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 2022 – 2023 (EVEN)

Prepared by

Mr. R.IssanRaj, Asst. Prof. (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
1.	Define Abstract data type.	BTL1	Remember
2.	Illustrate the advantages of ADT.	BTL 3	Apply
3.	Give a C routine to deallocate the entire linked list.	BTL 2	Understand
4.	What are the applications of lists?	BTL1	Remember
5.	List the major operations in linear data structure.	BTL1	Remember
6.	Develop a Pseudo code to create a list of integers using arrays and to search for an element from the list.	BTL6	Create
7.	Express an algorithms to insert an element from a linked list.	BTL 2	Understand
8.	Point out the four major operations in linear data structures.	BTL 4	Analyze
9.	Compare arrays and linked lists.	BTL 5	Evaluate
10.	Design a function to find the position of the given element in the linked list.	BTL6	Create
11.	Summarize the disadvantages of linked list over array?	BTL 3	Apply
12.	Define linked list.	BTL1	Remember
13.	What are the advantages of linked list over array?	BTL1	Remember
14.	Differentiate between linear linked list and circular linked list.	BTL 4	Analyze
15.	Assess the use of Header node in a linked list.	BTL 5	Evaluate
16.	Discuss the operations can be done with set ADT?	BTL 2	Understand
17.	List any three applications of linked list.	BTL1	Remember
18.	What data structure is used to implement recursion? Why?	BTL 2	Understand
19.	Demonstrate the differences between singly and doubly linked lists.	BTL 3	Apply
20.	Analyze and write the array representation of a polynomial: $p(x) = 4x^3+6x^2+7x+9$	BTL4	Analyze
21.	Explain the term Data structure.	BTL1	Remember
22.	What do you meant by linear data structure?.Give Examples.	BTL 2	Understand
23.	State the advantages of abstract data type	BTL 3	Apply
24.	Why is linked list used for polynomial arithmetic?	BTL4	Analyze

PART -B

S.No	Questions	BT level	Competence
1.	**(i)** Describe about the classification of Data structures with necessary examples. **(9)** **(ii)** Define the various operations on Data structures. **(4)**	BTL1	Remember
2.	**(i)** Recall about the abstract data type. **(4)** **(ii)** Describe the list ADT with examples. **(9)**	BTL1	Remember
3.	**(i)** List the limitations of array based implementation of ADT. **(3)** **(ii)** Describe the advantages of using linked list implementation with	BTL1	Remember

	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
5.	Explain the steps involved in the following insertion operations in a singly linked list **(i)** Insert the node in the start and end. **(7)** **(ii)** The new node is inserted after a given node. **(6)**	BTL 4	Analyze
6.	Discuss the steps involved in the following deletion operations in a singly linked list **(i)** Delete the node in the start and end. **(7)** **(ii)** The new node is deleted after a given node. **(6)**	BTL 2	Understand
7.	Analyze the program to find the maximum and minimum elements from a singly linked list.	BTL 4	Analyze
8.	Demonstrate about the algorithms to insert and delete elements from a circular linked list. Consider all cases.	BTL 3	Apply
9.	**(i)** Summarize the basic operations of linked list. **(8)** **(ii)** Describe how to insert an element in circular linked list. **(5)**	BTL 2	Understand
10.	**(i)** List the various operations on array? Write a procedure to insert an element in the middle of the array. **(7)** **(ii)** Describe procedure to deleting the last node from a circular linked list. **(6)**	BTL1	Remember
11.	Express the program that uses functions to perform the following operation on doubly linked list **(i)** Creation **(5)** **(ii)** Insertion **(4)** **(iii)** Deletion **(4)**	BTL 2	Understand
12.	Analyze about the insertion and deletion operations in a circularly doubly linked list with suitable ADT's and examples.	BTL 4	Analyze
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)** **(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 5	Evaluate
14.	**(i)** Prepare a functions to multiply two polynomials using linked list implementation. **(7)** **(ii)** Develop the program in C to delete a node the minimum value from a singly linked list. **(6)**	BTL 6	Create
15.	Demonstrate the addition of Two polynomials using singly linked list with necessary diagram.	BTL 3	Apply
16.	Summarize about the circularly linked list with necessary diagrams.	BTL 2	Understand
17.	(i) Summarize about the linked list and memory management. **(5)** (ii) Briefly discuss about Dynamic Memory Management and also in any one high level language **.** **(8)**	BTL 3	Apply

