University of Mumbai

Examination 2020 under cluster 4 (Lead College: PCE, Panvel)

Examinations Commencing from 15th June 2021 to 26th June 2021

Program: COMPUTER ENGINEERING

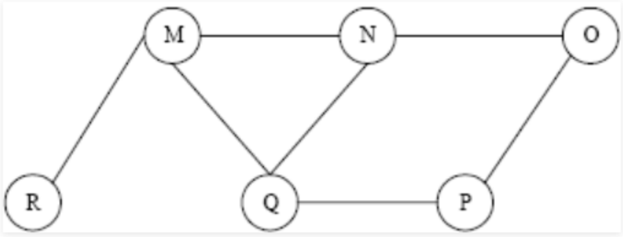
Curriculum Scheme: Rev2019

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

Course Code: CSC303 and Course Name: DATA STRUCTURE

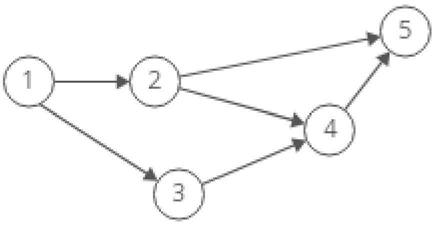
Time: 2 hour Max. Marks: 80

Q1.	**Choose the correct option for following questions. All the Questions are compulsory and carry equal marks**
1.	Which of the following option is true about nonlinear data structures?
Option A:	data elements are present at multiple levels.
Option B:	Garbage each element is traversable through a single run.
Option C:	data elements are sequentially connected
Option D:	Efficient utilization of memory.
2.	The operation of processing each element in the list is known as
Option A:	Creation
Option B:	Insertion
Option C:	Deletion
Option D:	Traversal
3.	A full binary tree with n leaves contains
Option A:	n - 1 nodes
Option B:	$\log_2 n$ nodes
Option C:	2n – 1 nodes
Option D:	2^n nodes
4.	Queue data structure is used for -
Option A:	Preorder traversal in tree
Option B:	Postorder traversal in tree
Option C:	Depth first traversal in graph
Option D:	Breadth first traversal in graph
5.	Top value in stack changes -
Option A:	While checking overflow
Option B:	While checking underflow
Option C:	Before deletion of an element from stack
Option D:	After deletion of an element from stack
6.	For which of the following operation, Linked lists are not suitable data structures?
Option A:	Linear search
Option B:	Binary search

Option C:	Sorting
Option D:	traversal
7.	Stacks cannot be used to
Option A:	evaluate an arithmetic expression in postfix form
Option B:	implement recursion
Option C:	convert a given arithmetic expression in infix form to is equivalent postfix form
Option D:	allocates resources (like CPU) by the operating system
8.	The Depth First Search algorithm has been implemented on following graph. One possible order of visiting the nodes of the graph is 
Option A:	MRQNOP
Option B:	NMRQPO
Option C:	OPMQNR
Option D:	NORMQP
9.	Which of the following is essential for evaluating a postfix expression?
Option A:	An operator stack
Option B:	An operand stack
Option C:	An operator stack and an operand stack
Option D:	A parse tree
10.	A tree in which, at every node the height of its left sub tree and right sub tree differ at most by one is known as
Option A:	AVL Tree
Option B:	Complete Binary Tree
Option C:	Binary Search Tree
Option D:	Threaded Binary Tree
11.	Hash function f defined as f(key)=key mod 11, with linear probing, is used to insert the keys 37,38,72,48,98,56 into a table index starting from 0. What will be the location of key 16?
Option A:	5
Option B:	6
Option C:	7
Option D:	8
12.	Assume a binary search tree created by inserting the values 27, 9, 23, 22, 29, 25, 15, 50, 95, 60, 40. Number of nodes in the right subtree will be
Option A:	4
Option B:	5

