University of Mumbai

Examination 2020 under cluster 4 (Lead College: PCE)

Examinations Commencing from 10th April 2021 to 17th April 2021

Program: Computer Engineering

Curriculum Scheme: Rev2019

Examination: SE Semester: III(for Direct Second Year-DSE)

Course Code: CSC303 and Course Name: Data Structure

Time: 2 hour Max. Marks: 80

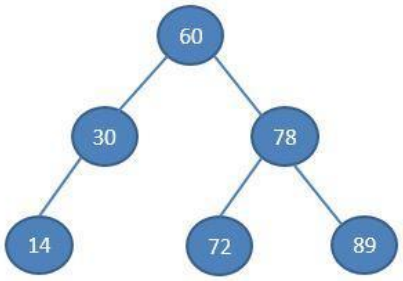
==========

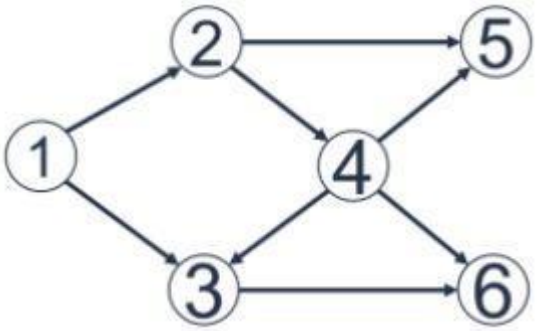
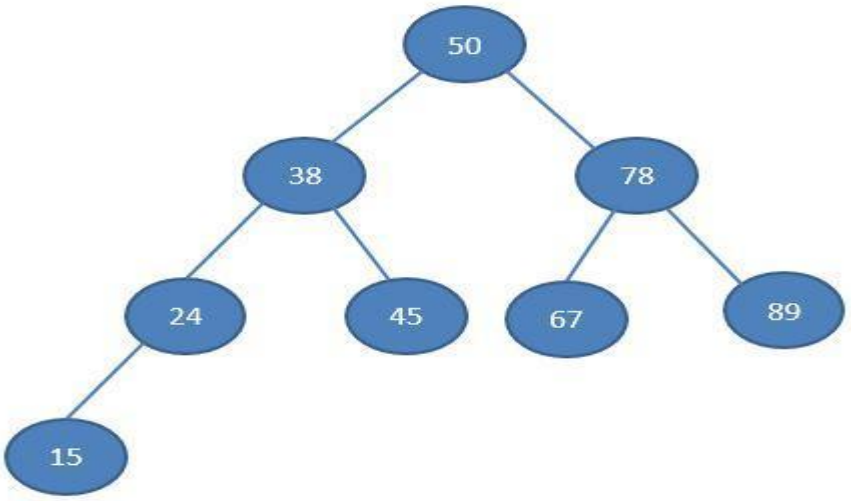
Q1.	**Choose the correct option for following questions. All the Questions are compulsory and carry equal marks**
1.	Which among the following is not a linear data structure?
Option A:	Stack
Option B:	Queue
Option C:	Tree
Option D:	Array
2.	Using division method, in a given hash table of size 114, the key 131 will be placed at position.
Option A:	31
Option B:	17
Option C:	14
Option D:	16
3.	For the implementation of parentheses balancing program using stack. What is the maximum number of parentheses that will remain on the stack [({ ( ) })] [[ ] { ([ ]) } ?
Option A:	0
Option B:	1
Option C:	2
Option D:	3
4.	Which of the following data structure is based on LIFO principle?
Option A:	Tree
Option B:	Graph
Option C:	Queue
Option D:	Stack
5.	If we insert the values 25, 14, 9, 18 and 37 in the Binary Search Tree then degree of root node will be
Option A:	0
Option B:	1
Option C:	2
Option D:	3
6.	Given the following input (22, 34, 71, 79, 89, 51, 73, 99) and the hash function x mod 10, which of the following statements are true?

