INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

COMPUTER SCIENCE AND ENGINEERING

TUTORIAL QUESTION BANK

Course Name	**DATA STRUCTURES**
Course Code	**ACS002**
Class	B. Tech II Sem
Branch	Common for CSE / ECE / EEE / IT
Year	2017-2018
Course Coordinator	Ms. B Padmaja, Associate Professor, CSE
Team of Instructors	Dr. J Sirisha Devi, Professor, CSE Ms. G Vasavi, Assistant Professor, CSE Ms. K Radhika, Assistant Professor, CSE Ms. B Rekha, Assistant Professor, IT Ms. A Soujanya, Assistant Professor, IT Mr. D Rahul, Assistant Professor, IT

COURSE OBJECTIVES (COs):

The course should enable the students to:

I	Learn the basic techniques of algorithm analysis.
II	Demonstrate searching and sorting algorithms and analyze their time complexities.
III	Implement linear data structures viz. stack, queue and linked list
IV	Demonstrate non-linear data structures viz. tree and graph traversal algorithms.
V	Study and choose appropriate data structure to solve problems in real world.

COURSE LEARNING OUTCOMES (CLOs):

Students, who complete the course, will have demonstrated the ability to do the following:

CACS002.01	Understand algorithms and data structures in terms of time and space complexity of basic operations.
CACS002.02	Analyze a given problem; choose an appropriate data structure and an algorithm to solve the problem.
CACS002.03	Choose a suitable algorithm to organize the data in ascending or descending order.
CACS002.04	Understand the difference between iterative and recursion approaches to solve problems.
CACS002.05	Explore an algorithm to find the location of an element in a given list.
CACS002.06	Understand the usage of divide and conquer strategy in searching and sorting applications.
CACS002.07	Compare the time complexities of various searching and sorting algorithms.
CACS002.08	Understand the working principle of linear data structures and their real time applications.
CACS002.09	Organize the data in various linked representation format.

CACS002.10	Design and implement abstract data types for linear and non-linear data structures.
CACS002.11	Describe the concept of non-linear data structures viz. trees and graphs and their applications.
CACS002.12	Compare and Contrast the operations of binary search trees and AVL trees.
CACS002.13	Understand the concept of M-way search trees, operations and applications.
CACS002.14	List out different tree and graph traversal techniques.
CACS002.15	Understand the implementation of hashing using hash table and hash function.
CACS002.16	Describe the concept of collision and its resolving methods in applications.
CACS002.17	Strengthen the knowledge of data structures and algorithms for employability.

TUTORIAL QUESTION BANK

UNIT – I INTRODUCTION TO DATA STRUCTURES, SEARCHING AND SORTING PART – A (SHORT ANSWER QUESTIONS)			
S. No	**Question**	**Blooms Taxonomy Level**	**Course Learning Outcome (CLOs)**
	UNIT – I INTRODUCTION		
1.	Draw the diagram showing classification of data structures?	Remember	CACS002.02
2.	List out various linear and non-linear data structures?	Remember	CACS002.02
3.	State the name of the method in which the solution of a problem depends on the solution of smaller instances of the same problem?	Understand	CACS002.04

4.	Find the output of the following code?
	def my_function(n):
	if(n==0):
	return;
	my_function(n-1)
	print(n)
	
	my_function(10)
		Understand	CACS002.04

5.	Write any two advantages of recursive approach than an iterative approach?	Remember	CACS002.04
6.	Write the two main measures for finding the efficiency of an algorithm?	Remember	CACS002.01
7.	Write the best case and worst case complexity of ordered linear search?	Remember	CACS002.01
8.	Find the output of the following code? for i in range(len(arr) -1) for j in range(i+1, len(arr)) if((arr[i].equals(arr[j])) && (i != j)) print(arr[i]	Remember	CACS002.01
9.	Write the disadvantage of linear search compared to other searching techniques?	Understand	CACS002.05
10.	Given a list arr = {2, 5, 7, 55, 72}, key = 72, Find the level of recursion using binary search?	Remember	CACS002.04
11.	Write the worst case time complexity of binary search using recursion?	Understand	CACS002.01
12.	Write any two applications of binary search?	Understand	CACS002.05
13.	Given a list arr = {45, 77, 89, 90, 94, 99, 100} and key = 99. Find the mid values in the first and second levels of recursion using binary search?	Remember	CACS002.05
14.	Write the names of algorithms which come under divide-and-conquer strategy?	Understand	CACS002.06
15.	Write the name of algorithmic technique which Fibonacci search uses?	Understand	CACS002.05

