Advanced Data structures

Bharatiya Vidya Bhavan's

Sardar Patel Institute of Technology

(Autonomous Institute Affiliated to University of Mumbai)

Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India

Department of CSE-DS

Course (Category) Code	Course Name	Teaching Scheme (Hrs/week)					Credits Assigned			
		L	T	P	O	E	L	T	P	Total
(PC)	Data Structures	3	0	2	5	10	3	0	1	4
		Examination Scheme								
		Component	ISE		MSE		ESE		Total	
DS202		Theory	75		75		150		300	
		Laboratory	50		--		50		100	

Pre-requisite Course Codes, if any.	1. Problem solving using imperative programming
Course Objective: To introduce the fundamentals and abstract concepts of Data Structures for Problem Solving.	
Course Outcomes (CO): At *the End of the course students will be able to*	
DS202.1	Apply various operations of linear and non-linear data structures to given problems.
DS202.2	Apply the concepts of Trees and Graphs to a given problem.
DS202.3	Apply various operations of heap data structures.
DS202.4	Apply the concepts of hashing on a given problem

Bharatiya Vidya Bhavan's

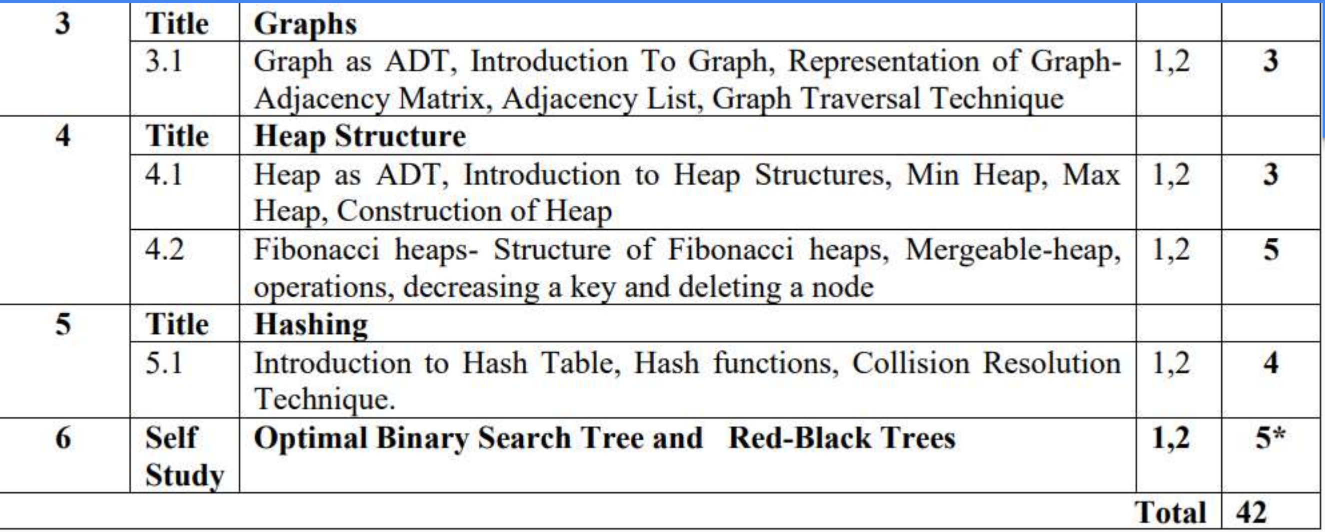
Sardar Patel Institute of Technology

(Autonomous Institute Affiliated to University of Mumbai)
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India