	i) 79, 89, 99 hash to the same value ii) 71, 51 hash to the same value iii) All elements hash to the same value iv) Each element hashes to a different value
Option A:	i only
Option B:	ii only
Option C:	i and ii
Option D:	iii or iv
7.	What will be the front and rear of an initially empty queue after the following operations on it? enqueue(12), enqueue(10), enqueue(3), dequeue(),enqueue(18), dequeue(), enqueue(15), enqueue(15), dequeue()
Option A:	12, 15
Option B:	15, 18
Option C:	18, 15
Option D:	15, 15

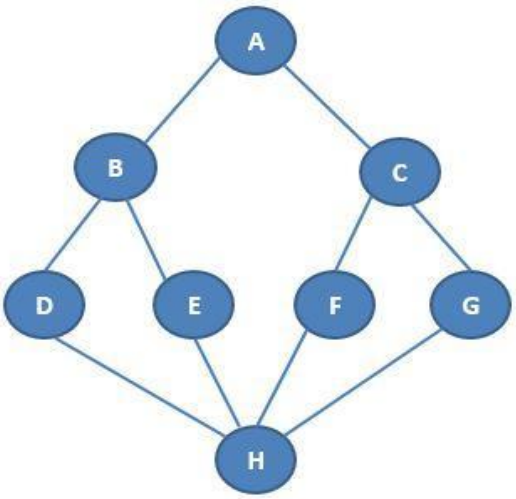
8.	In a Doubly linked list which statement is correct for dynamically allocating a memory for the node?
	`struct node`
	`{`
	`  struct node *prev;`
	`  char data;`
	`  struct node *next;`
	`};`
	`typdef struct node NODE;`
	`NODE *ptr;`

Option A:	ptr=(NODE*)malloc(sizeof(NODE));
Option B:	ptr=(NODE*)malloc(NODE);
Option C:	ptr=(NODE*)malloc(sizeof(NODE*));
Option D:	ptr=(NODE)malloc(sizeof(NODE));
9.	Which node pointers should be updated if a node B present between node A and node C of a doubly linked list is to be deleted.
Option A:	NEXT pointer of A, PREVIOUS pointer of B, NEXT pointer of C and PREVIOUS pointer of C
Option B:	NEXT pointer of A, PREVIOUS pointer of A, NEXT pointer of C and PREVIOUS pointer of C
Option C:	NEXT pointer of A, PREVIOUS pointer of C
Option D:	PREVIOUS pointer of A, NEXT pointer of C
10.	Consider the Binary Search Tree given below and find the result of in-order traversal sequence.

	
Option A:	60, 30, 14, 78, 72, 89
Option B:	14, 30, 72, 89, 78, 60
Option C:	60, 30, 78, 14, 72, 89
Option D:	14, 30, 60, 72, 78, 89
11.	You are given a stack with elements 2, 5, 8, 3, 9, 10 where 10 is the top of the stack. The elements are popped one-by-one and enqueued into a queue, until the stack becomes empty. The elements are again dequeued from the queue one-by-one and pushed into the stack. What is the final arrangement of elements in the stack (from top to bottom)?
Option A:	10, 9, 3, 8, 5, 2
Option B:	2, 5, 8, 3, 9, 10
Option C:	2, 3, 5, 8, 9, 10
Option D:	10, 9, 8, 5, 3, 2
12.	Which of the following is false about a doubly linked list?
Option A:	We can navigate in both the directions
Option B:	It requires more space than a singly linked list
Option C:	The insertion and deletion of a node take a bit longer
Option D:	Implementing a doubly linked list is easier than singly linked list
13.	The Data structure used in the standard implementation of Breadth First Search is?
Option A:	Tree
Option B:	Linked List
Option C:	Queue
Option D:	Stack
14.	In the linked list implementation of a queue, where does a new element get inserted?
Option A:	At the head of the linked list
Option B:	At the tail of the linked list
Option C:	At the centre position in the linked list
Option D:	After the specified position in a linked list
15.	Which type of linked list begins with a pointer to the first node and each node contains a pointer to the next node, and the pointer in the last node points back to the first node?
Option A:	Singly linked list

