PART – A (SHORT ANSWER QUESTIONS)

S. No	Question	Blooms Taxonomy Level	Course Outcome
	UNIT – I		
1	**Define** the term algorithm and state the criteria the algorithm should satisfy?	Remember	5
2	**Define** recursive algorithm?	Remember	2
3	**Differentiate** between recursive and iterative algorithms?	Remember	2
4	**Define** asymptotic notations: big 'Oh', omega and theta?	Remember	5
5	**Describe** best case, average case and worst case efficiency of an algorithm?	Remember	5
6	**How** do you measure the algorithm running time?	Understand	5
7	**Describe** the role of space complexity and time complexity in measuring the performance of a program?	Understand	5
8	**Define** the term Data abstraction?	Remember	6
9	**Define** data structure?	Remember	3
10	**List** linear and nonlinear data structures?	Remember	3
11	**List** the operations performed in the Linear Data Structure?	Remember	3
12	**List** out any four applications of data structures?	Understand	4
13	**Define** Linked List?	Remember	6
14	**State** the different types of linked lists?	Remember	6
15	**List** the basic operations carried out in a linked list?	Remember	6
16	**List** the advantages and disadvantages of linked list?	Remember	6

S. No	Question	Blooms Taxonomy Level	Course Outcome
17	**Define** Sparse Matrix and its Representation with example?	Remember	6
18	**Define** Doubly Linked List?	Remember	6
19	**List** areas where data structures can be applied?	Remember	6
20	**Define** Circular Linked List?	Remember	6
	UNIT – II		
1	**Define** Stack?	Remember	1
2	**List** the applications of stack?	Remember	6
3	**Define** Queue?	Remember	6
4	**List** the applications of queue?	Remember	6
5	**Differentiate** Stack and Queue?	Understand	6
6	**List** out the basic operations that can be performed on a stack and queue?	Remember	6
7	**List** the different types of queues?	Remember	6
8	**Define** Circular Queue?	Remember	6
9	**List** the operations that can be performed on Circular Queue?	Remember	6
10	**Define** Circular Queue full condition?	Remember	6
11	**Define** DEQUEUE?	Remember	6
12	**List** the operations that can be performed on DEQUEUE?	Remember	6
13	**State** the different ways of representing expressions?	Remember	6
14	**State** the rules to be followed during infix to postfix conversions?	Remember	4
15	**Convert** the infix expression (a+b)-(c*d) into post fix form?	Apply	4
16	**List** how Stacks and Queues are represented in data structure ?	Understand	6
17	**Discuss** which data structure used in recursion?	understand	6
18	**Explain** the difference between stack implementation using array and linked list?	Understand	6
19	**Write** the necessity of infix to post fix conversion?	Understand	4
20	**Write** the Dequeue empty condition?	Remember	6
	UNIT – III		
1	**Define** Tree?	Remember	6
2	**List** the applications of Trees?	Understand	6
3	**Define** the terms node, degree, siblings, depth/height, level?	Remember	6
4	**Define** path in a tree	Remember	6
5	**Define** Binary Tree?	Remember	6
6	**Define** full binary tree?	Remember	6
7	**Define** complete binary tree?	Remember	6
8	**Define** a right-skewed binary tree and Left-skewed binary tree?	Remember	6
9	**State** the properties of a Binary Tree?	Remember	6
10	**Discuss** how to represent Binary Tree?	Remember	6
11	**List** the different tree traversals?	Remember	10
12	**Discuss** threaded binary tree?	Remember	6
13	**Define** heap?	Remember	6
14	**Define** Priority Queue?	Remember	6
15	**Differentiate** Max-heap and Min-heap?	Understand	6
16	**Define** graph?	Remember	6
17	**Discuss** representation of graph with examples?	Understand	10
18	**List** the different graph traversals?	Remember	10
19	**Differentiate** BFS and DFS?	Understand	10
20	**Differentiate** max priority queue and min priority queue?	Understand	6
	UNIT – IV		
1	**Differentiate** Linear search and binary search?	Understand	8
2	**Define** Hashing?	Remember	9
3	**Explain** Hash Function?	Remember	9
4	**List** different types of popular hash functions?	Remember	9

