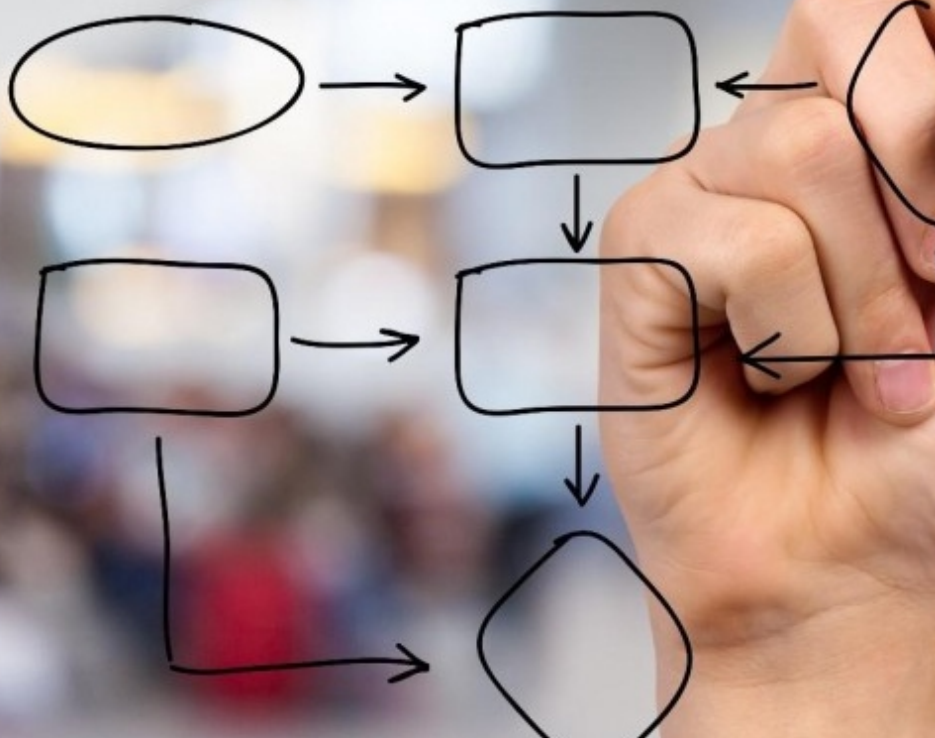


Data Structures And Algorithms With Java Syllabus



Introduction to Java

- 1.Introduction
- 2.How it works
- 3.Setup Installation
- 4.Input and Output in Java
- 5.Conditionals & Loops in Java
 - if-else
 - loops
 - Switch statements

Functions

- 1.Introduction
- 2.Scoping in Java
- 3.Shadowing
- 4.Variable Length Arguments
- 5.Overloading

Strings

- 1.Introduction
- 2.How Strings work
- 3.Comparison of methods
- 4.Operations in Strings
- 5.StringBuilder in java



What are Data Structures?

1. What are Data Structures in Java?
2. Types of Data Structures in Java
3. Advantages of Data Structures in java
4. Classification of Data Structures

Arrays

1. Introduction
2. Memory management
3. Input and Output
4. Array List Introduction
5. Searching
 - Linear Search
 - Binary Search
 - Modified Binary Search
 - Binary Search on 2D Arrays
6. Sorting
 - Insertion Sort
 - Selection Sort
 - Bubble Sort
 - Cyclic Sort



Maths For DSA

1. Introduction
2. Complete Bitwise Operators
3. Range of numbers
4. Prime numbers
5. Sieve of Eratosthenes
6. Newton's Square Root Method
7. Factors
8. Modulo properties
9. Number Theory
10. HCF / LCM
11. Euclidean algorithm

Recursion

1. Introduction
2. Flow of recursive programs - stacks
3. Why recursion?
4. Tree building of function calls
5. Tail recursion
6. Sorting
 - Merge Sort
 - Quick Sort



7. Backtracking

- Sudoku Solver
- N-Queens
- N-Knights
- Maze problems

8. Recursion String Problems

9. Recursion Array Problems

10. Recursion Pattern Problems

11. Subset Questions

Space and Time Complexity Analysis

1. Introduction

2. Comparisons of various cases

3. Solving Linear Recurrence Relations

4. Solving Divide and Conquer Recurrence Relations

5. Big-O, Big-Omega, Big-Theta Notations

6. Little Notations

7. Get equation of any relation easily - best and easiest approach

8. Complexity discussion of all the problems we do

9. Space Complexity

10. Memory Allocation of various languages

11. NP-Completeness Introduction



Object Oriented Programming

1. Introduction
2. Classes & its instances
3. this keyword in Java
4. Properties
 - Inheritance
 - Abstraction
 - Polymorphism
 - Encapsulation
5. Overloading & Overriding
6. Static & Non-Static
7. Access Control
8. Interfaces
9. Abstract Classes
10. Singleton Class
11. final, finalize, finally
12. Object Class
 - Generics
 - Exception Handling
 - Collections Framework
 - Lambda Expression
 - Enums
 - Fast IO
 - File handling



Greedy Algorithms

Stacks & Queue

1. Introduction
2. Interview problems
3. Push efficient
4. Pop efficient
5. Queue using Stack and Vice versa
6. Circular Queue

Linked List

1. Introduction
2. Fast and slow pointer
3. Cycle Detection
4. Single and Doubly LinkedList
5. Reversal of LinkedList

Dynamic Programming

1. Introduction
2. Recursion + Recursion DP + Iteration + Iteration Space Optimization
3. Complexity Analysis
4. 0/1 Knapsack
5. Subset Questions
6. Unbounded Knapsack
7. Subsequence questions
8. String DP



Trees

1. Introduction
2. Binary Trees
3. Binary Search Trees
4. DFS
5. BFS
6. AVL Trees
7. Segment Tree
8. Fenwick Tree / Binary Indexed Tree

Square Root Decomposition

Heaps

1. Introduction
2. Theory
3. Priority Queue
4. Heapsort
5. Two Heaps Method
6. k-way merge
7. Top k-elements
8. Interval problems

HashMap

1. Introduction
2. Theory - how it works
3. Comparisons of various forms



4. Limitations and how to solve
5. Map using LinkedList
6. Map using Hash
7. Count Sort
8. Radix Sort
9. Chaining
10. Probing
11. Huffman-Encoder

Subarray Questions: Sliding window, Two Pointer, Kadane & #39;s Algorithm

Tries

Graphs

1. Introduction
2. BFS
3. DFS
4. Working with graph components
5. Minimum Spanning Trees
6. Kruskal Algorithm
7. Prims Algorithm
8. Dijkstra's shortest path Algorithm
9. Topological Sort
10. Bellman ford
11. A* pathfinding Algorithm



Bitwise + DP

Extended Euclidean algorithm

Modulo Multiplicative Inverse

Linear Diophantine Equations

Matrix Exponentiation

Mathematical Expectation

Fermat's Theorem

Wilson's Theorem

Lucas Theorem

Chinese Remainder Theorem

NP-Completeness