16.	Choose the recursive formula for the Fibonacci series.(n>=1) i. F(n) = F(n+1) + F(n+2) ii. F(n) = F(n) + F(n+1) iii. F(n) = F(n-1) + F(n-2) iv. F(n) = F(n-1) – F(n-2)	Understand	CACS002.04
17.	Consider a list arr = {1, 2, 4, 3}. Bubble sort is used to sort the elements of a list. Find out the number of iterations that will be required to sort the list?	Remember	CACS002.03
18.	Write the best, average and worst case time complexities of selection sort?	Understand	CACS002.03
19.	Write the worst case time complexity of quicksort when the input array is already sorted?	Remember	CACS002.03
20.	Write the best, average and worst case time complexities of merge sort?	Remember	CACS002.01
PART – B (LONG ANSWER QUESTIONS)			
1.	Define an algorithm? Write the structure and properties of an algorithm? How to measure the efficiency of an algorithm?	Remember	CACS002.01
2.	Use Big O notation to represent the complexity of an algorithm as a function of the size of the input n. Draw a table showing time complexities of different algorithms?	Remember	CACS002.01
3.	Define recursion? Write a recursive function that generates first N Fibonacci number. Discuss advantages and disadvantages of it.	Understand	CACS002.04
4.	Explain the asymptotic notations used for measuring the complexities of algorithms?	Understand	CACS002.04
5.	List out linear and non-linear data structures? Write a recursive algorithm to print GCD of two numbers?	Remember	CACS002.02
6.	Define sorting? Write the procedure for bubble sort using a suitable example?	Understand	CACS002.03
7.	Write a recursive algorithm to implement Towers of Hanoi problem?	Understand	CACS002.04
8.	Explain the following two comparison sort algorithms with an example and write their time complexities? i. Bubble sort ii. Selection sort	Understand	CACS002.03
9.	Explain Binary Search procedure for the following list of elements and assume the key element is 49. 12, 23, 34, 45, 55, 62, 71, 85, 96	Understand	CACS002.05
10.	Explain merge sort procedure for the given list of elements and also write its time complexity? 33, 14, 25, 45, 62, 85, 77, 65, 40, 22, 94	Understand	CACS002.03
11.	Write the name of the sorting technique which is used in playing cards game? Write a procedure for sorting a given list of numbers using that technique? 14, 25, 36, 74, 85, 6, 53, 62, 41	Understand	CACS002.03
12.	List out the algorithms which are based on divide-and-conquer algorithmic design strategy? Write the procedure for quick sort along with its time complexities?	Understand	CACS002.06
13.	Explain the procedure, advantages and disadvantages of linear and binary search with a suitable example?	Understand	CACS002.05
14.	Compare the time complexities of various searching and sorting algorithms?	Understand	CACS002.07
15.	Define searching? Explain Fibonacci search with a suitable example?	Understand	CACS002.05
PART – C (PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)			
1.	An algorithm has a run time complexity of O(log N). The algorithm requires 110 operations for an input size 1000. When the input size is doubled to 2000, the algorithm now requires 120 operations. Find the number of operations required when the size is doubled to 4000?	Understand	CACS002.01

2.	Explain the importance of data structures and discuss typical algorithm complexities of different problems? Write the best, average and worst case analysis of various searching algorithms?	Understand	CACS002.02

3.	Analyze the time complexity of the following code?
	long SumMN(int n, int m)
	{
	long sum = 0;
	for (int x = 1; x <= n; x++)
	{
	for (int y = 1; y <= m; y++)
	{
	if (x == y)
	{
	for (int i = 1; i <= n; i++)
	{
	sum += i * x * y;
	}
	}
	}
	}
	return sum;
	}
		Understand	CACS002.01