S. No	Question	Blooms Taxonomy Level	Course Outcome
5	**Define** Collision?	Remember	9
6	**State** different types of collision resolving techniques?	Remember	9
7	**Define** Separate Chaining?	Remember	9
8	**Define** Open Addressing?	Remember	9
9	**Define** Linear probing?	Remember	9
10	**Define** Quadratic Probing?	Remember	9
11	**Define** Double Hashing?	Remember	9
12	**Define** rehashing?	Remember	9
13	**List** the uses of hash table?	Understand	9
14	**Define** sorting and list the different types of sorting techniques?	Remember	8
15	**Discuss** the advantage of quick sort and its time complexity?	Understand	8
16	**State** the main idea behind Selection sort?	Remember	8
17	**Discuss** the time complexity of Heap sort?	Understand	8
18	**Discuss** the main idea behind Insertion sort?	Understand	8
19	**Discuss** is the space complexity of Radix sort?	Understand	8
20	**Compare** efficiencies of quick sort and heap sort	Understand	8
	UNIT – V		
1	**Define** balanced search tree?	Remember	6
2	**Define** binary search tree with example?	Remember	6
3	**State** the operations on binary search tree?	Remember	6
4	**Compare** binary tree and binary search tree?	Understand	6
5	**Define** balance factor and what is the height of an AVL tree?	Understand	6
6	**Define** AVL tree with example?	Remember	6
7	**List** the different AVL tree rotations to insert a node ?	Remember	6
8	**Discuss** the drawbacks of AVL trees?	Understand	6
9	**Define** splay tree?	Remember	6
10	**Define** B-tree with example?	Remember	6
11	**Discuss** the different operation's on B-Trees?	Remember	6
12	**Write** the properties of B-Trees?	Remember	6
13	**Explain** the procedure to insert a node into B-Tree?	Apply	6
14	**State** the properties of red black tree?	Remember	6
15	**Define** and discuss the properties of tries?	Remember	6
16	**List** some pattern matching algorithms?	Remember	6
17	**Discuss** the time and space needed by Knuth Morris Pratt algorithm?	Understand	6
18	**List** types of Tries?	Remember	6
19	**Define** Prefixes and Suffixes?	Remember	6
20	**Define** failure function in KMP algorithm?	Understand	6

PART – B (LONGANSWER QUESTIONS)

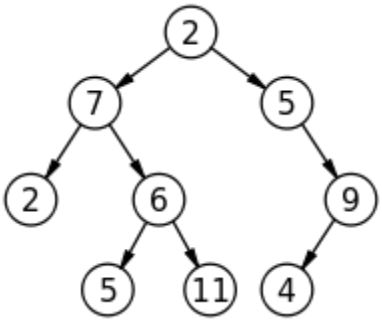
S. No	Question	Blooms Taxonomy Level	Course Outcome
	UNIT – I		
1	**Discuss** various the asymptotic notations used for best case average case and worst case analysis of algorithms.	Understand	5
2	**Explain** Performance Analysis in Detail.	Understand	5
3	**Define** recursion. Explain with it Fibonacci series and factorial of a number.	Apply	5
4	**Explain** time and space complexities in detail	Understand	5
5	**Explain** the different operations on singly liked list	Remember	6

