

Week 3 DSA

Tree (only theory)

- N-Ary Tree, Basic Implementation
- Classification of Trees
 - Binary Tree
 - Binary Search Tree
 - AVL Tree
 - Red-Black Tree
 - B-Tree
 - Trie
 - Heap
 - Ternary Search Tree
- Classification of Trees (based on Structure)
 - Complete Binary Tree
 - Full Binary Tree
 - Perfect Binary Tree
 - Balanced Binary Tree
 - Degenerate Binary Tree
 - Skewed Binary Tree
 - Height Balanced Binary Tree

Binary Search Tree

- Implementation
 - insert(), find(), delete()
 - Traversal
 - Depth-First-Search
 - In-order

- Pre-order
- Post-order
- Breadth-First-Search
- Additionally
 - Validate BST
 - Find height
 - Find minimum/maximum.
- Applications
- Time complexity

Trie

- Implementation
 - insert(), search(), startsWith()
 - Find all words in Trie.
- Applications
- Time complexity

Heap

- Implementation
 - Minheap, Maxheap
 - insert (enqueue), delete (dequeue)
 - sink down, bubble up
 - Array representation (and equations to find parent/child)
- Priority Queue
- Heap Sort
 - Heapify
- Applications
- Time complexity

Graph

- Implementation
 - Adjacency Matrix
 - Adjacency List
 - Traversal
 - Depth-First-Search
 - Breadth-First-Search
 - Shortest path between two nodes.
 - Dijkstra's Algorithm
 - Bellman-Ford Algorithm
 - Whether path exists between two nodes.
- Types of Graph
 - Connected, Disconnected
 - Weighted, Unweighted
 - Directed, Undirected
 - Cyclic, Acyclic
- Minimum/Maximum Spanning Tree
 - Kruskal's Algorithm
 - Prim's Algorithm
- Applications
- Time complexity