4.	Searching in a phone book: A phone book is stored in a text file, containing names of people, their city names and phone numbers. Choose an appropriate data structure to search a person's phone number based on his / her first name and city.	Understand	CACS002.05
5.	Sorting a phone book: Given a text file containing people's names, their city and phone numbers. Write a program which prints all the city names in an alphabetical order and for each one of them print their names in alphabetical order and their corresponding phone number?	Understand	CACS002.03
UNIT – II			
LINEAR DATA STRUCTURES			
PART – A (SHORT ANSWER QUESTIONS)			
1.	Write the minimum number of queues required to implement a priority queue?	Understand	CACS002.08
2.	Convert the following expression from infix to postfix notation? ((A + B) * C – (D – E) ^ (F + G))	Understand	CACS002.08
3.	Evaluate the following postfix operation using a stack? 8 2 3 ^ / 2 3 * + 5 1 * - Where ^ is the exponentiation operator.	Understand	CACS002.08
4.	Write the conditions for stack overflow and underflow situation?	Understand	CACS002.08
5.	Write the time complexity of POP() operation when the stack is implemented using a list?	Understand	CACS002.08
6.	Consider a stack of size N. Find the list position that will be occupied by a new element when PUSH() operation is performed?	Understand	CACS002.08
7.	In a circular queue, write the formula for inserting an element at the rear end of the queue?	Understand	CACS002.08
8.	Write the time complexity of enqueue and dequeue operation?	Remember	CACS002.10
9.	State the name of the data structure, in which deletion can be done from one end and insertion can take place only at the other end?	Remember	CACS002.10
10.	Identify the data structure, in which elements can be inserted or deleted at/from both the ends, but not in the middle?	Understand	CACS002.10
11.	List out any two applications of linear queue?	Understand	CACS002.10
12.	Write the conditions for linear queue full and empty?	Understand	CACS002.10
13.	State the disadvantages of linear queue?	Understand	CACS002.08

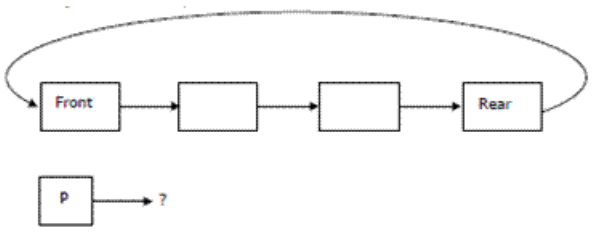
14.	Write any two applications of stack?	Understand	CACS002.08
15.	Find the result of evaluating the postfix expression 5, 4, 3, +, *, 4, 9, 3, /, +, *?	Understand	CACS002.08
16.	Convert the following infix expression to postfix expression using a stack using the usual precedence rule: x + y * z + (p * q + r) * s	Understand	CACS002.08
17.	Consider the following operation performed on a stack of size 5. Push(1); Pop(); Push(2); Push(3); Pop(); Push(4); Pop(); Pop(); Push(5); After the completion of all operation, find the number of elements present in stack?	Understand	CACS002.08
18.	If the elements "A", "B", "C" and "D" are placed in a stack and are deleted one at a time, write the order of removal?	Understand	CACS002.10
19.	State the data structure which is required to check whether an expression contains balanced parenthesis or not?	Understand	CACS002.10
20.	Write the prefix form of an infix expression p + q – r * t?	Understand	CACS002.08

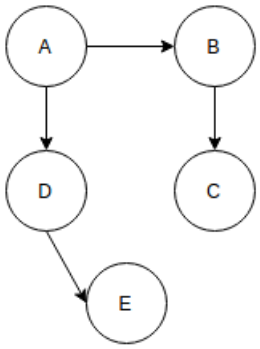
PART – B (LONG ANSWER QUESTIONS)			
1.	Explain the concept of stack. Write an algorithm to reverse a string using stack.	Remember	CACS002.08
2.	List out the application of stack and write down the algorithm to convert an infix expression to postfix form?	Understand	CACS002.10
3.	Implement the operations of a stack using single linked list?	Understand	CACS002.09
4.	Write the equivalent prefix and postfix expression for the given infix expression: (a * b) / 2 – (c / d – e)	Understand	CACS002.08
5.	Write the functional difference between stack and queue? Also write the applications of stack?	Understand	CACS002.08
6.	Compare between linear queue and circular queue? Write down algorithms for insert and delete operations in a circular queue?	Understand	CACS002.08
7.	Define a double ended queue (DEQUE)? Explain input restricted and output restricted DEQUE. Write an algorithm of input restricted DEQUE?	Understand	CACS002.10
8.	Explain the concept and basic operations of a linear queue? Write algorithms for ENQUEUE and DEQUEUE operations using a list?	Understand	CACS002.10
9.	Implement the operations of a linear queue using single linked list?	Understand	CACS002.0
10.	Explain the operations of a circular queue? Write a program for implementation of circular queue?	Understand	CACS002.10
11.	Convert following infix expressions into postfix form: i. A + (B*C-D/E*G) + H ii. (A+B) * (C-D/E)* G+H	Understand	CACS002.08
12.	Evaluate the following postfix notation of expression (Show status of stack after execution of each operations): 5 20 15 - * 25 2 * +	Understand	CACS002.08
PART – C (PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)			
1.	The following postfix expression with single digit operands is evaluated using stack. 8 2 3 ^ / 2 3 * + 5 / * - Note that ^ is exponential operator. Find the top two elements of the	Understand	CACS002.08