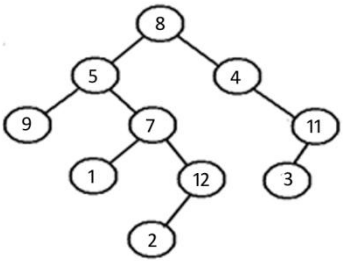
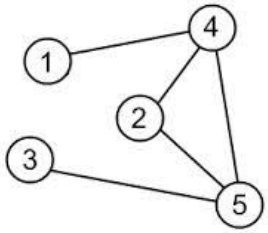
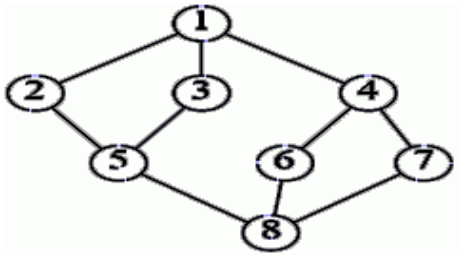
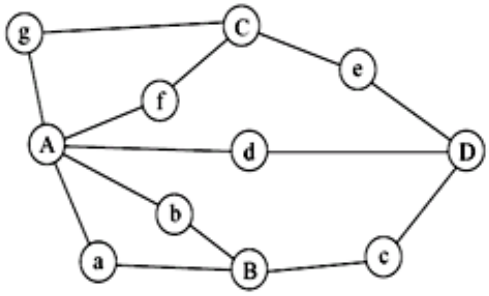
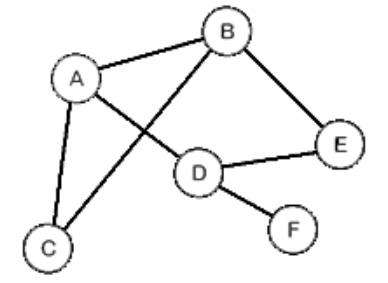
6	**Explain** concatenation of singly linked lists	Apply	6
7	**Explain** circular linked list operations	Remember	6
8	**Explain** doubly linked list operations	Remember	6
9	**List** the advantages and disadvantages of doubly linked list over singly linked list?	Understand	6
10	**Explain** the applications of doubly linked lists	Understand	6
11	**Explain** the following operations in a doubly linked list. (i) Insert an element (ii) Delete an element (iii) Reverse the list	Remember	6
12	**Write** an algorithm to insert and delete a key in a circular queue	Remember	6
13	**Explain** Array and Linked representation of Sparse Matrix	Understand	6
14	**Write** a program to insert an element in between two nodes in a double linked list	Apply	6
15	**Explain** how to create circular linked list and insert nodes at end	Apply	6
	UNIT - II		
1	**Write** an algorithm for basic operations on Stack	Remember	1
2	**Explain** the procedure to evaluate postfix expression	Remember	4
3	**Evaluate** the following postfix expression: 6 2 3 + - 3 8 2 / + * 2 | 3 +	Apply	4
4	**Explain** the procedure to convert infix expression into postfix expression	Remember	4
5	**Convert** the following expression A + (B * C) - ((D * E + F) / G) into post form.	Apply	4
6	**Explain** the operations on simple Queue	Remember	6
7	**Write** an algorithm for basic operations on circular queue	Remember	6
8	**Explain** DEQUEUE ADT and its operations	Remember	6
9	**Implement** a queue using two stacks.	Apply	6
10	**Implement** a Circular queue of integer of user specified size and write the functions for intilize () enque () and deque()	Understand	6
	UNIT - III		
1	**Explain** Binary tree ADT.	Remember	6
2	**Discuss** representation of binary tree	Remember	6
3	**Explain** tree traversals with example	Understand	10
4	**Discuss** max priority queue ADT with examples	Remember	6
5	**List** the advantages of priority queue? Explain the implementation of Priority Queue.?	Understand	6
6	**Define** threaded binary tree? Explain the impact of such a representation on the tree traversal procedure?	Understand	6
7	**Explain** graph ADT.	Remember	10
8	**Explain** different ways representation of graphs.	Remember	6
9	**Explain** BFS graphs traversal algorithms with suitable example.	Understand	10
10	**Explain** DFS graphs traversal algorithms with suitable example.	Understand	10
11	**Differentiate** BFS and DFS	Understand	6
12	**Explain** with an example how to insert an element to max heap	Apply	6
13	**Explain** with an example how to delete an element from max heap	Apply	6
14	**Define** Graph and explain how graphs can be represented in adjacency matrix and adjacency list	Understand	6
15	**Write** the advantages of using BFS over DFS or using DFS over BFS? What are the applications and downsides of each?	Understand	10
	UNIT – IV		
1	**Explain** linear search with example	Understand	8
2	**Explain** Binary search with example	Understand	8
3	**Differentiate** linear search algorithm with binary search algorithm.	Understand	8
4	**Define** hashing and discuss the different hashing functions with an example.	Understand	9
5	**Define** collision and discuss any two collision resolution techniques	Understand	9

