1. Arrays:

  - Contiguous memory, fixed size, random access.
  - Operations: traversal, insertion, deletion, searching.

2. Linked Lists:

  - Dynamic size, sequential access.
  - Types: Singly, Doubly, Circular.
  - Useful for efficient insertions/deletions.

3. Stacks and Queues:

  - Stack: LIFO, operations: push, pop, peek.
  - Queue: FIFO, types: simple, circular, deque, priority.

4. Trees:

  - Hierarchical structure.
  - Binary Tree, Binary Search Tree (BST), AVL Tree (self-balancing), Heap.

5. Graphs:

  - Set of vertices connected by edges.
  - Representations: adjacency matrix/list.
  - Traversals: BFS, DFS.
  - Algorithms: Dijkstra, Kruskal, Prim.

6. Sorting Algorithms:

  - Bubble, Insertion, Selection ($O(n^2)$).
  - Merge Sort, Quick Sort ($O(n \log n)$).

7. Searching Algorithms:

  - Linear Search: $O(n)$.
  - Binary Search: $O(\log n)$, needs sorted array.

8. Recursion and Dynamic Programming:

- Divide problem into subproblems.
- Memoization and tabulation.

9. Time and Space Complexity:

- Big O notation to express performance.
- Best, Average, Worst cases.

10. Useful Formulae:

- Number of leaf nodes in a binary tree: L = N + 1 (for N internal nodes)
- Time complexities:
  - Binary Search: O(log n)
  - Merge Sort: O(n log n)
  - Quick Sort Avg: O(n log n), Worst: O(n^2)