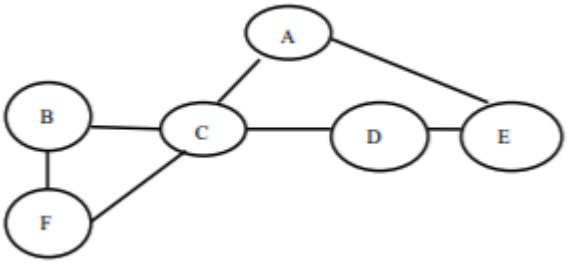
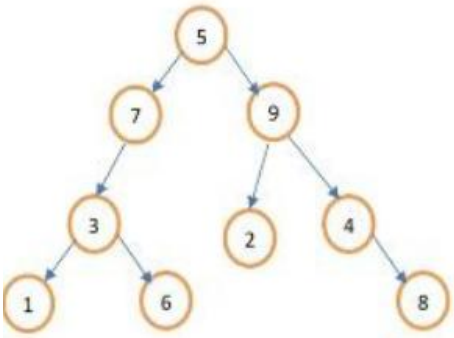
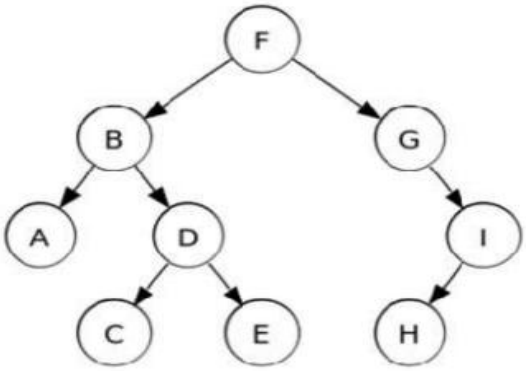
	stack after the first * is evaluated?		
2.	A single array A[1..MAXSIZE] is used to implement two stacks. The two stacks grow from opposite ends of the array. Variable top1 and top2 (top1 < top2) point to the top most elements in each of these stack. If the space is to be used efficiently, then **write** the condition for stack full?	Understand	CACS002.10
3.	To implement a queue using PUSH, POP and REVERSE operation, show how to implement ENQUEUE and DEQUEUE operations using a sequence of given operations?	Understand	CACS002.10
4.	The following postfix expression containing single digit operands and arithmetic operators + and * is evaluated using a stack 5 2 * 3 4 + 5 2 * * + Show the content of the stack after evaluating the above expression?	Understand	CACS002.08

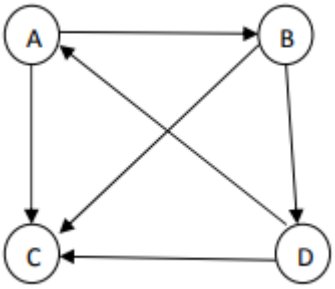
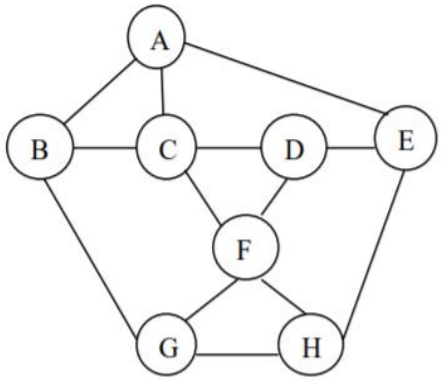
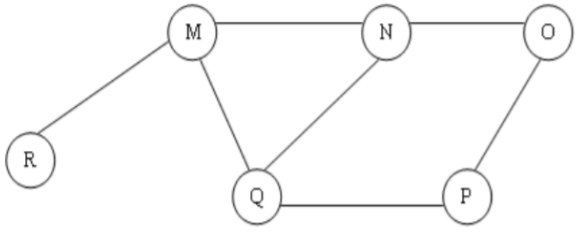
5.	Suppose that stack data structure is implemented with POP and PUSH operations along with clearstack() and emptystack() functions. Consider the code where test function takes any integer and returns true or false.
	clearstack();
	for (i=1;i<3;i++)
	{
	if (test(i))
	putc(i);
	else
	push(stack, i);
	}
	while(!emptystack(stack))
	{
	pop(stack, i)
	putc(i);
	}
	Find the output of the following code?
		Understand	CACS002.10