6	**Explain** Chaining with an example	Understand	9
7	**Compare** different sorting techniques	Understand	8
8	**Write** C programs for implementing Quick sort to arrange a list of integers in ascending order	Apply	8
9	**Write** C programs for implementing Merge sort to arrange a list of integers in ascending order	Apply	8
10	**State** and explain insertion sort with an example	Apply	8
11	**State** and explain selection sort with an example	Apply	8
12	**State** and explain radix sort with an example	Apply	8
13	**State** and explain heap sort with an example	Apply	8
14	**State** andexplain quick sort with an example	Apply	8
15	**Explain** quick sort algorithm and simulate it for the following data 20, 35, 10, 16, 54, 21, 25	Apply	8
	UNIT – V		
1	**Describe** the insertion, deletion ,searching operations on binary search trees	Understand	6
2	**Explain** the insertion operation on AVL trees	Understand	6
3	**Describe** the insertion, searching operations on B-Trees	Understand	6
4	**Explain** knuth-Morris-pratt algorithm with example	Understand	6
5	**Define** binary search tree. Construct the binary search Tree for the below given data. P, F ,B, H, G , S, R, Y, T, W, Z	Apply	6
6	**State** the properties of Red-Black trees with example.	Understand	6
7	**Write** a short note on tries	Understand	6
8	**Compare** different search trees with their time complexities	Understand	6
9	**Explain** various rotations of AVL Trees maintaining balance factor while insertion takes place.	Understand	6
10	**Explain** Splay trees with example.	Understand	6

PART – C (PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)

S. No	Question	Blooms Taxonomy Level	Course Outcome
	UNIT – I		
1	$F(n)=3n^2-n+4$ **show** that $f(n)=O(n^2)$	Apply	5
2	$F(n)=5n^2+10n$ convert this to Ω() notation	Apply	5
3	$F(n)=\sqrt{n}$ and $g(n)=\log n$, **show** that $f(n)+g(n)=O(\sqrt{n})$	Apply	5
4	List out few of the applications that make use of Multilinked Structures?	Understand	2
5	Write a C program that uses functions to perform the following: a) Create a singly linked list of integers. b) Delete a given integer from the above linked list. c) Display the contents of the above list after deletion.	Apply	7
6	Write a C program that uses functions to perform the following: a) Create a doubly linked list of integers. b) Delete a given integer from the above doubly linked list. c) Display the contents of the above list after deletion.	Apply	7
7	Given a Singly linked list with each node containing either 0, 1 or 2. Write code to sort the list. Input: 1 -> 1 -> 2 -> 0 -> 2 -> 0 -> 1 -> 0 Output: 0 -> 0 -> 0 ->1 -> 1 -> 1 -> 2 -> 2	Apply	7
8	Given a linked list and two integers M and N. Traverse the linked list such that you retain M nodes then delete next N nodes, continue the same until end of the linked list. Input: M = 2, N = 2 Linked List: 1->2->3->4->5->6->7->8 Output: Linked List: 1->2->5->6 The main part of the problem is …	Apply	7