Option B:	Doubly linked list
Option C:	Circular singly linked list
Option D:	Circular doubly linked list
16.	What will be the topological ordering for the below graph. 
Option A:	1 2 3 4 5 6
Option B:	1 2 3 4 6 5
Option C:	1 3 2 4 5 6
Option D:	1 2 4 5 3 6
17.	Deletion and Insertion operation in Queue and Stack are known as?
Option A:	Enqueue and Dequeue, Push and Pop
Option B:	Push and Pop, Enqueue and Dequeue
Option C:	Pop and Push, Dequeue and Enqueue
Option D:	Dequeue and Enqueue, Pop and Push
18.	After adding a left child to the node 15 in an AVL Tree below, how many nodes will be unbalanced? 
Option A:	1
Option B:	2
Option C:	3
Option D:	4

19.	Degree of a leaf node is _____.
Option A:	0
Option B:	1
Option C:	2
Option D:	3
20.	When the left sub-tree of the tree is one level higher than that of the right sub-tree, then the balance factor is
Option A:	0
Option B:	1
Option C:	-1
Option D:	2

Q2	**Solve any Four out of Six** **5 marks each**
A	What is Data Structure? List different data structures along with applications.
B	Write an algorithm to check the well-formedness of parenthesis in an algebraic expression using Stack data structure.
C	Write functions in 'C' for the following operations of Input Restricted Deque. i) insert_right() ii) delete_left() iii) delete_right()
D	Make a comparison between linked list and linear array. Which one will you prefer to use and when?
E	Construct Huffman tree and determine the code for each symbol in the string "SUCCESSFUL".
F	Show Depth First Search traversal for the following graph with all the steps. 

Q3	**Solve any Two Questions out of Three** **10 marks each**
A	Write a program to perform the following operations on doubly linked list: i) Insert a node in the beginning ii) Delete a node from the end iii) Search for a given element in the list

	iv) Display the list
B	Insert the following elements in an AVL tree: 25, 44, 58, 15, 19, 11, 37, 32. Explain different rotations that can be used.
C	Using modulo division method, hash the following elements in a table of size 10. Use Linear probing and Quadratic probing to resolve the collisions. 28, 55, 71, 67, 11, 10, 90, 44

University of Mumbai

Examination 2020 under cluster 4 (Lead College: PCE)

Examinations Commencing from 10th April 2021 to 17th April 2021

Program: Computer Engineering

Curriculum Scheme: Rev2019

Examination: SE Semester: III(for Direct Second Year-DSE)

Course Code: CSC303 and Course Name: Data Structure

Time: 2 hour Max. Marks: 80

==========

Question Number	Correct Option (Enter either 'A' or 'B' or 'C' or 'D')
Q1.	C
Q2.	B
Q3.	B
Q4	D
Q5	C
Q6	C
Q7	C
Q8.	A
Q9.	C
Q10.	D
Q11.	B
Q12.	D
Q13.	C
Q14.	B
Q15.	C
Q16.	D
Q17.	D
Q18.	C
Q19.	A
Q20.	B

Q2 **Solve any Four out of Six** **5 marks each**

A

What is Data Structure? List different data structures along with applications.

Solution:

A data structure is a way of storing and organizing the data so that the data can be used efficiently. Different kinds of data structures are suited to different kinds of applications:

Data Structures	Applications
Arrays	• Arrangement of leader-board of a game can be done simply through arrays to store the score and arrange them in descending order to clearly make out the rank of each player in the game • 2D arrays, commonly known as, matrix, are used in image processing.
Stacks	• Converting infix to postfix expressions. • History of visited websites
Queues	• Operating System uses queue for job scheduling. • To handle congestion in networking queue can be used.
Linked List	• Web pages can be accessed using the previous and the next URL links which are linked using linked list. • The music players also use the same technique to switch between music.
Trees	• Databases uses tree data structures for indexing. • Huffman coding
Graphs	• Facebook's Graph API uses the structure of Graphs. • Networking components has huge application of graph

Definition – 1M

Any four data structures along with application – 4M

B

Write an algorithm to check the well-formedness of parenthesis in an algebraic expression using Stack data structure.