PART -C

S.No	**Questions**	**BT level**	**Competence**
1.	Develop the C program for the following linked list operation **(i)** Displaying the content of the linked list **(5)**	BTL 6	Create

	(ii) Counting the number of nodes in a linked list **(5)** **(iii)** Reversing the linked list. **(5)**		
2.	Create the algorithm to insert a new node and to delete a node from a circular header linked list with necessary examples.	BTL 6	Create
3.	Develop the program in linked list to do the following **(i)** To edit the content of a particular node with a given value. **(8)** **(ii)** To merge two linked lists. **(7)**	BTL 6	Create
4.	**(i)** Write a function to add two polynomials represented by a linked representation. Apply the function for the following input. **(8)** $A = 3x^{14} + 2x^{18} + 1$ and $B = 8x^{12} + 3x^{10} + 3x^{8} + 10x^{6}$ **(ii)** Write a function to delete the node 'n' from the given doubly linked list. $p \leftrightarrow q \leftrightarrow r \leftrightarrow n \leftrightarrow s \leftrightarrow t \leftrightarrow z \leftrightarrow$ **(7)**	BTL 6	Create
5.	Design the implementation of polynomial operation using singly linked list.	BTL 6	Create

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
1.	Give the applications of stacks?	BTL2	Understand
2.	Point out the advantage of representing stack using a linked list than array.	BTL4	Analyze
3.	Write an ADT for insertion in stack.	BTL1	Remember
4.	Develop an algorithm for inserting a new element into the stack.	BTL6	Create
5.	Point out the rules followed during the infix to postfix conversions.	BTL4	Analyze
6.	Describe how the following "infix" expression is evaluated with the help of the help of Stack: 5 * (6 + 2) - 12 / 4.	BTL2	Understand
7.	Solve the following infix expression into post fix (A+B)*(C+B)*(E/F).	BTL 3	Apply
8.	Discover the postfix and prefix forms of the expression: A + B* (C – D) / (P – R).	BTL 3	Apply
9.	List the applications of Stack and Queue.	BTL1	Remember
10.	Compare Stack and an Array.	BTL 5	Evaluate
11.	Discuss about queue? List its advantages.	BTL2	Understand
12.	Analyze and write a routine to check whether the queue is full or empty.	BTL4	Analyze
13.	What are the applications of queue?	BTL1	Remember
14.	Write a routine to display the contents of queue.	BTL3	Apply
15.	Circular queue is better than standard linear queue, Why?	BTL2	Understand
16.	What is priority queue?	BTL1	Remember
17.	Define double ended queue.	BTL1	Remember
18.	What is a deque? What are the two ways in which a deque can be implemented?	BTL1	Remember
19.	Compare the working of stack and queue data structure.	BTL5	Evaluate
20.	Develop an algorithm for deleting an element in a double ended queue.	BTL6	Create

21.	What do you understand by polish notation?	BTL4	Analyze
22.	Write any two applications of stack.	BTL1	Remember
23.	List the characteristics of stacks.	BTL3	Apply
24.	Write the steps to reverse the contents of the list with the help of stack data structure.	BTL2	Understand

PART -B

S.No	Questions	BT level	Competence
1.	**(i)** Give the ADT operations for array implementations of a stack. **(9)** **(ii)** Summarize about the concept of multiple stacks. **(4)**	BTL 2	Understand
2.	**(i)** Describe algorithm to display to contents of a stack with an example. **(6)** **(ii)** Recall the algorithms for inserting and deleting values from a stack. **(7)**	BTL1	Remember
3.	**(i)** Write the procedure for stack operations. **(7)** **(ii)** Show that the Recursion is implicit application of STACK ADT with necessary examples. **(6)**	BTL 2	Understand
4.	Describe the Tower of Hanoi implementation of stack application with necessary diagram, algorithm and examples.	BTL1	Remember
5.	Solve the following operation in stack with its program **(i)** Push **(ii)** Pop **(iii)** To display Stack top element **(iv)** To swap the top two element **(3+3+3+4)**	BTL 3	Apply
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
7.	**(i)** Assess the procedure to convert the infix expression to post fix expression and steps involved in evaluating the postfix expression. **(7)** **(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 5	Evaluate
8.	**(i)** Demonstrate about the conversion of Infix to post fix expression in stack. **(5)** **(ii)** Illustrate the ADT operations for array implementations of a queue. **(8)**	BTL 3	Apply
9.	**(i)** Describe the ADT operations for a linear queue using linked list implementation. **(9)** **(ii)** List the application of stacks. **(4)**	BTL1	Remember
10.	**(i)** Describe the operations of queue with example. **(7)** **(ii)** Develop algorithms for inserting and deleting values from a queue. **(6)**	BTL1	Remember
11.	**(i)** Point out the advantages of circular queue over linear queue .Write the functions for Insertion in circular queue. **(7)** **(ii)** Explain how to delete an element in a circular queue. **(6)**	BTL 4	Analyze
12.	Design a program to implement all the queue operations using array implementation of ADT.	BTL 6	Create
13.	Summarize about the priority queue and discuss about the linked representation and linked list representation of priority queue with example.	BTL 2	Understand