S. No	Question	Blooms Taxonomy Level	Course Outcome
9	Given two linked lists in a way such that the resultant must contain the elements alternatively from one list to other list. Input : LL1:1→2→3→4 LL2: 5→ 6→7 Output: 1→5→2→6→3→7→4	Apply	7
10	Write a program to remove duplicate vales from a double linked list		
	UNIT - II		
1	**Convert** the expression ((A + B) * C - (D - E) ^ (F + G)) into equivalent Postfix notation.	Apply	1
2	**Transform** the following expression to postfix expression using stacks. (a+b)*((d-e)+f)	Apply	1
3	**Convert**infix expression into its equivalent post fix expression A*(B+D)/E-F*(G+H/K)	Apply	1
4	**Transform** the following expression to postfix expression using stacks. (A+B)*(C$(D-E)+F)-G	Apply	1
5	**Write** a C program that uses stack operations to convert a given infix expression into its postfix Equivalent.	Apply	1
6	**Evaluate** the postfix expression 6 2 3 + - 3 8 2 / + * 2 $ 3 +	Apply	1
7	**Evaluate** the postfix expression 1 2 + 3 * 6 + 2 3 + /	Apply	1
8	**Evaluate** the postfix expression 10 2 8 * + 3 - 1 2 3 * + -	Apply	1
9	**Write** C programs to implement stack ADT using Arrays	Apply	7
10	**Write** C programs to implement stack ADT using Linked List	Apply	7
11	**Write** C programs to implement queue ADT using Arrays	Apply	7
12	**Write** C programs to implement queue ADT using Linked List	Apply	7
13	Write an algorithm for basic operations on simple queue	Apply	7
14	Write C programs to implement a double ended queue ADT using arrays	Apply	7
15	Write C programs to implement a double ended queue ADT using doubly linked list	Apply	7
	UNIT – III		
1	**Write** inorder, preoreder, post order traversal of the following tree 	Apply	10

S. No	Question	Blooms Taxonomy Level	Course Outcome
2	**Write** inorder, preoreder, post order traversal of the following tree 	Apply	10
3	**Illustrate** BFS and DFS traversals of following graph 	Apply	10
4	**Illustrate** DFS traversal of following graph 	Apply	10
5	**Illustrate** DFS and BFS traversals of following graph 	Apply	10
6	**Illustrate** BFS and DFS traversals of following graph 	Apply	10