Option C:	6
Option D:	7
13.	Which is not the valid balance factor for an AVL tree
Option A:	0
Option B:	1
Option C:	-1
Option D:	2
14.	B+ tree can contain a maximum of 7 pointers in a node. What is the minimum number of keys in leaves?
Option A:	3
Option B:	4
Option C:	5
Option D:	6
15.	Which of the following statement is not true about the doubly linked list?
Option A:	We can traverse in both the directions.
Option B:	It requires extra space
Option C:	Implementation of doubly linked list is easier than the singly linked list
Option D:	It stores the addresses of the next and the previous node
16.	Given, arr = {1,3,5,6,7,9,14,15,17,19} and the search_key = 19, how many comparisons are required using binary search?
Option A:	1
Option B:	2
Option C:	3
Option D:	4
17.	B-tree of order n is a order-n multiway tree in which each non-root node contains
Option A:	at most (n - 1)/2 keys
Option B:	exact (n - 1)/2 keys
Option C:	at least 2n keys
Option D:	at least (n - 1)/2 keys
18.	Postfix expression corresponding to the infix expression “(1+ 4) / (8 - 6) * 3” is
Option A:	1 4 / 8 6 * 3 -
Option B:	14 / 8 6 *- 3 +
Option C:	1 4 + 8 6 / - * 3
Option D:	1 4 + 8 6 - / 3 *
19.	Which of the following trait of a hash function is most desirable?
Option A:	It should be easy to implement
Option B:	It should occupy less space
Option C:	It should cause less collisions
Option D:	It should cause more collisions
20.	Topological sort can be implemented on a?
Option A:	Linked list
Option B:	Binary tree

Option C:	Directed acyclic graph
Option D:	Directed cyclic graph

Q2 **(20 Marks Each)**	**Solve any Four out of Six** **5 marks each**
A	Write a C functions to implement insertion and deletion in queue using linked list.
B	Explain deletion of a node in a binary search tree.
C	Find topological sorting sequence in the following graph: 
D	Consider a hash table with size = 7. Using Linear probing, insert the keys 99,33,23, 44, 56,43,19 into the table.
E	Define ADT. Write ADT for stack.
F	Write an algorithm to check the well-formedness of parenthesis in an algebraic expression using Stack data structure.

Q3. **(20 Marks Each)**	**Solve any Two Questions out of Three** **10 marks each**
A	Create a Huffman tree and find Huffman codes for each character in the string “CONNECTION”.
B	Write a C program for Singly Linked list for performing following operations i. Create SLL ii. Display SLL iii. Delete last node from SLL iv. Insert a node at start of SLL
C	Draw the B-tree of order 4 created by inserting the following data arriving in sequence: 25,10,16,32,20,5,27,39,7,11.

University of Mumbai

Examination 2020 under cluster 4 (Lead College: PCE)

Examinations Commencing from 15th June 2021 to 26th June 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.	A
Q2.	D
Q3.	C
Q4	D
Q5	D
Q6	B
Q7	D
Q8.	B
Q9.	B
Q10.	A
Q11.	D
Q12.	B
Q13.	D
Q14.	A
Q15.	C
Q16.	D
Q17.	D
Q18.	D
Q19.	C
Q20.	C

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

A

Write a C functions to implement insertion and deletion in queue using linked list.

Solution:

Let the node declaration for queue using linked list implementation is:

```
struct node{
        int data;
        struct node *next;
        };
struct node *front=NULL, *rear=NULL,*temp,*newNode;
```

// Insertion function 'enqueue' for queue.

```
void equeue(int item)
{
  newNode=(struct node*) malloc(sizeof(struct node));
  newNode->data =item;
  newNode->next=NULL;
  if(front==NULL)
  {
    front=rear=newNode;
  }
  else
  {  rear->next=newNode;
    rear=newNode;
  }
}
```

// Deletion function 'dequeue' from queue.

```
void dequeue()
{
  if(front==NULL)
  {
    printf("queue is empty!!! Deletion not possible!!!\n");
    return;
   }
   else
   {
     temp=front;
     printf("\ndeleted item=%d",temp->data);
     front=front->next;
     free(temp);
    }
}
```

B

Explain deletion of a node in a binary search tree.

Solution:

Deletion Operation is performed to delete a particular element from the Binary Search Tree.

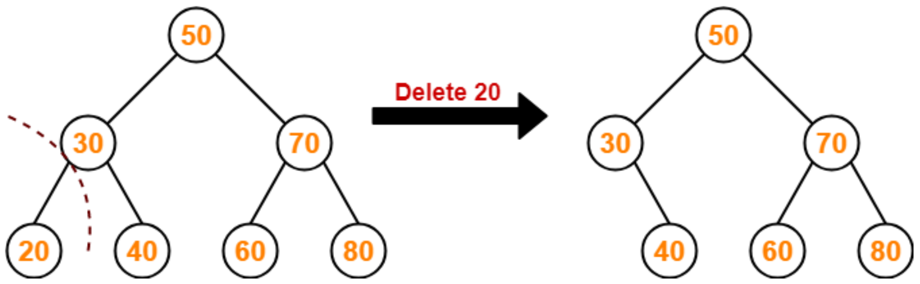
When it comes to deleting a node from the binary search tree, following three cases are possible-

Case-01: Deletion Of A Node Having No Child (Leaf Node)-

Just remove / disconnect the leaf node that is to deleted from the tree.

