GeeksforGeeks

DSA PYTHON

Self Paced

Detailed
Course Syllabus

DATA STRUCTURE (BASICS)

- **ANALYSIS OF ALGORITHMS**
  - Analysis of Algorithms (Background)
  - Asymptotic Analysis
  - Order of Growth
  - Best, Average and Worst Cases
  - Asymptotic Notations
  - Big O Notation
  - Omega Notation
  - Theta Notation
  - Analysis of Common Loops
  - Analysis of multiple loops
  - Analysis of Recursion
  - Recursion Tree method for solving recurrences
  - More example recurrences
  - Upper bound using Recursion tree method
  - Space Complexity

- **MATHEMATICS**
  - Python DSA - Count Digits
  - Palindrome Number
  - Factorial of a number
  - Trailing Zeros in Factorial
  - GCD and HCF of two numbers
  - LCM of two numbers
  - Check for Prime
  - Prime Factors
  - All Divisors of a Number
  - Sieve of Eratosthenes
  - Computing Power
  - Iterative Power

Practice Problems

This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

Course Contents

- **LIST**

  - List (Dynamic Sized Array) Introduction
  - Working of List in Python
  - Average or Mean of a List
  - Separate Even and Odd
  - Get Smaller Elements
  - Slicing (List,Tuple And String)
  - Comprehensions in Python
  - Largest Element in a List
  - Second Largest Element in a list
  - Check if a list is Sorted
  - Find the only Odd
  - Reverse a List in Python
  - Remove duplicates from sorted array
  - Move Zeros to End
  - Leaders in an Array problem
  - Frequencies in a sorted array
  - Left Rotate a List by one

  Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **RECURSION**

  - Applications of Recursion
  - Writing Base Cases in Recursion
  - Tail Recursion
  - Practice For Recursion (Part 1)
  - Practice For Recursion (Part 2)
  - Print N to 1 using Recursion in Python
  - Print 1 to N using Recursion in Python
  - Sum of Natural Numbers Using Recursion
  - Sum Of Digits Using Recursion
  - Palindrome Check using Recursion

  Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

Course Contents

- **SEARCHING**

  - Binary Search in Python
  - Recursive Binary Search in Python
  - Analysis of Binary Search
  - Index of first occurrence in a sorted array
  - Index of Last Occurrence
  - Count Occurrences in a Sorted Array
  - Count 1s in a Sorted Binary Array

  Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **SORTING**

  - Sorting in Python
  - List Sort in Python
  - Sorted in Python
  - Stability in Sorting Algorithm
  - Bubble Sort
  - Selection Sort
  - Insertion Sort in Python
  - Merge Sort Algorithm
  - Merge Two Sorted Arrays
  - Merge Subarrays
  - Count inversions in Array
  - Merge Sort Analysis
  - Quick Sort Introduction
  - Partition a Given Array
  - Lomuto Partition
  - Hoare's Partition
  - Quick Sort using Lomuto Partition
  - Quick Sort using Hoare's Partition
  - Analysis of Quick Sort
  - Space Analysis of Quick Sort
  - Heap Sort

  Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **HASHING**

  -Introduction to Hashing
  - Hashing Application
  - Direct Address Table
  - Hashing Functions
  - Collision Handling
  - Chaining
  - Implementation of Chaining in Python
  - Open Addressing
  - Double Hashing
  - Implementation of Open Addressing in Python
  - Chaining vs Open Addressing
  - Set in Python
  - Dictionary in Python
  - Count Distinct Elements in a List
  - Frequencies of array elements

  Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **STRING**

  - Strings in Python
  - Escape Sequences and Raw Strings
  - Formatted String in Python
  - String Comparison in Python
  - String Operations Part (1)
  - String Operations Part (2)
  - Reverse A String in Python
  - Check if string is rotated
  - Check For Palindrome In Python
  - Check if a String is Subsequence of Other
  - Check for Anagram in Python
  - Leftmost Repeating Character
  - Leftmost Non-Repeating Element
  - Reverse words in a string

  Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

Course Contents

- **LINKED LIST**

  - Video - Problems with Array Data Structure
  - Linked List Introduction in Python
  - Simple Linked List Implementation in Python
  - Applications of Linked List
  - Traversing a Linked List in Python
  - Search in Linked List
  - Insert At The Bigenning of Linked list in Python
  - Insert at The End Of Linked List
  - Insert at Given Position in Singly Linked list
  - Delete First Node Of Linked List in Python
  - Delete Last Node of Linked List
  - Delete a node with pointer given to it
  - Sorted Insert Linked List in Python
  - Middle of Linked List
  - Nth Node From end of Linked List
  - Remove duplicates from a sorted Singly Linked List
  - Reverse a Linked List In Python
  - Recursive Reverse A Linked List (Part 1)
  - Recursive Reverse A Linked List (Part 2)

  Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **CIRCULAR LINKED LIST**

  - Circular Linked List in Python
  - Circular Linked List (Advantages & Disadvantages)
  - Circular Linked List traversal
  - Insert at the Beginning of Circular Linked List
  - Insert at The End of A Circular Linked List
  - Delete Head of circular Linked List
  - Delete Kth Node of Circular Linked List

  Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **DOUBLY LINKED LIST**

  - Doubly Linked List in Python
  - Singly Vs Doubly Linked List (Advantages & Disadvantages)
  - Insert at the Beginning of DLL in Python
  - Insert at the End of DLL in Python
  - Delete Head of A Doubly Linked List
  - Delete Last Node of DLL in Python
  - Reverse A Doubly Linked Llist in Python

  Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **STACK**

  - Stack Data Structure
  - Stack in Python
  - Linked List Implementation of Stack in Python
  - Stack Applications
  - Check for Balanced Parenthesis in Python

  Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **QUEUE**

  - Queue in Python
  - Queue Data Structure
  - Application of Queue Data structure
  - Implementation of Queue using Array
  - Linked List Implementation of Queue in Python

  Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

Course Contents

- **DEQUE**

  - Deque Introduction
  - Deque Applications
  - Deque in Python
  - List Implementation of Deque in Python
  - Linked List Implementation of Deque

  Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **TREE**

  - Tree Data Structure
  - Application of Tree
  - Binary Tree in Python
  - Tree Traversal
  - Inorder Traversal in Python
  - Preorder Traversal in Python
  - Postorder Traversal in Python
  - Height of Binary Tree
  - Print Node at K distance
  - Level Order Traversal
  - Size of Binary Tree in Python
  - Maximum in Binary Tree
  - Iterative Inorder Traversal
  - Iterative Preorder Traversal
  - Iterative Preorder Traversal (Space Optimized)

  Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **BINARY SEARCH TREE**

  -Binary Search Tree(Background)
  - Binary Search Tree(Introduction)
  - Search in BST in Python
  - BST insert in Python
  - BST Delete in Python
  - Floor in BST (Problem and Solution Idea)

- BST Floor in Python
- Ceiling in BST in Python
- Self Balancing BST
- AVL Tree
- Red Black Tree
- Applications of BST

Practice Problems

This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

• HEAP

- Binary Heap Introduction
- Heap Python Implementation (Introduction)
- Binary Heap Insert
- Binary Heap (Extract min and Heapify)
- Decrease Key and Delete Operations
- Build Heap
- Heap Sort
- Heapq in Python

Practice Problems

This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

DATA STRUCTURE - ADVANCED

• BIT MAGIC

- Bitwise Operator in Python - Part 1
- Bitwise Operator in Python - Part 2
- Check Kth bit is set or not
- Count Set Bits
- Power of Two
- One Odd Occuring
- Two Odd Occuring
- Power Set Using Bitwise

Practice Problems

This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **LIST**

  - Left Rotate by d Places
  - Maximum difference
  - Stock Buy &Sell Part 2
  - Trapping Rainwater
  - Maximum Consecutive 1s
  - Longest even odd subarray
  - Majority element
  - Minimum Consecutive flips
  - Sliding Window Technique
  - Maximum subarray sum
  - Maximum circular sum subarray
  - Prefix Sum Technique

  ### Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **RECURSION**

  - Subset of a given string
  - Printing all Permutations
  - Tower of Hanoi in Python
  - Josephus Problem in Python
  - Subset sum problem
  - Rope Cutting Problem

  ### Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **SEARCHING**

  - Search in Sorted Rotated Array
  - Median of two sorted arrays
  - Repeating Elements Part (1)
  - Repeating Elements Part (2)
  - Allocate Minimum Pages (Naive Method)
  - Allocate Minimum Pages (Binary Search)

Practice Problems

This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **SORTING**

  - Tail Call Elimination in Quick Sort
  - Kth Smallest
  - Minimum Difference in an Array
  - Chocolate Distribution Problem
  - Sort an array with two types of element
  - Sort an array with three types of elements
  - Merge overlapping intervals
  - Meeting the maximum guests
  - Counting Sort
  - Cycle Sort
  - Bucket Sort
  - Radix Sort
  - Overview of sorting algorithm

  #### Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **HASHING**

  - Intersection of two arrays
  - Union of two unsorted arrays
  - Pair with given sum in unsorted array
  - Subarray with 0 sum in Python
  - Check for Palindrome Permutation
  - Subarray with given sum
  - Longest Subarray with equal number of 0s and 1s
  - Longest common span with same sum in binary array
  - Longest Consecutive Subsequence
  - Longest Subarray with given sum
  - More than n/k Occurences (O(nk) solution)

  #### Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

Course Contents

- **STRING**

  - Overview of Pattern Searching
  - Pattern Searching in Python
  - Naive Pattern Searching
  - Improved Naive Pattern Searching for Distinct
  - Rabin Karp Algorithm
  - KMP Algorithm (Part 1 : Constructing LPS Array)
  - KMP Algorithm (Part 2 : Complete Algorithm)
  - Anagram Search
  - Lexicographic rank of a String
  - Longest Substring With Distinct Characters

  ### Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **LINKED LIST**

  - Reverse a linked list in groups of size k
  - Detect loop using floyd's cycle detection algorithm
  - Detect and remove loop in linked list
  - Intersection Point of two linked list
  - Segregate even odd nodes of linked list
  - Pairwise swap nodes of linked list
  - Clone a linked list using a random pointer
  - LRU Cache Design
  - Merge two sorted linked lists
  - Palindrome Linked List

  ### Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **STACK**

  - Two stacks in an array
  - K Stacks in an array
  - Previous Greater Element
  - Next Greater Element
  - Stock span problem
  - Largest Rectangular Area in a Histogram (Part 1)

- Largest Rectangular Area in a Histogram (Part 2)
- Largest Rectangle with all 1's
- Stack with getMin() in O(1)
- Design a stack with getMin() in O(1) space
- Infix to Postfix (Simple Solution)
- Infix to Postfix (Efficient Solution)
- Evaluation of Postfix
- Infix to Prefix (Simple Solution)
- Infix to Prefix (Efficient Solution)
- Evaluation of Prefix

Practice Problems

This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

• QUEUE

- (Queue and Deque)
- Queue Implementation using Circular List
- Implementing stack using queue
- Reversing a Queue
- Generate numbers with given digits
- Design a data structure with min/max operations
- Maximums of all subarrays of size k
- First Circular Tour

Practice Problems

This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

• TREE

- Level Order Traversal by Line - Part 1
- Level Order Traversal by Line - Part 2
- Check for Balanced Binary Tree
- Maximum Width of Binary Tree
- Convert Binary Tree to Doubly Linked List
- Construct Binary Tree from Inorder and Preorder
- Tree Traversal in Spiral Form
- Diameter of a Binary Tree

- LCA of Binary Tree (Part 1)
- LCA of Binary Tree (Part 2)
- Burn a Binary Tree from a Leaf
- Count nodes in a Complete Binary Tree
- Serialize and Deserialize a Binary Tree

Practice Problems

This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

• BINARY SEARCH TREE

- Ceiling on the left side in an array
- Find Kth Smallest in BST
- Check for BST
- Fix BST with Two Nodes Swapped
- Pair Sum with Given BST
- Vertical Sum in a Binary Tree
- Vertical Traversal of binary tree
- Top View of Binary Tree
- bottom view of binary tree

