UNIT I — Introduction to Algorithms

1. Algorithm Basics 📖

- Definition and properties
- Characteristics of a good algorithm

2. Analysis Framework 📖

- Time Complexity vs Space Complexity
- Best Case / Worst Case / Average Case
- How to measure efficiency

3. Asymptotic Notations 📖

- Overview and why we use them
- Big O Notation — upper bound — definition + example
- Omega Notation — lower bound — definition + example
- Theta Notation — tight bound — definition + example

4. Solving Recurrence Relations

- What is a recurrence relation 📖
- Master Theorem — 3 cases with examples 📖🔢
- Substitution Method — steps with example 📖🔢

5. Searching Algorithms

- Linear Search — how it works + complexity 📖💻
- Binary Search — how it works + complexity 📖💻

UNIT II — Divide and Conquer & Greedy

1. Divide and Conquer 📖

- Principle and how it works
- Control abstraction
- Advantages

2. Merge Sort 💻🔢

- How it divides and merges

- Step by step sorting on a given array
- Time complexity — O(n log n) all cases

3. Quick Sort 💻 🔢

- Pivot selection and partitioning
- Step by step sorting on a given array
- Best case O(n log n) / Worst case $O(n^2)$

4. Divide and Conquer vs Greedy 📖

- Comparison table — 4 to 5 points
- When to use which

5. Greedy Strategy 📖

- Principle and how it works
- Control abstraction
- Characteristics — local optimal choice

6. Fractional Knapsack 📖 🔢

- Greedy approach using profit/weight ratio
- Sort by ratio, pick greedily
- Step by step solving on given items

7. Job Sequencing with Deadlines 📖 🔢

- Sort jobs by profit descending
- Check feasibility slot by slot
- Step by step solving on given jobs

8. Huffman Encoding 📖 🔢

- Build min-heap from frequencies
- Combine two lowest nodes repeatedly
- Generate codes from tree
- Calculate compression ratio

UNIT III — Graphs + Symbol Tables

1. Graph Basic Terminology 📖

- Vertices, edges, degree

- Vertices, edges, degree
- Path, cycle, connected graph
- Weighted vs unweighted

2. Types of Graphs 📖

- Directed graph vs Undirected graph
- Differences with examples

3. Graph Representations 📖

- Adjacency Matrix — how it works + advantages/disadvantages
- Adjacency List — how it works + advantages/disadvantages
- Comparison table between the two

4. Graph Traversals 💻🔢

- BFS — Breadth First Search — level by level using queue
- DFS — Depth First Search — deep first using stack/recursion
- Traversal on a given graph step by step

5. Prim's Algorithm 📖💻🔢

- Find Minimum Spanning Tree
- Start from any vertex, always pick minimum edge
- Step by step on a given weighted graph

6. Kruskal's Algorithm 📖💻🔢

- Find Minimum Spanning Tree
- Sort all edges by weight, pick smallest without forming cycle
- Step by step on a given weighted graph

7. Prim's vs Kruskal's 📖

- Comparison table — 4 to 5 points
- When to use which

8. Dijkstra's Algorithm 📖💻🔢

- Single source shortest path
- Greedy approach — always pick minimum distance vertex
- Step by step table on a given graph

9. Topological Sorting 📖🔢

- Only for Directed Acyclic Graph (DAG)

- Find valid ordering of vertices
- Step by step on a given DAG

10. Symbol Tables 📖

- What is a symbol table and why it is used
- Static Tree Table — structure and properties
- Dynamic Tree Table — structure and properties
- Difference between static and dynamic

11. AVL Tree 📖 🔢

- What is a height balanced tree
- Balance Factor — definition and formula (BF = height of left subtree − height of right subtree)
- Valid balance factors are −1, 0, +1 only
- When imbalance occurs — 4 rotation types
    - LL Rotation
    - RR Rotation
    - LR Rotation
    - RL Rotation
- Insert elements one by one and apply rotations