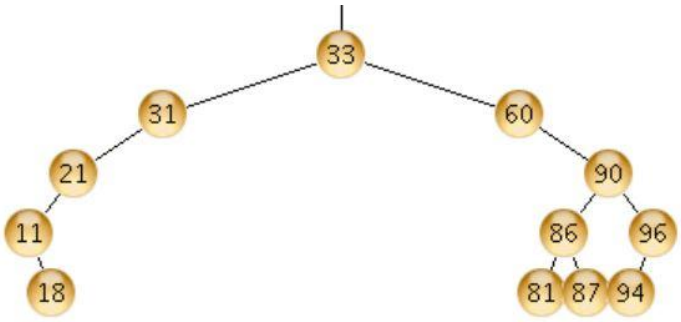
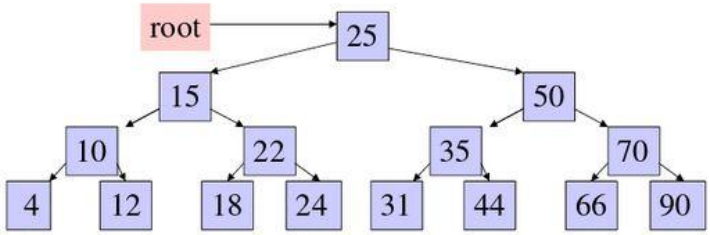
UNIT – III			
LINKED LISTS			
PART – A (SHORT ANSWER QUESTIONS)			
1.	Write the advantages of linked lists?	Remember	CACS002.09
2.	List out types of linked lists?	Remember	CACS002.09
3.	Write the advantages of double linked list over single linked list?	Remember	CACS002.09
4.	Write the applications of linked lists?	Remember	CACS002.09
5.	Find the time complexity to count the number of elements in a linked list?	Remember	CACS002.09
6.	Define a circular single linked list?	Remember	CACS002.09
7.	Write any two operations that is performed more efficiently by doubly linked list than singly linked list?	Remember	CACS002.09
8.	Consider a single linked list, list out any two operations that can be implemented in O(1) time?	Understand	CACS002.09
9.	Define a node in a doubly linked list?	Remember	CACS002.09
10.	Write the asymptotic time complexity to add an element in the linked list?	Understand	CACS002.09
11.	Identify the operation which is difficult to perform in a circular single linked list?	Understand	CACS002.09
12.	Write the asymptotic time complexity to insert an element at the second position in the linked list?	Understand	CACS002.09
13.	Identify the variant of linked list in which none of the node contains a NULL pointer?	Understand	CACS002.09
14.	In a circular linked list, how many pointers requires modification if a node is inserted?	Understand	CACS002.09
15.	Identify the searching technique for which linked lists are not suitable data structures?	Understand	CACS002.09