Example-

Consider the following example where node with value = 20 is deleted from the BST-

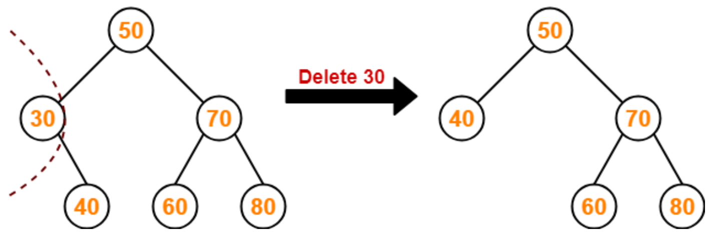


Case-02: Deletion Of A Node Having Only One Child-

Just make the child of the deleting node, the child of its grandparent.

Example-

Consider the following example where node with value = 30 is deleted from the BST-



Case-03: Deletion Of A Node Having Two Children-

A node with two children may be deleted from the BST in the following two ways-

Method-01:

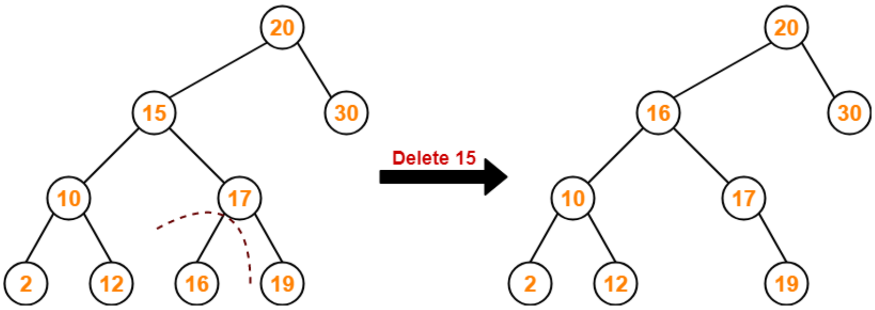
Visit to the right subtree of the deleting node.

- Pluck the least value element called as inorder successor.

- Replace the deleting element with its inorder successor.

Example-

Consider the following example where node with value = 15 is deleted from the BST-

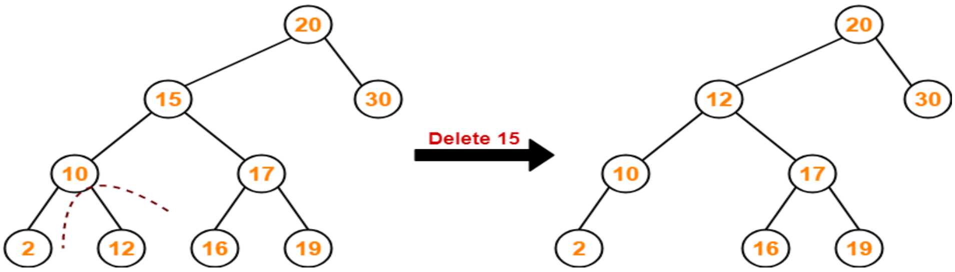


Method-02:

- Visit to the left subtree of the deleting node.
- Pluck the greatest value element called as inorder predecessor.
- Replace the deleting element with its inorder predecessor.

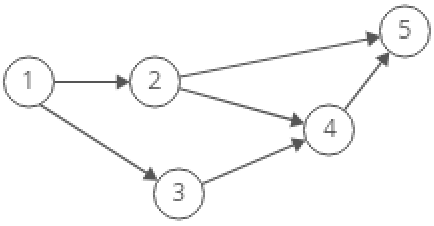
Example-

Consider the following example where node with value = 15 is deleted from the BST-



C

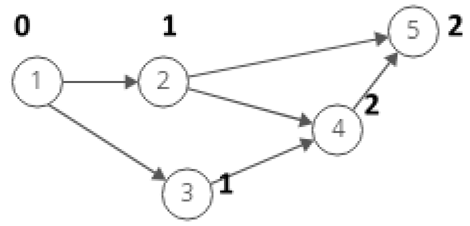
Find topological sorting sequence in the following graph:



Solution:

Step-01:

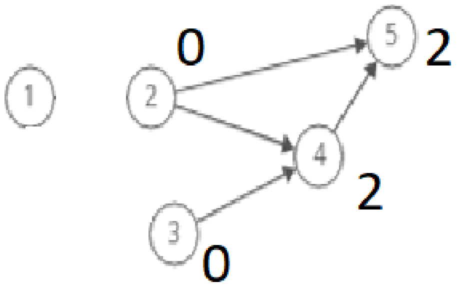
Write in-degree of each vertex-



Step-02:

- Vertex-1 has the least in-degree and add it in topological order list.
- So, remove vertex-1 and its associated edges.
- Now, update the in-degree of other vertices.

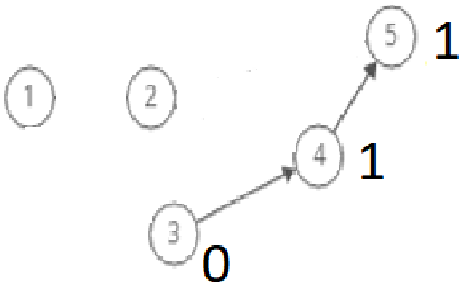
Topological order list : 1



Step-03:

- Vertex-2 & Vertex-3 has the least in-degree. So any one can be selected for removal and add that vertex in topological order list.
- So, remove vertex-2 as selected for ordering and its associated edges.
- Now, update the in-degree of other vertices.

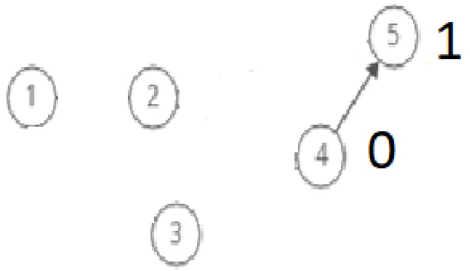
Topological order list : 1,2



Step-04:

- Vertex-3 has the least in-degree add that vertex in topological order list.
- So, remove vertex-3 and its associated edges.
- Now, update the in-degree of other vertices.

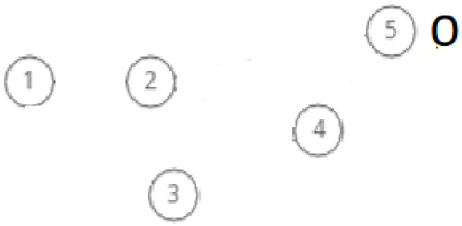
Topological order list : 1,2,3



Step-05:

- Vertex-4 has the least in-degree add that vertex in topological order list.
- So, remove vertex-4 and its associated edges.
- Now, update the in-degree of other vertices.

Topological order list : 1,2,3,4



Step-06:

- Vertex-5 has the least in-degree add that vertex in topological order list.
- So, remove vertex-5 and its associated edges.
- Now, update the in-degree of other vertices.

Topological order list : 1,2,3,4,5

Another possible topological ordering sequence is: 1,3,2,4,5.

D

Consider a hash table with size = 7. Using Linear probing, insert the keys 99,33,23, 44, 43 into the table.

Solution:

Formula with correct insertion for each key : 1Mark

Hash table of size=7

Index	0	1	2	3	4	5	6
Key		99	23	44	43	33	

E

Define ADT. Write ADT for stack.

Solution:

Definition: 2 Marks

ADT for Stack: 3 marks

F

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
```

Q3 Solve any Two Questions out of Three 10 marks each

A

Write a C program for Singly Linked list for performing following operations

i. Create SLL
ii. Display SLL
iii. Delete last node from SLL
Insert a node at start of SLL

Node definition – 1M
Main function – 1M
Create function – 2M
Display function – 2M
Insert at Beginning function – 2M
Delete last node function– 2M

B

Create a Huffman tree and find Huffman codes for each character in the string “CONNECTION”.

Char.	C	O	N	E	T	I
Frequency	2	2	3	1	1	1

Computing frequency:1 mark
Arranging and creating a nodes: 5 marks
Final tree: 1 mark
Assigning codes: 1mark
Computing code for each character: 2 marks

C

Draw the B-tree of order 4 created by inserting the following data arriving in sequence: 25,16,20,5,39,7,11.
Insertion of each key :7 Marks

	Correct Splitting: 2 marks Final tree : 1mark