Categorized DSA Problems from Previous Year Papers

Trees

- Binary Search Tree (BST): Create, locate the smallest key, and count parent nodes (excluding root).
- AVL Tree: Construct AVL tree with specified keys, showing violations and adjustments.
- B+ Tree: Construct a B+ tree of a given order and sequence of insertions.
- Binary Tree: Check if the root node is balanced.
- Define and illustrate strict and full binary trees.

Graphs

- Dijkstra's Algorithm: Find the shortest path from a specific node in a graph representing cities and flights.
- Depth-First Search (DFS) and Breadth-First Search (BFS): Compute DFS and BFS trees from a specified vertex in an undirected graph.
- Implement DFS traversal using adjacency matrix, beginning from a specified vertex.
- Define complete and strongly connected directed graphs.

Linked Lists

- Singly Linked List: Insert a node at the end, remove a node at a given position, and reverse the list.
- Integer Linked List: Delete all negative numbers from the linked list.
- Doubly Linked List: Swap the k-th and (k+1)-th nodes in a doubly linked list.

Queues

- Circular Queue: Implement a circular queue ADT without global storage for pointers.
- Double-Ended Queue (Deque): Implement a deque with insertion and deletion from both ends using an array.

- Two Queues in One Array: Store two queues in a single array, handling overflow and underflow conditions for each.

Heap Sort

- Perform heap sort on a list of keys in descending order, showing heap at each sorting stage.

Most Repeated and Important Questions

- Implementing and traversing BSTs and AVL trees.
- Graph traversal techniques (DFS and BFS).
- Queue implementations, including circular and double-ended queues.
- Heap sort operations and understanding sorting steps.