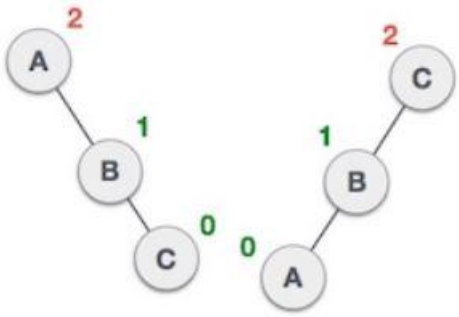
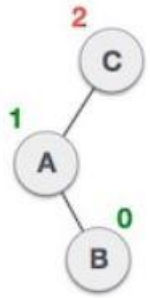
16.	In worst case, find the number of comparisons needed to search a singly linked list of length n for a given element?	Remember	CACS002.09
17.	State the name of data structure in which data elements is logically adjacent to each other?	Remember	CACS002.12
18.	Write the disadvantages of double linked list over single linked list?	Remember	CACS002.12
19.	Write the time complexity of enqueue() and dequeue() operations of a linked list implementation of a linear queue?	Remember	CACS002.12
20.	Write an example of a non-contiguous data structure?	Remember	CACS002.12
PART – B (LONG ANSWER QUESTIONS)			
1.	Write a program to implement the following operations of a single linked list: i. Creating a list ii. List traversal	Understand	CACS002.09
2.	A node can be inserted at various places in a linked list. Write algorithms for inserting a new node in a single linked list at: i. At the front of the linked list ii. After a given node iii. At the end of the linked list	Understand	CACS002.09
3.	Write a program to count the number of nodes present in a single linked list?	Understand	CACS002.09
4.	Write a program to search for an element present in a single linked list?	Understand	CACS002.09
5.	Write a program to delete a node from the middle position of the single linked list?	Understand	CACS002.09
6.	Write a program to reverse a single linked list of length n?	Understand	CACS002.09
7.	Write a program to implement the following operations of a double linked list: i. Creating a list ii. Inserting a node at the beginning	Understand	CACS002.09
8.	Write a program to implement the following operations of a circular single linked list: i. Creating a list ii. Deleting a node at the end	Understand	CACS002.09
9.	Write a program to merge two sorted linked list into a third linked list using recursion?	Understand	CACS002.09
10.	Write a function to delete a given node in a double linked list?	Understand	CACS002.09
PART – C (PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)			
1.	Write a program to split a circular linked list into two halves?	Understand	CACS002.09
2.	Define a node in a linked list? Explain the difference between creation of single linked list node and double linked list node?	Understand	CACS002.09
3.	Write a program to display node values in reverse order for a double linked list?	Understand	CACS002.09
4.	Write a program to swap nodes in a linked list without swapping data?	Understand	CACS002.09
5.	A circularly linked list is used to represent a Queue. A single variable p is used to access the Queue. Find the node to which p should point such that both the operations enQueue and deQueue can be performed in constant time? 	Understand	CACS002.09
UNIT – IV			

NON LINEAR DATA STRUCTURES			
PART – A (SHORT ANSWER QUESTIONS)			
1.	Write the children for node 'w' of a complete-binary tree in an array representation?	Remember	CACS002.11
2.	Write the advantages of linked list representation of binary trees over arrays?	Remember	CACS002.09
3.	Write the different tree traversal algorithms in linked list representation?	Remember	CACS002.09
4.	State the graph traversal technique which is similar to level order tree traversal?	Remember	CACS002.14
5.	Write the recursive algorithm for pre-order traversal?	Remember	CACS002.14
6.	Write the name of the tree traversal technique which would print the numbers in an ascending order in a binary search tree?	Remember	CACS002.14
7.	Define a full binary tree and complete binary tree?	Remember	CACS002.11
8.	Write the time complexity for finding the height of the binary tree?	Remember	CACS002.11
9.	Write the worst case and average case complexities of a binary search tree?	Understand	CACS002.12
10.	Write the number of edges present in a complete graph having n vertices?	Understand	CACS002.11
11.	Write the different ways used to represent a graph in computer?	Understand	CACS002.11
12.	Write the DFS traversal of the given graph? 	Understand	CACS002.14
13.	Write the maximum number of edges present in a simple directed graph with 7 vertices if there exists no cycles in the graph?	Remember	CACS002.14
14.	State the difference between pre-order traversal and post-order traversal?	Remember	CACS002.14
15.	Write the applications of trees?	Remember	CACS002.11
16.	Define binary search tree and its operations?	Remember	CACS002.12
17.	Define strictly binary tree with an example?	Remember	CACS002.11
18.	Write any two applications of priority queue?	Remember	CACS002.11
19.	Write the advantages of priority queue?	Understand	CACS002.13
20.	Write the time complexity to insert a node based on position in a priority queue?	Understand	CACS002.11
PART – B (LONG ANSWER QUESTIONS)			
1.	Construct a Binary Search Tree for the following data and do in-order, Preorder and Post-order traversal of the tree. 50, 60, 25, 40, 30, 70, 35, 10, 55, 65, 5	Understand	CACS002.12
2.	Explain the breadth first search and depth first search tree traversal on the following graph.	Understand	CACS002.14