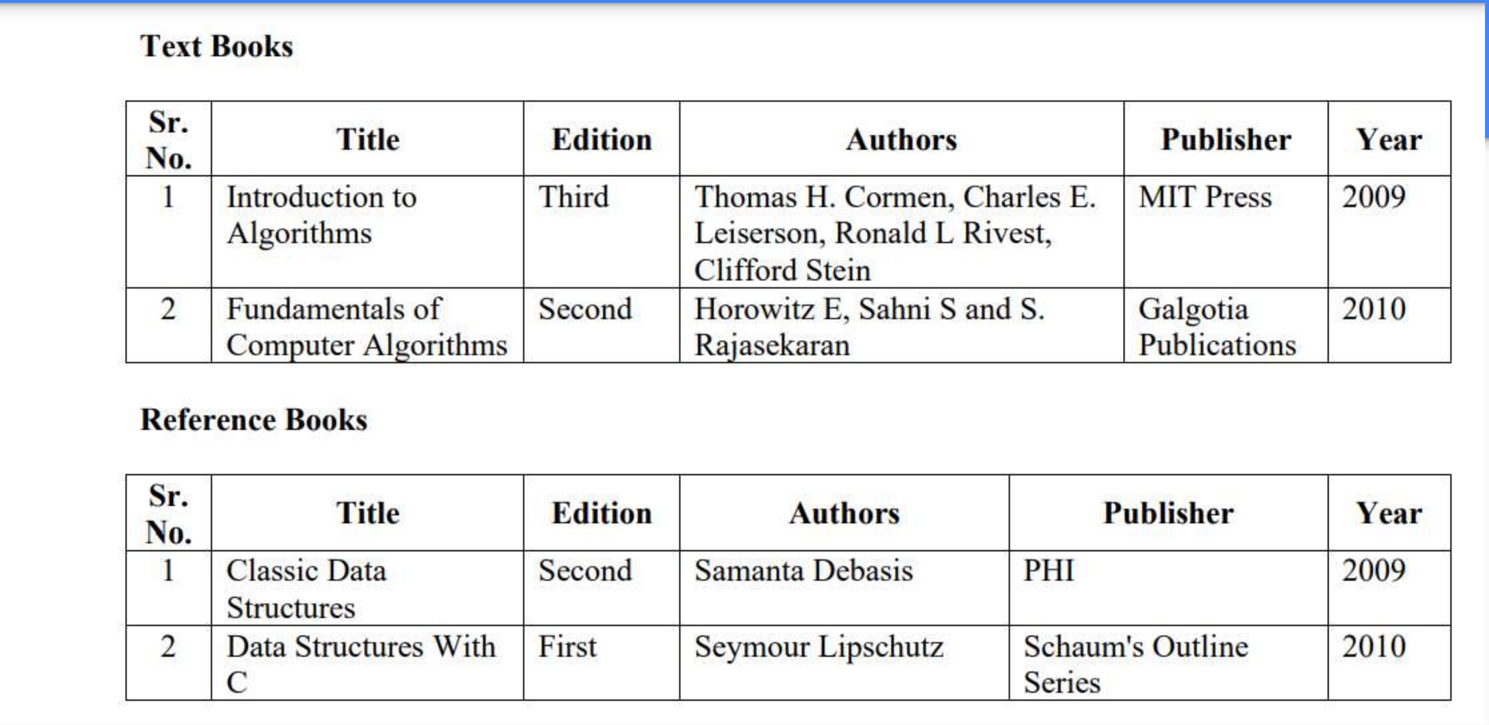
Department of CSE-DS

Theory Component

Module No.	Unit No.	Topics	Ref.	Hrs.
1	**Title**	**Introduction to Data Structures**		
	1.1	Concept of Linear and Nonlinear Data Structures	1,2	**1**
	1.2	Stack: Stack as ADT, operations on stack, Applications of Stacks	1,2	**4**
	1.3	Queue: Queue as ADT, Operations on Queue, Applications of Queue, Types of Queue-Circular and Priority Queue	1,2	**4**
	1.4	Linked List: Linked List as ADT, Operations on Singly Linked List.Types of linked list- Linear and circular linked lists, Doubly Linked List, Circular Linked List and its operations, Generalized Linked List (GLL) concept, Applications of linked List and Generalized Linked List (GLL).	1,2	**6**
2	**Title**	**Trees**		
	2.1	Trees as ADT, General tree v/s Binary Tree Terminology, Traversal of Binary Tree, Operations on Binary tree, Binary Search Tree and its operations, Expression Tree	1,2	**5**
	2.2	AVL Trees- Properties of AVL trees, Rotations, Insertion, and Deletion	1,2	**4**
	2.3	Introduction to B tree- Insertion, Deletion.	1,2	**3**



