*DSA Syllabus*

**Introduction to Data Structures and Algorithms**

1. Definition and Importance
2. Algorithm Complexity
3. Time Complexity (Big-O Notation)
4. Space Complexity
5. Asymptotic Analysis
6. Best, Worst, and Average Case Analysis

**Basic Data Structures**

1. **Arrays**
   * Definition, Memory Representation
   * Applications and Limitations
2. **Linked Lists**
   * Singly Linked List
   * Doubly Linked List
   * Circular Linked List
   * Operations: Insertion, Deletion, Traversal, Searching
3. **Stacks**
   * Definition and Properties
   * Stack Operations: Push, Pop, Peek
   * Applications: Expression Evaluation, Function Calls
4. **Queues**
   * Simple Queue
   * Circular Queue
   * Double-Ended Queue (Deque)
   * Priority Queue
   * Applications: Scheduling, Buffers

**Recursion**

1. Concept and Examples
2. Recursion vs Iteration
3. Tail Recursion
4. Solving Recurrence Relations
5. Applications: Factorial, Fibonacci Sequence, Tower of Hanoi

**Trees**

1. **Binary Trees**
   * Definition and Properties
   * Types: Full, Complete, Perfect, Balanced
   * Traversal Techniques: Preorder, Inorder, Postorder, Level Order
2. **Binary Search Tree (BST)**
   * Insertion, Deletion, Searching
   * Time Complexity and Applications
3. **Heaps**
   * Min-Heap and Max-Heap
   * Heap Operations: Insertion, Deletion, Heapify
   * Applications: Priority Queue, Heap Sort
4. **Balanced Trees**
   * AVL Trees (Rotations)
   * Red-Black Trees
5. **Trie**
   * Structure and Uses: Prefix Trees, Autocomplete

**Graphs**

1. Graph Representation: Adjacency Matrix, Adjacency List
2. Graph Traversal: Depth First Search (DFS), Breadth First Search (BFS)
3. **Graph Algorithms**
   * Shortest Path Algorithms: Dijkstra’s, Bellman-Ford
   * Minimum Spanning Tree: Prim’s, Kruskal’s
   * Topological Sorting
   * Strongly Connected Components (Kosaraju’s Algorithm)
4. Applications of Graphs: Networks, Web Crawling

**Hashing**

1. Hash Function and Characteristics of a Good Hash Function
2. Collision Handling: Chaining, Open Addressing
3. Applications: Hash Tables, Dictionaries, Maps
4. Hashing Techniques: Linear Probing, Quadratic Probing, Double Hashing

**Sorting Algorithms**

1. **Comparison-Based Sorting**
   * Bubble Sort
   * Selection Sort
   * Insertion Sort
   * Merge Sort
   * Quick Sort
2. **Non-Comparison Sorting**
   * Counting Sort
   * Radix Sort
   * Bucket Sort
3. Sorting Algorithm Analysis: Best, Worst, and Average Case Time Complexity
4. Stable vs Unstable Sorting

**Searching Algorithms**

1. Linear Search
2. Binary Search (Recursive and Iterative Approaches)
3. Applications and Complexity
4. Search in Rotated Arrays
5. Interpolation Search

**Greedy Algorithms**

1. Introduction to Greedy Technique
2. Famous Problems:
   * Activity Selection Problem
   * Fractional Knapsack Problem
   * Huffman Coding
   * Job Scheduling Problem

**Divide and Conquer Algorithms**

1. Introduction and Concept
2. Algorithms:
   * Binary Search
   * Merge Sort
   * Quick Sort
   * Matrix Multiplication (Strassen's Algorithm)

**Dynamic Programming**

1. Introduction to DP: Memoization vs Tabulation
2. Famous DP Problems:
   * Fibonacci Numbers
   * Longest Common Subsequence
   * 0/1 Knapsack Problem
   * Coin Change Problem
   * Matrix Chain Multiplication

**Backtracking**

1. Introduction and Concept
2. Famous Problems:
   * N-Queens Problem
   * Sudoku Solver
   * Hamiltonian Path Problem
   * Subset Sum Problem

**Advanced Data Structures**

1. Segment Trees: Build, Query, Update
2. Fenwick Tree (Binary Indexed Tree)
   * Applications in Range Queries
3. Disjoint Set Union (Union-Find)
   * Union by Rank and Path Compression
   * Applications in Graph Algorithms
4. Sparse Table
   * Applications in Range Queries (Range Minimum/Maximum Queries)

**String Algorithms**

1. String Matching:
   * Naive String Matching
   * Knuth-Morris-Pratt (KMP) Algorithm
   * Rabin-Karp Algorithm
   * Boyer-Moore Algorithm
2. Suffix Arrays: Construction and Applications
3. Trie and Aho-Corasick Algorithm

**Complexity Classes and Advanced Topics**

1. NP-Completeness: P vs NP Problem
2. NP-Hard and NP-Complete Problems
3. Reductions
4. Approximation Algorithms
   * Greedy and Dynamic Programming-Based Approximations
   * Examples: Traveling Salesman Problem, Vertex Cover