-->Data Structure is field of study in which we discovers the techniques of handling the data, managing the data.

-->Data Structures (DS) tutorial provides basic and advanced concepts of Data Structure. Our Data Structure tutorial is designed for beginners and professionals.

-->Data Structure is a way to store and organize data so that it can be used efficiently.

-->Our Data Structure tutorial includes all topics of Data Structure such as Array, Pointer, Structure, Linked List, Stack, Queue, Graph, Searching, Sorting, Programs, etc.

04/02/2022

On this day had a brief introduction regarding data structures was given during the sessions as well as the concepts are to be taught in the course outline were briefly explained like Stacks, Queues, Linked List.

Data Structure is a representation of the logical relationship existing among individual elements of data.

Data Structure mainly specifies the following three things

-->Organization of Data

-->Accessing Methods

-->Degree of Associativity

Data Structure mainly deals with concept of Structures and Pointers thus a short practice on structures and pointers was also taken before the start of Data Structure concepts.

Data structures are classified into various types. The types are as follows :

1)Primitive data structures:-

->Primitive data structures are basic structures and are directly operated upon by machine instructions.

->Integers, floats, character and pointers are examples of primitive data structures.

2)Non primitive data structure:-

->These are derived from primitive data structures.

->The non-primitive data structures emphasize on structuring of a group of homogeneous or heterogeneous data items.

->Examples of Non-primitive data type are Array, List, and File.

3)Linear list:-

->A data structure is said to be Linear, if its elements are connected in linear fashion by means of logically or in sequence memory locations.

->Examples of Linear Data Structure are Stack and Queue, linked list

4)Nonlinear list:-

->Nonlinear data structures are those data structure in which data items are not arranged in a sequence.

->Examples of Non-linear Data Structure are Tree and Graph.

Along with the types few DS algorithms were also discussed like Merging, Sorting, Updation, Deletion, Traversal, etc.

Also we discussed Time and Space Analysis of the algorithms.

07/02/2022

--> On this dayy we have disvcovered the array in deep concept with respect to data structure.

Array was discussed in the session.

Representation of arrays:

1)One Dimensional Array:-

2)Two Dimensional Array:-

Application of Array:

1)Symbol Manipulation or Matrix Representation

2)Sparse Matrix

Further in the day few exercises were given on the topic learnt.

08/02/2022

In the session Stack was discussed.

Stack:

- A linear list which allows insertion and deletion of an element at one end only is called stack.

- The insertion operation is called as PUSH and deletion operation as POP.

- The most accessible elements in stack is known as top.

- The elements can only be removed in the opposite orders from that in which they were added to the stack.

- Such a linear list is referred to as a LIFO (Last In First Out) list.

Operations on Stack: Few operations were done on stack

->PUSH

->POP

->PEEP

->CHANGE

Applications of Stack: Various algorithms were discussed in which stack is widely used. These applications are

1)Evaluating Infix Expression

2)Conversion of Infix to Postfix and Evaluating Postfix Expression

3)Conversion of Infix to Prefix and Evaluating Prefix Expression

Further the given tasks were done.

09/02/2022

--> On this day we had learned about most popular technique like Queue.

In the session Queue was discussed.

Queue:

->A linear list which permits deletion to be performed at one end of the list and insertion at the other end is called queue.

->The information in such a list is processed FIFO (first in first out) or FCFS (first come first served) manner.

->Front is the end of queue from that deletion is to be performed.

->Rear is the end of queue at which new element is to be inserted.

->Insertion operation is called Enqueue & deletion operation is called Dequeue.

Operations on Queue:

1)Enqueue (Q, F, R, N, Y)

- This procedure inserts Y at rear end of Queue.

- Queue is represented by a vector Q containing N elements.

- F is pointer to the front element of a queue.

- R is pointer to the rear element of a queue.

2)Dequeue (Q, F, R)

- This function deletes & returns an element from front end of the Queue.

- Queue is represented by a vector Q containing N elements.