3	**Title**	**Graphs**		
	3.1	Graph as ADT, Introduction To Graph, Representation of Graph-Adjacency Matrix, Adjacency List, Graph Traversal Technique	1,2	**3**
4	**Title**	**Heap Structure**		
	4.1	Heap as ADT, Introduction to Heap Structures, Min Heap, Max Heap, Construction of Heap	1,2	**3**
	4.2	Fibonacci heaps- Structure of Fibonacci heaps, Mergeable-heap, operations, decreasing a key and deleting a node	1,2	**5**
5	**Title**	**Hashing**		
	5.1	Introduction to Hash Table, Hash functions, Collision Resolution Technique.	1,2	**4**
6	**Self Study**	**Optimal Binary Search Tree and Red-Black Trees**	**1,2**	**5***
			Total	**42**



Text Books

Sr. No.	Title	Edition	Authors	Publisher	Year
1	Introduction to Algorithms	Third	Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein	MIT Press	2009
2	Fundamentals of Computer Algorithms	Second	Horowitz E, Sahni S and S. Rajasekaran	Galgotia Publications	2010

Reference Books

Sr. No.	Title	Edition	Authors	Publisher	Year
1	Classic Data Structures	Second	Samanta Debasis	PHI	2009
2	Data Structures With C	First	Seymour Lipschutz	Schaum's Outline Series	2010

ISEs-

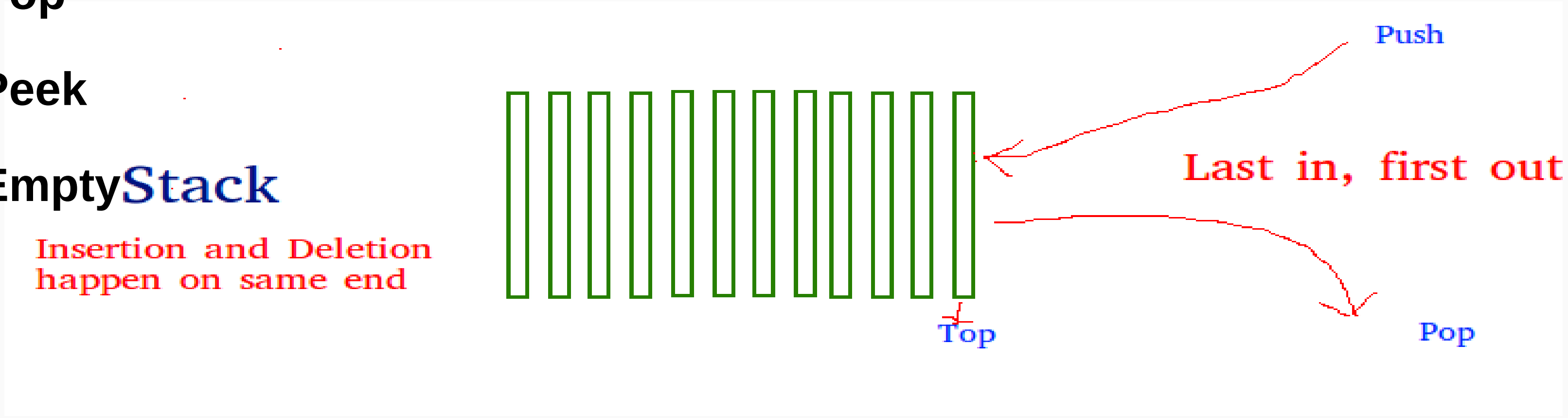
Stack

➢ Linear data structure

➢ LIFO(Last In First Out) or FILO(First In Last Out).

➢ Mainly the following three basic operations are performed in the stack:

- **Push**
- **Pop**
- **Peek**
- **Empty**

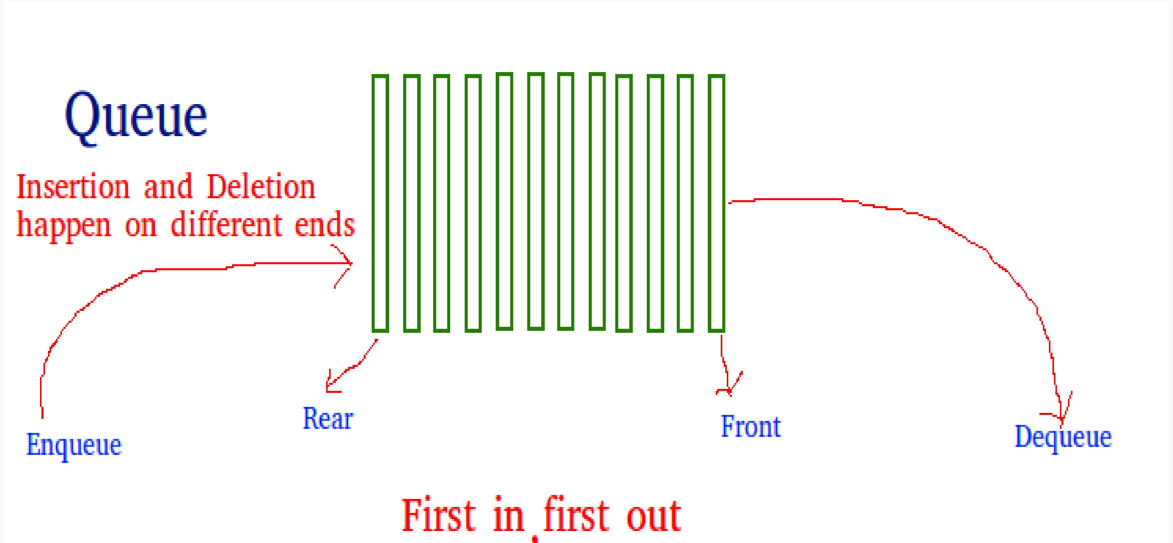


Applications of stack

- Balancing of symbols
- Infix to Postfix /Prefix conversion
- Redo-undo features at many places like editors, photoshop.
- Forward and backward feature in web browsers
- Used in many algorithms like Tower of Hanoi, tree traversals, stock span problem, histogram problem.
- Other applications can be Backtracking, Knight tour problem, rat in a maze, N queen problemand sudoku solver
- In Graph Algorithms like Topological Sorting and Strongly Connected Components

Queue

- ➢ Linear data structure
- ➢ FIFO(First In First Out)
- ➢ Mainly the following basic operations are performed in the stack:
  - **Enqueue**
  - **Dequeue**



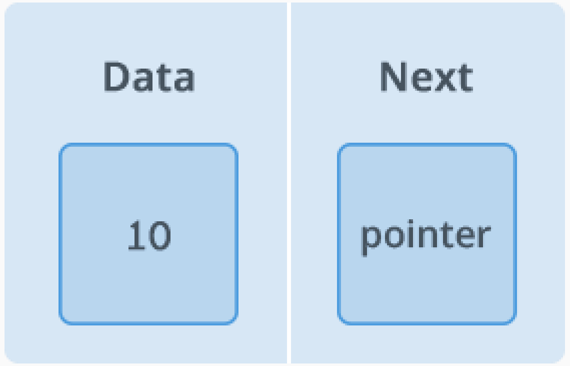
Applications of Queue

- When things don’t have to be processed immediately, but have to be processed in **F**irst **I**n **F**irst **O**ut order like Breadth First Search
- When a resource is shared among multiple consumers. Examples include CPU scheduling, Disk Scheduling.
- When data is transferred asynchronously (data not necessarily received at same rate as sent) between two processes. Examples include IO Buffers, pipes, file IO, etc.

Linked list

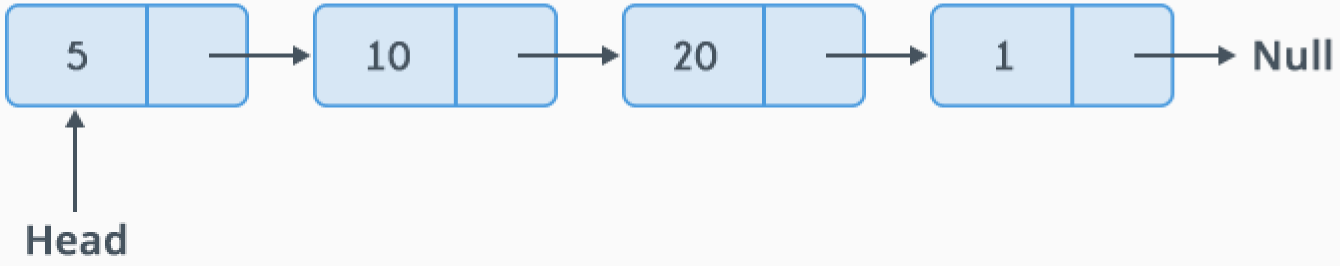
A linked list is a way to store a collection of elements. Like an array these can be character or integers. Each element in a linked list is stored in the form of a node.