14.	Explain about the following with necessary examples **(i)** Deques **(7)** **(ii)** Multiple queues **(6)**	BTL 4	Analyze
15.	Summarize about the stack ADT and primitive stack operations with necessary example.	BTL 2	Understand
16.	Create a program to perform the conversion of infix expression to post fix expression.	BTL 3	Apply
17.	Write an algorithm to convert a postfix expression into an infix expression. Consider following arithmetic expression in postfix notation: 752+*415-/- (i) Find the value of the expression **(7)** (ii) Find the equivalent prefix form of above expression. **(6)**	BTL 3	Apply

PART C

S.No	Questions	BT level	Competence
1.	**(i)** Develop a program to perform Push, Pop, and Peek operation on a stack. **(11)** **(ii)** Develop a program to calculate the factorial of a given number using recursive function. **(4)**	BTL 6	Create
2.	Design a program to implement all the stack operations using array implementation of ADT.	BTL 6	Create
3.	Develop a program to implement all the stack operations using linked list implementation of ADT.	BTL 6	Create
4.	Design a program to implement all the queue operations using linked implementation of ADT	BTL 6	Create
5.	Develop the program for the implementation of circular queue using arrays.	BTL 6	Create

UNIT-III
NON LINEAR DATA STRUCTURES TREES- GRAPHS.

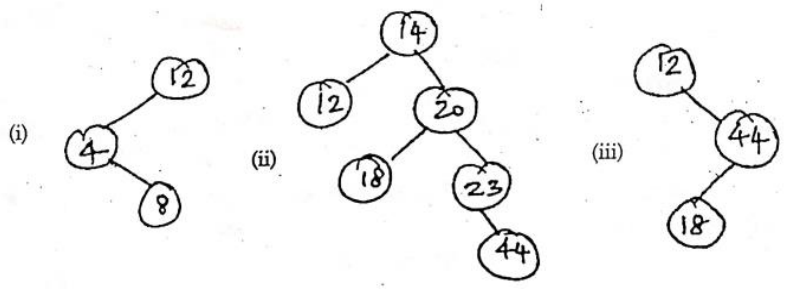
Binary Trees – Binary tree representation and traversals – Application of trees: – Graph and its representations – Graph Traversals – Connected components.

PART A

S.No	Questions	BT level	Competence
1.	What is a threaded binary tree?	BTL1	Remember
2.	List few applications of tree.	BTL1	Remember
3.	Simulate the result of inserting 3,1,4,6,2,8,9 into an initially empty AVL tree.	BTL6	Create
4.	Deduce the need for tree representation?	BTL 5	Evaluate
5.	How do you calculate the depth of a B-tree?	BTL 2	Understand
6.	Can you define tree in terms of graph? Analyze.	BTL 4	Analyze
7.	What do you mean by splay tree?	BTL1	Remember
8.	Give an example for expression tree.	BTL 2	Understand
9.	Illustrate the height balanced tree:“AVL”.	BTL 3	Apply
10.	Design the expression tree for ((b+c)*a)+((d+e*f)+g).	BTL6	Create
11.	Infer the advantages of threaded binary tree?	BTL 4	Analyze

12.	List the operations performed in splay trees?	BTL1	Remember
13.	Differentiate strongly connected and weakly connected graph	BTL 4	Analyze
14.	How do you represent a graph using linked list? Give example.	BTL 2	Understand
15.	Illustrate the algorithm for pre-order traversal?	BTL 3	Apply
16.	Define in degree of Graph.	BTL1	Remember
17.	Interpret about undirected graph.	BTL 2	Understand
18.	Write the two applications of graph.	BTL1	Remember
19.	The depth of complete binary tree is 8 and evaluate the number of nodes in leaf.	BTL 5	Evaluate
20.	Demonstrate about connected graph.	BTL 3	Apply
21.	Define complete binary tree?	BTL1	Remember
22.	List the applications of trees.	BTL 2	Understand
23.	Illustrate about graph traversal	BTL 3	Apply
24.	Explain Connected components in Data structure	BTL 4	Analyze

