Curriculum for the course of

DATA

STRUCTURE

Course Instructor –

**Learning Objectives**

This course will serve as a comprehensive introduction to various topics in data Structure. This course is a proper blend of theory, mathematical derivations and the practical hands on session for each and every concept. Throughout the course participants will learn and work on various datatypes, algorithms and it’s applications.

**Learning Outcomes**

At the end of the course participants should be able to -

* Solve problems related to graph, trees, lists.
* Solve time complexity and space complexity problems.

**Prerequisites**

There are no prerequisites required for the course. But a little knowledge of programming and high school mathematics is a benefit for candidate.

**Schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Topic** |  | **Total Duration** |
| 1 | Introduction, Array, String, Recursion | 2 - weeks |  |
| 2 | Linked List, Stack, Queue | 2 - weeks |  |
| 3 | Searching, Sorting, Hashing | 2 - weeks |  |
| 4 | Trees - Basic Terminology, Binary Trees, Binary Trees Representation, Algebraic Expressions, Complete Binary Trees, Binary Search Tree etc. | 1 - weeks |  |
| 5 | Graphs - Multi-Graphs, Sequential representation of graphs, Adjacent Matrices, Traversal, Connected Component, Spanning Tree etc. | 1 - weeks |  |

**CURRICULUM**

##### Introduction

##### Basic Terminology

##### Algorithm Complexity

##### Time-Space trade-off

##### Array

1. Array Introduction
2. Algorithm Complexity
3. Time-Space trade-off

##### Strings

##### String Introduction

##### String operation

##### Practice Questions

#### Recursion

1. Recursion Definition
2. Finding the complexity of Recursion
3. Tower of Hanoi problem
4. Backtracking
5. Practice Question

#### Linked List

1. S​ingle Linked list

Introduction

Creating a Linked List

Traversing a Linked List

Adding a node in Front

Adding a node in last

Adding a node in middle

Deleting a node in front

Deleting a node in middle

Deleting a node in last

Reversing a linked list

Check whether a linked list is a Palindrome or not

Detect a loop in a linked list

Find middle element in a linked list

1. Doubly Linked List

Introduction

Insertion

Deletion

Reverse a linked List

1. Circular Linked List

Introduction

Traversal

#### Stack

​​Introduction

Operation on stack – Push and Pop

Array Representation of Stack

Linked Representation Of Stack

Queue using Stacks

Reverse a stack using recursion

1. **Application of stack**

Conversion Infix to Postfix

Infix to prefix

Postfix to Infix

Prefix to Infix

Prefix to Postfix

#### Queues:

1. Introduction
2. Operations on Queue:

Create

Add

Delete

Full

Empty

1. Reversing a Queue
2. Reversing a queue using recursion
3. Circular Queues
4. D-Queues
5. Priority Queues

#### Searching

1. ​Linear Search
2. Binary Search
3. Comparison and analysis

#### Sorting

1. ​​Bubble Sort
2. Insertion Sort
3. Selection Sort
4. Merge Sort
5. Quick Sort
6. Heap Sort
7. Comparison and analysis

#### Hashing

1. ​Introduction
2. Hash Table
3. Hash Functions
4. Hash Table Implementation

#### Trees

1. Basic Terminology
2. Binary Trees
3. Binary Trees Representation
4. Algebraic Expressions
5. Complete Binary Trees
6. Extended Binary Trees
7. Traversing Binary Trees
8. Construct Tree from given Inorder and Preorder traversals.
9. Construct a tree from Inorder and Level order traversals.
10. Check for Children Sum Property in a Binary Tree.
11. Check if a given Binary Tree is SumTree.

1. **Binary Search Tree(BST)**
2. Insertion
3. Deletion
4. Find the node with minimum value in a Binary Search Tree.
5. A program to check if a binary tree is BST or not.
6. Lowest Common Ancestor in a Binary Tree.
7. Complexity of Search algorithm
8. Path Length

#### Graphs

1. Basic Terminology
2. Representations
3. Graphs
4. Multi-Graphs
5. Sequential representation of graphs
6. Adjacent Matrices
7. Traversal
8. Connected Component
9. Spanning Tree
10. Minimum Cost Spanning Tree

### Alogorithm

### Analysis of Algorithm

### Searching of Algorithm

### Sorting of Algorithm

### Greedy Algorithm

### Dynamic Programming

### Divide and Conquer

### Backtracking. Branch and Bound