# Data Structures & Algorithms (DSA) Proficiency

I have a solid foundation in **Data Structures and Algorithms**, with deep practical knowledge and implementation experience in **Java** and **C**. My skillset covers:

- **Core Data Structures**: Arrays, Linked Lists (Singly, Doubly, Circular), Stacks, Queues (including Priority Queues and Deques)

- **Trees & Graphs**: Binary Trees, Binary Search Trees, AVL Trees, Heaps, Tries, and Graphs using both adjacency lists and matrices

- **Searching & Sorting Algorithms**: Binary Search, Merge Sort, Quick Sort, Heap Sort, Counting Sort, and others, with attention to time and space complexity optimization

- **Advanced Concepts**: Hashing, Sliding Window Technique, Two Pointers, Backtracking, Recursion, Bit Manipulation, Union-Find, and Dynamic Programming

- **Graph Algorithms**: BFS, DFS, Dijkstra's, Floyd-Warshall, Kruskal's, Prim's, and Topological Sorting

- **Problem Solving**: I've solved numerous problems on platforms like LeetCode, Codeforces, and HackerRank, improving both algorithmic thinking and code efficiency.

I apply these concepts in building optimized and scalable solutions, with a strong emphasis on clean code, edge case handling, and performance tuning. Whether it's competitive programming or real-world applications, I consistently aim to write logical, maintainable, and efficient code.