. | BTL 3 | Apply | CO5 |
| **4.** | Develop the algorithm and also program to perform the B tree indices. | BTL 4 | Analyze | CO5 |
| **5.** | Develop the program for the implementation of hash table and collision handling by linear probing | BTL 3 | Apply | CO5 |