Solution:

Step 1: Scan the expression from left to right.
Step 2: Set flag = 1
Step 3: Repeat until each symbol in the expression is scanned
If symbol is '(' or '{' or '[', push it on the stack.
If symbol is ')' or '}' or ']', then
If stack is empty, then set flag = 0

	Else pop top of the stack and place it in temp. If symbol is ')' and temp is either '{' or '[', then set flag=0 and GOTO step 5 If symbol is '}' and temp is either '(' or '[', then set flag=0 and GOTO step 5 If symbol is ']' and temp is either '(' or '{', then set flag=0 and GOTO step 5 Step 4: If stack is not empty, then set flag=0 and GOTO step 5 Step 5: If flag =1, then Print " Valid expression" Else Print "Invalid expression" Step 6: END

C	Write functions in 'C' for the following operations of Input Restricted Deque.
	i) insert_right()
	ii) delete_left()
	iii) delete_right()
	
	Solution:
	#define MAX 10
	int deque[MAX];
	int left = –1, right = –1;
	
	void insert_right()
	{
	int val;
	printf("\n Enter the value to be added:");
	scanf("%d", &val);
	if((left == 0 && right == MAX–1) || (left == right+1))
	{
	printf("\n OVERFLOW");
	return;
	}
	if (left == –1)
	{
	left = 0;
	right = 0;
	}
	else
	{
	if(right == MAX–1)
	right = 0;
	else
	right = right+1;
	}
	deque[right] = val ;
	}
	
	void delete_left()
	{
	if (left == –1)

```
        {
            printf("\n UNDERFLOW");
            return ;
        }
        printf("\n The deleted element is : %d", deque[left]);
        if(left == right)
        {
             left = –1; right = –1;
        }
        else
        {
              if(left == MAX–1)
                    left = 0;
              else
                    left = left+1;
        }
}

void delete_right()
{
       if (left == –1)
       {
              printf("\n UNDERFLOW");
              return ;
       }
       printf("\n The element deleted is : %d", deque[right]);
       if(left == right)
       {
              left = –1; right = –1;
       }
       else
       {
              if(right == 0)
                     right=MAX–1;
              else
                     right=right–1;
        }
}
```

Insert_right() – 2M
Delete_left() – 1.5M
Delete_right() – 1.5M

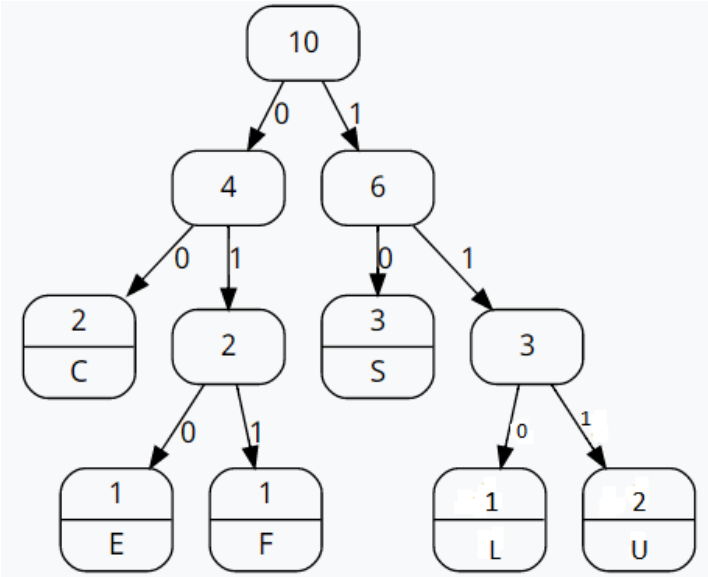
D	Make a comparison between linked list and linear array. Which one will you prefer to use and when? **Soluiton:** 1M for each point