PART B

S.No	Questions	BT level	Competence
1.	**(i)** Summarize the array representation of a binary tree with suitable example. **(7)** **(ii)** Give the program to delete all the leaf nodes of a binary tree. **(6)**	BTL2	Understand
2.	**(i)** Describe about tournament trees with an example. **(5)** **(ii)** Examine how binary tree is created from a general tree. **(8)**	BTL1	Remember
3.	Demonstrate the various tree traversal of a binary tree with algorithm and examples.	BTL 3	Apply
4.	Estimate the AVL tree for the following after rotation. **(4+6+3)** 	BTL 2	Understand
5.	Deduce the B tree to insert the following key elements (consider order of the B tree is 3). 55,4,44,3,6,7,9,45,46,56,57	BTL 5	Evaluate
6.	**(i)** Explain how to implement binary search tree. **(9)** **(ii)** Infer about threaded binary tree? Explain its use. **(4)**	BTL 4	Analyze
7.	**(i)** Describe how deletion can take place in AVL trees with suitable algorithms. **(8)** **(ii)** Recall about Huffman's tree with an example. **(5)**	BTL1	Remember
8.	Analyze about the insertion and deletion operations of B-tree with simulate abstract data types.	BTL 4	Analyze
9.	Define the following graph terminology **(i)** Adjacent node **(ii)** Degree of a node **(iii)** Regular graph **(iv)** Connected graph **(v)** Complete graph	BTL1	Remember

	(vi) Weighted graph **(2+2+2+2+2+3)**		
10.	Explain the following representation of graph **(i)** Adjacency Matrix representation **(7)** **(ii)** Adjacency list representation **(6)**	BTL 4	Analyze
11.	**(i)** Describe about the Breadth –First Search algorithm for graph traversal with an example and give its features. **(7)** **(ii)** Describe about the Adjacency Multi list representation of graph with an example. **(6)**	BTL1	Remember
12.	**(i)** Demonstrate Pseudo code for prim's algorithm. Also give an example to construct a minimum spanning tree. **(9)** **(ii)** Illustrate the applications of tree. **(4)**	BTL 3	Apply
13.	Summarize the Depth –First Search algorithm for graph traversal with an example and give its applications.	BTL 2	Understand
14.	**(i)** Develop a program to find an Euler circuit in a graph. Trace the algorithm with example. **(5)** **(ii)** Write the Pseudo code for Dijkstra's shortest path algorithm. Give suitable example to trace the algorithm. **(8)**	BTL 6	Create
15.	Write the B tree index for the following data 12,24,13,5,7,65,54,32,7	BTL 3	Apply
16.	Write the program for Depth –First Search algorithm for graph traversal.	BTL 3	Apply
17.	Demonstrate about the tree traversal with necessary examples	BTL 2	Understand
	PART C		
S.No	**Questions**	**BT level**	**Competence**
1.	**(i)** Create a binary search tree using the following data elements: 45,39,56,12,34,78,32,10,89,54,67,81 **(8)** **(ii)** Develop the algorithm to search for a given value in a binary search tree. **(7)**	BTL 6	Create
2.	Develop a program to implement Binary Tree and perform in order, pre order and post order traversals.	BTL 6	Create
3.	Design B tree to insert the following key elements (consider order of the B tree is 3). 37,7,24,3,5,7,8,35,77,56,61	BTL 6	Create
4.	Apply Kruskal's algorithm on the graph if F={{A},{B},{C},{D},{E},{F}} MST={} Q={(A,D),(E,F),(C,E),(E,D),(C,D),(D,F)(A,C),(A,B),(B,C)}	BTL 6	Create
5.	Develop the program to demonstrate the tree concept	BTL 6	Create

UNIT-IV
SORTING

Selection sort-Insertion sort – Merge sort – Quick sort – Heap sort – Bubble sort- Shell sort – Radix sort.