Practice Problems

This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

• HEAP

- Sort K Sorted Array
- Purchase Maximum Items
- K Largest Elements
- K Closest Elements
- Merge K Sorted Arrays
- Median of a Stream

Practice Problems

This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **GRAPH**

  - Introduction to Graph
  - Graph Representation (Adjacency Matrix)
  - Graph Representation (Adjacency List)
  - Graph Adjacency List Representation in Python
  - Adjacency Matrix and List Comparison
  - Breadth First Search in Python
  - BFS for Disconnected Graph
  - Connected Components in an Undirected Graph using BFS
  - Applications of BFS
  - Depth First Search
  - DFS For Disconnected Graph
  - Connected Components in an Undirected Graph using DFS
  - Applications of DFS
  - Shortest Path in an Unweighted Graph
  - Detect Cycle in Undirected Graph
  - Detect Cycle in a Directed Graph (Part 1)
  - Topological Sorting (Kahn's BFS Based Algortihm)
  - Detect Cycle in a Directed Graph (Part 2)
  - Topological Sorting (DFS Based Algorithm)
  - Shortest Path in DAG
  - Prim's Algorithm/Minimum Spanning Tree
  - Implementation of Prim's Algorithm
  - Dijkstra's Shortest Path Algorithm
  - Implementation of Dijkstra's Algorithm
  - Kosaraju's Algorithm Part 1
  - Kosaraju's Algorithm Part 2
  - Bellman Ford Shortest Path Algorithm
  - Articulation Point
  - Bridges in Graph
  - Tarjans Algorithm
  - Kruskal's Algorithm

  Practice Problems

  This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **GREEDY**

  - Introduction to Greedy Algorithms
  - Activity Selection Problem
  - Activity selection
  - Fractional Knapsack

- Fractional Knapsack in Python
- Job Sequencing Problem
- Huffman Coding (introduction)
- Huffman Algorithms
- Python Implementation of Huffman coding

Practice Problems

This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

• BACKTRACKING

- Concept of backtracking
- Rat In a Maze
- N Queen Problem
- Sudoku Problem

Practice Problems

This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

• DYNAMIC PROGRAMMING

- Introduction to DP
- Dynamic Programming Memoization
- Dynamic Programming Tabulation
- Longest Common Subsequence (Part 1)
- Longest Common Subsequence (Part 2)
- Variation of LCS
- Coin Change Count Combinations
- Edit Distance Problem
- Edit Distance Problem DP solution
- Longest Increasing Sub sequence Problem
- Longest Increasing Subsequence O(nlogn)
- Variation of LIS (Part 1)
- Variations of LIS (Part 2)
- Maximum Cuts
- Minimum coins to make a value
- Minimum Jumps to reach the end
- 0-1 knapsack problem
- 0-1 knapsack problem DP Solution

- Optimal Strategy for a Game
- Egg Dropping Puzzle - Part 1
- Egg Dropping Puzzle - Part 2
- Count BSTs with n keys
- Maximum sum with no two consecutive
- Subset sum problem
- Subset Sum Problem (DP Solution)
- Matrix Chain Multiplication
- Matrix Chain Multiplication (DP Solution)
- Palindrome Partitioning
- Allocate Minimum Pages (Naive Method)
- Allocate Minimum Pages (DP Solution)

Practice Problems

This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

• TRIE

- Trie Data Structure (Introduction)
- Trie (Representation, Search and Insert)
- Trie Delete
- Count Distinct Rows in a Binary Matrix

Practice Problems

This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

• SEGMENT AND BINARY INDEXED TREES

- Segment Tree (Introduction)
- Constructing Segment Tree
- Range Query on Segment Tree
- Update Query on Segment Tree
- Binary Indexed Tree (Intoductiion)
- Binary Indexed Tree (An Example Problem)
- Binary Indexed Tree (Prefix Sum)
- Binary Indexed Tree (Prefix Sum Implemention)
- Binary Indexed Tree (Update Operation)

Practice Problems

This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

- **DISJOINT SET**
  - Disjoint Set Introduction
  - Find and Union Operations on Disjoint Sets
  - Union by Rank
  - Path Compression
  - Kruskal's Algorithm

Practice Problems

This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.