	Arrays	Linked List
	An array is a collection of elements of a similar data type.	A linked list is a collection of nodes where nodes consists of two parts, i.e., data and address.
	Array elements store in a contiguous memory location.	Linked list elements can be stored anywhere in the memory or randomly stored.
	Memory is allocated at compile-time.	Memory is allocated at run time.
	Accessing any element in an array is faster as the element in an array can be directly accessed through the index.	Accessing an element in a linked list is slower as it starts traversing from the first element of the linked list.
	An array is suitable for applications in which the maximum size is known ahead of time.	If the maximum size is not known beforehand, we could use a linked list.

E

Construct Huffman tree and determine the code for each symbol in the string "SUCCESSFUL".

Solution:

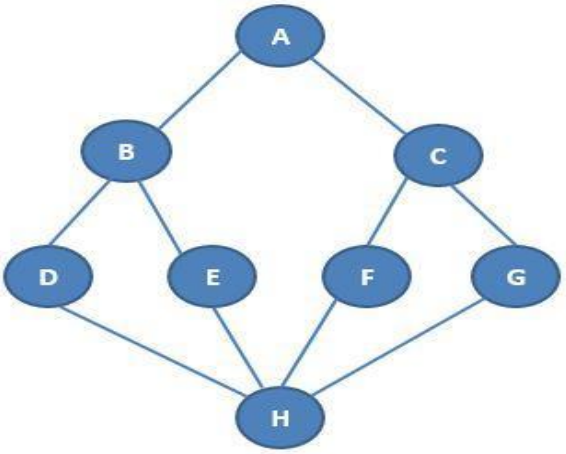


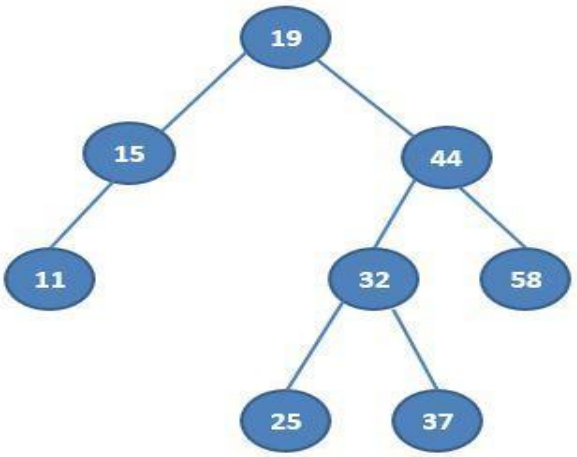
Character	Code
S	10
U	111
C	00
E	010
F	011
L	110

Huffman Tree – 4M
Code for each character – 1M

F

Show Depth First Search traversal for the following graph with all the steps.

	 DFS – A B D H E F C G Writing all steps with stack data structure – 4M Final DFS traversal order– 1M

Q3	**Solve any Two Questions out of Three** **10 marks each**
A	Write a program to perform the following operations on doubly linked list: i) Insert a node in the beginning ii) Delete a node from the end iii) Search for a given element in the list iv) Display the list Node definition – 1M Main function – 1M Insert Beginning – 2M Delete End – 2M Search – 2M Display – 2M
B	Insert the following elements in an AVL tree: 25, 44, 58, 15, 19, 11, 37, 32. Explain different rotations that can be used. 

	AVL tree – 8M Explanation of different rotations used – 2M
C	Using modulo division method, hash the following elements in a table of size 10. Use Linear probing and Quadratic probing to resolve the collisions. 28, 55, 71, 67, 11, 10, 90, 44 Writing all steps for calculating array index for given data – 3M Linear probing – 3M Quadratic probing – 4M