			
3.	Illustrate the output obtained after pre-order, in-order and post-order traversal of the following tree 	Understand	CACS002.14
4.	Develop a program in Python to implement Depth First Search traversal of a graph using Adjacency Matrix.	Understand	CACS002.14
5.	Construct a binary search tree by inserting following nodes in sequence: 68, 85, 23, 38, 44, 80, 30, 108, 26, 5, 92, 60. Write in-order, pre-order and post-order traversal of the above generated Binary search tree.	Understand	CACS002.14
6.	Write the in-order, pre-order and post-order traversals for the given binary tree. 	Understand	CACS002.14
7.	Define Adjacency Matrix? Draw the Adjacency Matrix of the following graph. Also give adjacency list representation for the same.	Remember	CACS002.11

			
8.	Explain the array and linked representation of a binary tree using a suitable example?	Understand	CACS002.11
9.	Define a binary tree? Construct a binary tree given the pre-order traversal and in-order traversals as follows: Pre-Order Traversal: G B Q A C K F P D E R H In-Order Traversal: Q B K C F A G P E D H R	Remember	CACS002.11
10.	Construct an expression tree for the following expression. A + (B + C* D + E) + F / G. Make a preorder traversal of the resultant tree.	Remember	CACS002.11
11.	Explain the binary tree traversal algorithms with a suitable example?	Remember	CACS002.14
12.	Write the basic tree terminologies and the properties of binary tree?	Understand	CACS002.11
13.	Explain the breadth first search and depth first search graph traversal algorithms for the following graph? 	Understand	CACS002.14
14.	Explain the following with example: i. Full binary tree ii. Strictly binary tree iii. Complete binary tree	Understand	CACS002.11
15.	Write the applications of trees and graphs?	Understand	CACS002.11
16.	The Breadth First Search algorithm has been implemented using the queue data structure. Discover breadth first search for the graph shown in Figure with starting node M 	Understand	CACS002.14
17.	Define a binary search tree and write the properties of a binary search	Remember	CACS002.11

	tree? Construct a binary search with the following keys: 8, 3, , 1, 6, 14, 4, 7, 13, 17, 5		
18.	Write the procedure for finding an element 85 in a given binary search tree? 	Understand	CACS002.11
19.	Write a program for breadth first traversal of a graph?	Understand	CACS002.14
20.	Write the in-order, pre-order and post-order traversal of a given tree? 	Understand	CACS002.14
PART – C (PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)			
1.	Let G be a graph with n vertices and m edges. Find the tightest upper bound on the running time on depth first search of graph G. Assume that graph is represented using adjacency matrix.	Understand	CACS002.11
2.	Let G be a undirected graph with n vertices and 25 edges such that each vertex has degree at least 3. Find the maximum possible value of n?	Understand	CACS002.11
3.	In a binary tree, for every node the difference between the number of nodes in the left and right sub trees is at most two. If the height of the tree is h > 0, then find the minimum number of nodes in the tree?	Understand	CACS002.11
4.	Write a program to find the number of occurrences of a number in a tree of numbers?	Understand	CACS002.11
5.	Write breadth first search (BFS) traversal algorithm, based on a queue, to traverse a directed graph of n vertices and m edges?	Understand	CACS002.14
UNIT – V			
BINARY TREES AND HASHING			
PART – A (SHORT ANSWER QUESTIONS)			
1.	Define binary search tree?	Understand	CACS002.11
2.	Write the worst case and average case complexities of a binary search tree?	Remember	CACS002.11
3.	Define an AVL tree and its operations?	Remember	CACS002.12
4.	State the maximum height of an AVL tree with p nodes?	Remember	CACS002.12
5.	State the data structure which checks the height of the left and the right sub-trees and assures that the difference is not more than 1?	Remember	CACS002.11
6.	Write the formula for balance factor in AVL trees?	Remember	CACS002.12
7.	List out the types of rotations performed in AVL trees?	Understand	CACS002.12
8.	Explain how to perform left and right rotations on the right and left unbalanced AVL trees given below	Understand	CACS002.12