S. No	Question	Blooms Taxonomy Level	Course Outcome
7	**Given** In order traversal of a binary tree is D,G,B,E,A,H,F,I,C and pre order traversal is A,B,D,G,E,C,F,H,I construct binary tree?	Apply	6
8	**Given** In order traversal of a binary tree is E,A,C,K,F,H,D,B,G and pre order traversal is F,A,E,K,C,D,H,G,B find the post order traversal?	Apply	6
9	**Given** a queue of elements with priorities: 21, 13,17,10,7,11 do the following: a)Build the binary heap (draw the tree at each step) and show the corresponding array b)Delete the element with the highest priority, drawing the tree at each step of the deleting procedure c)Insert a new element with priority 15 and draw the tree at each step of the insertion procedure	Apply	6
10	**Construct** max heap for 150, 80, 40,30,10, 70, 110, 100, 20, 90, 60, 50,120,140,130	Apply	6
	UNIT – IV		
1.	**Apply** binary search and find the average number of comparisons required to find an element 11,15,17,19,21,25,27,29,31	Apply	8
2.	**Using** linear search, delete the number 26 from the following list of numbers and give the steps 10 6 3 7 17 26 56 32 87	Apply	8
3.	**Apply** insertion sort on the following elements 3, 1, 4,7,5,9,2,6,5,10	Apply	8
4.	**Apply** the selection sort on the following elements21,11,5,78,49, 54,72,88	Apply	8
5.	**Rearrange** the following numbers using Quick sort procedure. 42, 12, 18, 98, 67, 83, 8, 10, 71	Apply	8
6.	Trace the quick sort algorithm for the following list of numbers. 90,77,60,99,55,88,66	Apply	8
7.	**Rearrange** the following numbers using radix sort. 77, 12, 8, 39, 27, 21, 44, 18, 6, 427, 117, 237, 5671 and 600	Apply	8
8.	**Apply** radix sort on the following list of elements 45,37,05,09,06,11,18,27	Apply	8
9.	**Apply** heap sort on list of elements 14,12,9,8,7,10,18,20,30	Apply	8
10.	**Explain** the heap sort algorithm by tracing the following elements stepwise 3, 5, 9, 7, 1, 4, 6, 8, 2	Apply	8
11.	**Use** quadratic probing to fill the Hash table of size 11. Data elements are 23,0,52,61,78,33,100,8,90,10,14,	Apply	9
12.	**Analyze** input (**371, 323, 173, 199, 344, 679, 989)** and hash function **h(x)=x mod 10**, Show the result Separate Chaining, linear probing	Apply	9
13.	**Analyze** input (**371, 323, 173, 199, 344, 679, 989)** and hash function **h(x)=x mod 10**, Show the result using quadratic probing, and double hashing **$h_2(x)$=7 - (x mod 7).**	Apply	9
14.	**Apply** quadratic hashing to fill the hash table of size 11 elements 20,5,10,22,33,40,50,30,51,31	Apply	9
15.	**Show** the each step of hash table entries for the given data set using linear probing 12,45,67,88,27,78,20,62,36,55 (size=10)	Apply	9
	UNIT – V		
1.	**Write** a C program that uses functions to perform the following: a) Create a binary search tree of characters. b) Traverse the above Binary search tree recursively in Postorder.	Apply	7
2.	**Give** an algorithm for constructing a binary search tree. While constructing the tree, take care that duplicate values are not added. Trace the algorithm on 2 ,5 , 9, 6, 12, 10, 13, 8	Apply	6
3.	**Construct** a binary search tree for the following 80, 40, 75, 30, 20, 90, 50	Apply	6
4.	**Construct** a binary search tree for the following 100, 50, 200, 25, 90, 80, 150	Apply	6
5.	**Insert** the following elements into an empty AVL Tree20,15,5,10,12,17,25,19	Apply	6

S. No	Question	Blooms Taxonomy Level	Course Outcome
6.	**Construct** an AVL Tree for following elements:10,20,15,3,2,16,18,26	Apply	6
7.	**Construct** AVL Tree for the following elements C,O,M,P,U,T,I,N,G	Apply	6
8.	**Construct** an AVL Tree for following elements:10,9,8,7,6,5,4,3,2,1	Apply	6
9.	**Construct** a B-tree of order 3 with the following elements 10,20,15,3,2,16,21,25,30,40	Apply	6
10.	**Insert** the following elements into an empty B-tree of order 5 3,14,7,1,8,5,11,17,13,6,23,12,20,4,16,18,24,25,19	Apply	6
11.	**Construct** a B-tree of order 3 with the following elements 25,10,20,30,80,40,50,60,82,70,90,85,93	Apply	6
12.	**Construct** a B-tree of order 7 with the following elements 4,40,23,50,11,34,62,78,66,22,90,59,25,72,64,77,39,12	Apply	6
13.	**Write** a C program that uses functions to perform the following: a) Create a binary search tree of integers. b) Traverse the above Binary search tree non recursively in inorder.	Apply	6
14.	**Write** a C program to perform the following operation: a)Insertion into a B-tree.	Apply	6
15.	**Find** the failure function for the pattern ”abacbba”	Apply	6
16.	**Define** failure function of KMP for the pattern “sisis”	Apply	6
17.	**Find** the failure function for the pattern ”abacab”	Apply	6
18.	**Apply** KMP algorithm on pattern “abacab” and text “abacaabaccabacabaabb”	Apply	6
19.	**Apply** KMP algorithm on pattern “abaa” and text “abbbaababaab”	Apply	6
20.	**Write** a C program for implementing Knuth-Morris- Pratt pattern matching algorithm to determine the index of the string S1 of length m in string S2 of length n where m<n	Apply	6