- F is pointer to the front element of a queue.

- R is pointer to the rear element of a queue.

Circular Queue:

-->A more suitable method of representing simple queue which prevents an excessive use of memory is to arrange the elements Q[1], Q[2]….,Q[n] in a circular fashion with Q[1] following Q[n], this is called circular queue.

-->In circular queue the last node is connected back to the first node to make a circle.

-->Circular queue is a linear data structure. It follows FIFO principle.

Operations on Circular Queue:

1.CQINSERT

2.CQDELETE

Dqueue:

-->A DQueue (double ended queue) is a linear list in which insertion and deletion are performed from the either end of the structure.

->There are two variations of Dqueue

Input restricted dqueue – allows insertion at only one end

Output restricted dqueue – allows deletion from only one end

Dqueue Algorithms:

1.DQINSERT\_REAR is same as QINSERT (Enqueue)

2.DQDELETE\_FRONT is same as QDELETE (Dequeue)

3.DQINSERT\_FRONT (Q, F, R, N, Y)

->This procedure inserts Y at front end of the Circular Queue.

->Queue is represented by a vector Q containing N elements.

->F is pointer to the front element of a queue.

->R is pointer to the rear element of a queue.

4.DQDELETE\_REAR(Q, F, R)

->This function deletes & returns an element from rear end of the Queue.

->Queue is represented by a vector Q containing N elements.

->F is pointer to the front element of a queue.

->R is pointer to the rear element of a queue.

Further in the session, the given tasks were completed.

10/02/2022

In this session Linked List was discussed

Linked List:

 Types:-

1. Singly Linked List

2. Circular Linked List

3. Doubly Linked List

 Operations:-

1. Insert

Insert at first position

Insert at last position

Insert into ordered list

2. Delete

3. Traverse list

4. Copy linked list

Singly Linked List:

->It is basic type of linked list.

->Each node contains data and pointer to next node.

->Last node’s pointer is null.

->First node address is available with pointer variable FIRST.

->Limitation of singly linked list is we can traverse only in one direction, forward direction.

Circular Linked List:

->If we replace NULL pointer of the last node of Singly Linked Linear List with the address of its first node, that list becomes circularly linked linear list or Circular List.

->FIRST is the address of first node of Circular List

->LAST is the address of the last node of Circular List

Doubly Linked List:

->In certain Applications, it is very desirable that a list be traversed in either forward or reverse direction.

->This property implies that each node must contain two link fields instead of usual one.

->The links are used to denote Predecessor and Successor of node.

->The link denoting its predecessor is called Left Link.

->The link denoting its successor is called Right Link.

->A list containing this type of node is called doubly linked list or two way chain.

11/02/2022

Concepts and Definitions of Graphs were discussed

1.Directed Graph

2.Indirected Graph

3.Mixed Graph

Concepts and Definitions of Trees were discussed

1.Directed Tree

- A directed tree is an acyclic digraph which has one node called its root with in degree 0, while all other nodes have in degree 1.

- Every directed tree must have at least one node.

- An isolated node is also a directed tree.

Further the given tasks were done.

14/02/2022

Tree Traversal concepts were discussed in the session

1.Preorder traversal of a binary tree is defined as follow

->Process the root node

->Traverse the left subtree in preorder

->Traverse the right subtree in preorder

2.Inorder traversal of a binary tree is defined as follow

->Traverse the left subtree in Inorder

->Process the root node

->Traverse the right subtree in Inorder

3.Postorder traversal of a binary tree is defined as follow

->Traverse the left subtree in Postorder

->Traverse the right subtree in Postorder

->Process the root node

Binary Search Tree (BST) :

A binary search tree is a binary tree in which each node possessed a key that satisfy the following conditions

 All key (if any) in the left sub tree of the root precedes the key in the root

 The key in the root precedes all key (if any) in the right sub tree

 The left and right sub trees of the root are again search trees

Further the given tasks were also completed.