			
9.	Explain how to perform left-right rotation on the given unbalanced AVL tree? 	Understand	CACS002.12
10.	Construct a binary search tree with the following keys 27, 14, 35, 10, 19, 31, 42 and write the procedure to search for a key 20?	Remember	CACS002.12
11.	The height of a BST is given as h. Consider the height of the tree as the no. of edges in the longest path from root to the leaf. Find the maximum no. of nodes possible in the tree?	Understand	CACS002.12
12.	In full binary search tree every internal node has exactly two children. If there are 100 leaf nodes in the tree, Find the no of internal nodes present in the tree?	Understand	CACS002.12
13.	If a node having two children is to be deleted from binary search tree, then it is replaced by its which successor?	Remember	CACS002.12
14.	State the run time for traversing all the nodes of a binary search tree with n nodes and printing them in an order?	Understand	CACS002.12
15.	If n elements are sorted in a binary search tree, find the time complexity to search a key in the tree?	Understand	CACS002.12
16.	Write the purpose of a hash table?	Understand	CACS002.15
17.	State the techniques required to avoid collision?	Understand	CACS002.16
18.	Define a hash function and list out popular hash functions?	Remember	CACS002.15
19.	In simple chaining technique used in hashing, state which data structure is appropriate?	Remember	CACS002.15
20.	Write the applications of hashing?	Remember	CACS002.15
PART – B (LONG ANSWER QUESTIONS)			
1.	Define the properties of binary search trees? Write a program to construct a binary search tree with the given keys 8, 3, 10, 1, 6, 14, 4, 7, 13?	Understand	CACS002.11
2.	List out the operations of a binary search tree and write the procedure to search for a key 45 in a given binary search tree containing elements 25, 15, 50, 10, 22, 35, 70, 4, 12, 18, 24, 31, 44, 66, 90?	Understand	CACS002.12
3.	Write the procedure for inserting an element 60 in a given binary search tree containing elements 25, 15, 50, 10, 22, 35, 70, 4, 12, 18, 24, 31, 44, 66, 90?	Understand	CACS002.12

4.	Explain the different possibilities that arise while deleting an element from a given binary search tree containing elements 50, 30, 70, 20, 40, 60, 80? i. Delete 20 ii. Delete 30 iii. Delete 50	Understand	CACS002.12
5.	Define an AVL tree and write the steps used to follow while inserting an element 3 into an given AVL tree containing elements 13, 10, 15, 5, 11, 16, 4, 8.	Understand	CACS002.12
6.	Draw a hash table with open addressing and a size of 9. Use the hash function (k mod 9). Insert the keys: 5, 29, 20, 0, 27 and 18 into the hash table (in that order).	Understand	CACS002.15
7.	Define a B Tree and its properties? Construct a B tree of minimum degree 3 from the following elements 1, 2, 3, 4, 5, 6, 30, 40, 50, 60, 70, 80, 82, 84, 86.	Understand	CACS002.13
8.	Write the procedure for insertion and deletion operation in a B tree with the following elements 10, 20, 30, 40, 50, 60, 70, 80, 90.	Understand	CACS002.13
9.	Explain the collision resolution techniques separate chaining and open addressing with suitable example?	Understand	CACS002.16
10.	Explain the following: i. Hashing ii. Hash table iii. Hash Function	Understand	CACS002.15
PART – C (PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)			
1.	The integers {1-1000} are stored in a binary search tree (BST). Suppose the search algorithm is implemented on the key 363, one of the following sequences is not a possible sequence of nodes that is examined. It is i. 2, 252, 401, 398, 330, 344, 397, 363 ii. 924, 220, 911, 244, 898, 258, 362, 363 iii. 925, 202, 911, 240, 912, 345, 245, 363 iv. 2, 399, 387, 219, 266, 382, 381, 278, 363	Understand	CACS002.12
2.	If h is any hashing function and used to hash n keys into a table of size m, where m >= n, find the expected number of collisions involving a particular key x?	Apply	CACS002.15
3.	Consider a hash table with 9 slots. The hash function is h(k) = k mod 9. The Collisions are resolved by chaining. The following 9 keys are inserted in the order: 5, 28, 19, 15, 20, 33, 12, 17, 10. Find the maximum, minimum and average chain length in the hash table?	Apply	CACS002.16
4.	A binary search tree contains the numbers 1, 2, 3, 4, 5, 6, 7, 8. When the tree is traversed in pre-order and the values in each node printed out, the sequence of values obtained is 5, 3, 1, 2, 4, 6, 7, 8. Find the post order traversal sequence of the tree?	Apply	CACS002.12
5.	A hash table contains 10 buckets and uses linear probing to resolve collisions. The key values are integers and hash function used is key % 10. If the values 43, 165, 62, 123, 142 are inserted in the table, then find the location of the key value 142 in the table?	Apply	CACS002.16
6.	Find the smallest number of keys that will force a B-tree of order 3 to have a height 2?	Apply	CACS002.13

Prepared by : Ms. B Padmaja, Associate Professor
Date : 15.01.2018

HOD, CSE