PART A

S.No	Questions	BT level	Competence
1.	Classify the different sorting methods.	BTL1	Remember
2.	What is topological sorting?	BTL1	Remember
3.	Demonstrate how do you do a selection sort?	BTL 3	Apply
4.	Illustrate the time complexity of insertion sort with an example.	BTL 3	Apply

5.	Develop Selection sort algorithm.	BTL6	Create
6.	Select the best sorting method out of the following - insertion sort, quick sort and merge sort and give justification.	BTL 2	Understand
7.	What is merge sort with example?	BTL1	Remember
8.	Define Merge sort.	BTL1	Remember
9.	Develop an algorithm for a quick sort.	BTL6	Create
10.	Point out the advantages of using quick sort.	BTL 4	Analyze
11.	How many type of heap are there?	BTL1	Remember
12.	Analyze about heap sort with example	BTL 4	Analyze
13.	How do you solve a bubble sort problem?	BTL 2	Understand
14.	Summarize about the time complexities of bubble sort and quick sort.	BTL 5	Evaluate
15.	Describe the complexity of bubble sort.	BTL 2	Understand
16.	Illustrate the basic idea of shell sort?	BTL 3	Apply
17.	What is the other name of shell sort? Why called so?	BTL 2	Understand
18.	Assess the advantage of shell sort over insertion sort.	BTL 5	Evaluate
19.	What is meant by Radix sort give example?	BTL1	Remember
20.	Analyze the fastest sorting algorithm, justify.	BTL 4	Analyze
21.	Name the slowest and fastest sorting technique	BTL1	Remember
22.	What are the two stages in which heap sort is conducted?	BTL 2	Understand
23.	Explain why binary search can not be performed using linked list?	BTL 4	Analyze
24.	List the sorting technique which uses logarithmic time complexity	BTL 3	Apply

PART B

S.No	Questions	BT level	Competence
1.	Summarize about the selection sort, its algorithm and also explain with an example.	BTL 2	Understand
2.	**(i)** Develop the sorting steps for the array given below using Selection sort 24,12,5,11,76,2,45,13 **(5)** **(ii)** Develop the program to sort an array using selection sort algorithm. **(8)**	BTL 6	Create
3.	**(i)** Give the program to sort an array using insertion sort algorithm.**(9)** **(ii)** Discuss the advantages of Insertion sort. **(4)**	BTL 2	Understand
4.	With an algorithm and example describe about the insertion sort.	BTL1	Remember
5.	**(i)** Infer a function to perform merge sort .Give example. **(6)** **(ii)** Explain a routine for Insertion sort. Sort the following sequence using Insertion sort. **(7)** 3,10,4,2,8,6,5,1	BTL 4	Analyze
6.	Discriminate about the Merge sort, its algorithm and also explain with an example.	BTL 5	Evaluate
7.	**(i)** Apply the merge sort algorithm to sort the following numbers. 4,7,2,9,1,8,3,15,12,6,14. **(7)** **(ii)** Illustrate the quick sort algorithm. **(6)**	BTL 3	Apply
8.	Describe about the quick sort, its algorithm and also explain with an example.	BTL1	Remember
9.	**(i)** Sort the following numbers using the quick sort algorithm. 35,11,39,17,24,46 **(8)** **(ii)** Point out the complexity of quick sort algorithm and also give its advantages and disadvantages. **(5)**	BTL 4	Analyze
10.	Explain in detail about heap sort algorithm with an example.	BTL 4	Analyze

11.	Describe about the Bubble sort, its algorithm and also explain with an example.	BTL1	Remember
12.	**(i)** Apply Bubble sort algorithm to sort the following numbers. 4,7,2,9,1,8,3,15,12,6,14. **(6)** **(ii)** Apply Shell sort algorithm to sort the following numbers. 63,19,7,90,81,36,54,45,72,27,22,9,41,59,33 **(7)**	BTL 3	Apply
13.	State and explain the shell sort. State and explain the algorithm for shell sort. Sort the elements using shell sort.	BTL1	Remember
14.	Summarize about the Radix sort, its algorithm and also explain with an example.	BTL 2	Understand
15.	Give the program for performing implementation of insertion sort	BTL 2	Understand
16.	Construct the algorithm and program for selection sort.	BTL 3	Apply
17.	Write the program for sorting the elements by radix sort.	BTL 3	Apply

PART C

S.No	Questions	BT level	Competence
1.	**(i)** Develop a program to enter 'n' numbers in an array .Redisplay the array with elements being sorted in ascending order. **(10)** **(ii)** Develop the algorithm for bubble sort. **(5)**	BTL 6	Create
2.	**(i)** Develop the program to sort an array using insertion algorithm. **(9)** **(ii)** Sort the following numbers using the selection sort algorithm. 54,45,72,27,22,9,41, 63,19,7,90 **(6)**	BTL 6	Create
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 6	Create
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 6	Create
5.	Sort the following list of numbers using bubble sort technique 52,1,27,85,66,23,13,57	BTL 6	Create