Node:



A node is a collection of two sub-elements or parts. A data part that stores the element and a next part that stores the link to the next node

Linked List



A linked list is formed when many such nodes are linked together to form a chain. Each node points to the next node present in the order. The first node is always used as a reference to traverse the list and is called HEAD. The last node points to NULL.

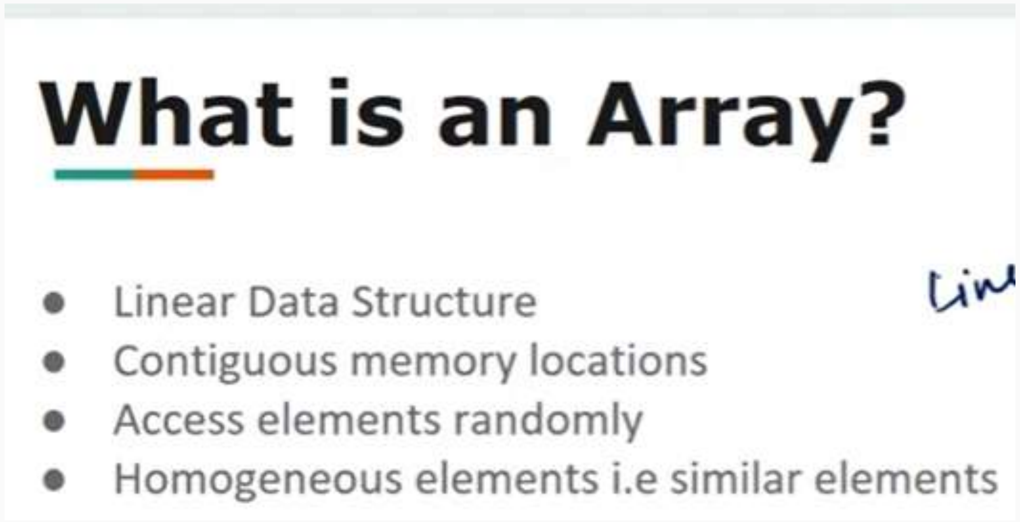
Declaration in C

```
struct LinkedList{
    int data;
    struct LinkedList *next;
 };

 typedef struct LinkedList *node; //Define node as pointer of data type struct LinkedList

 node createNode(){
     node temp; // declare a node
     temp = (node)malloc(sizeof(struct LinkedList)); // allocate memory using malloc()
     temp->next = NULL;// make next point to NULL
     return temp;//return the new node
 }
```

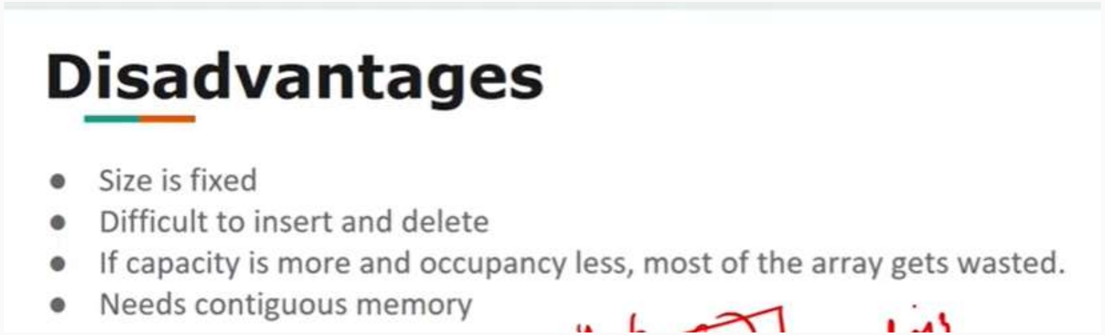
typedef is used to define a data type in C.
malloc() is used to dynamically allocate a single block of memory in C, it is available in the header file stdlib.h.
sizeof() is used to determine size in bytes of an element in C. Here it is used to determine size of each node and sent as a parameter to malloc.
The above code will create a node with data as value and next pointing to NULL



What is an Array?

Lin

- Linear Data Structure
- Contiguous memory locations
- Access elements randomly
- Homogeneous elements i.e similar elements



Disadvantages

- Size is fixed
- Difficult to insert and delete
- If capacity is more and occupancy less, most of the array gets wasted.
- Needs contiguous memory

[illegible]