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
1.	Discuss about linear search.	BTL 2	Understand
2.	Define binary search.	BTL1	Remember
3.	State the advantages of collision resolution strategies.	BTL 2	Understand
4.	Analyze the average case complexity of the linear search algorithm.	BTL 4	Analyze
5.	What are the advantage and disadvantage of separate chaining and linear probing?	BTL1	Remember
6.	Compare linear search and binary search.	BTL 4	Analyze
7.	Illustrate the complexity of binary search.	BTL 3	Apply
8.	Develop a simple algorithm for a linear search.	BTL6	Create
9.	Give the fastest searching algorithm.	BTL 2	Understand

10.	Define hash function.	BTL1	Remember
11.	Analyze how does the division remainder method help in hashing method?	BTL 4	Analyze
12.	Demonstrate the basic difference between static hashing and dynamic hashing?	BTL 3	Apply
13.	Define hash table?	BTL1	Remember
14.	Brief about Extendible hashing.	BTL 2	Understand
15.	Summarize briefly about Rehashing?	BTL 5	Evaluate
16.	List out the different types of hashing functions?	BTL1	Remember
17.	Illustrate about overflow handling?	BTL 3	Apply
18.	Summarize about Hash index.	BTL 5	Evaluate
19.	What is meant by B-Tree Indexing	BTL1	Remember
20.	Design the Pseudo code for fastest searching algorithm.	BTL6	Create
21.	What is overflow in hashing?	BTL1	Remember
22.	Interpret uniform hash function.	BTL 2	Understand
23.	Write the major problem in linear probing.	BTL 3	Apply
24.	Point out the advantages of B tree indices.	BTL 4	Analyze
	PART B		
S.No	**Questions**	**BT level**	**Competence**
1.	Summarize about the linear search with algorithm and an example.	BTL 2	Understand
2.	With an algorithm describe about the Binary search and also examine with an example.	BTL1	Remember
3.	Develop the program to demonstrate the binary search.	BTL6	Create
4.	**(i)** Differentiate linear search and binary search. State and explain the algorithms for both the search with example. **(7)** **(ii)** Explain Rehashing and extendible hashing. **(6)**	BTL 4	Analyze
5.	**(i)** Examine the collision resolution methods in hashing. **(7)** **(ii)** Describe about different types of hash functions. **(6)**	BTL1	Remember
6.	Deduce extendible hash structure to insert the following key elements 2,3,5,7,11,17,19,23,29,31.	BTL 5	Evaluate
7.	**(i)** When do you perform rehashing? Illustrate with example. **(8)** **(ii)** Summarize about the real world applications of hashing. **(5)**	BTL 4	Analyze
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 **(i)** Chaining **(4)** **(ii)** Linear probing **(4)** **(iii)** Double hashing .The second hash function h2(x)=7-(x mod 7). **(5)**	BTL 3	Apply
9.	Given input {4371,1323,6173,4199,4344,9679,1989} and a hash function h(x)=x (mod10).Estimate the resulting **(i)** Open hash table **(4)** **(ii)** Closed hash table using linear probing **(4)** **(iii)** Closed hash table using quadratic probing **(5)**	BTL 2	Understand
10.	Describe about the overflow handling in data structure with necessary example.	BTL1	Remember
11.	**(i)** What is meant by collision resolution by chaining? **(4)**	BTL1	Remember

	(ii) Describe about the operations on a chained hash table. **(9)**		
12.	Demonstrate about the different types of indexing used in data structure.	BTL 3	Apply
13.	Summarize about the following **(i)** B tree indices **(7)** **(ii)** Hashed Indices **(6)**	BTL 2	Understand
14.	Explain about the B tree indices algorithm with an example.	BTL 4	Analyze
15.	Demonstrate about the different types of hash function with necessary examples	BTL 3	Apply
16.	Summarize about the different collision resolution strategies with examples	BTL 2	Understand
17.	Describe about the Extendible Hashing with necessary examples.	BTL 3	Apply
	PART C		
S.No	**Questions**	**BT level**	**Competence**
1.	Develop a program to demonstrate linear search and binary search with algorithm and an example.	BTL 6	Create
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 6	Create
3.	Design the code to initialize, insert, delete and search a value in a chained hash table.	BTL 6	Create
4.	Develop the algorithm and also program to perform the B tree indices.	BTL 6	Create
5.	Develop the program for the implementation of hash table and collision handling by linear probing	BTL 6	Create