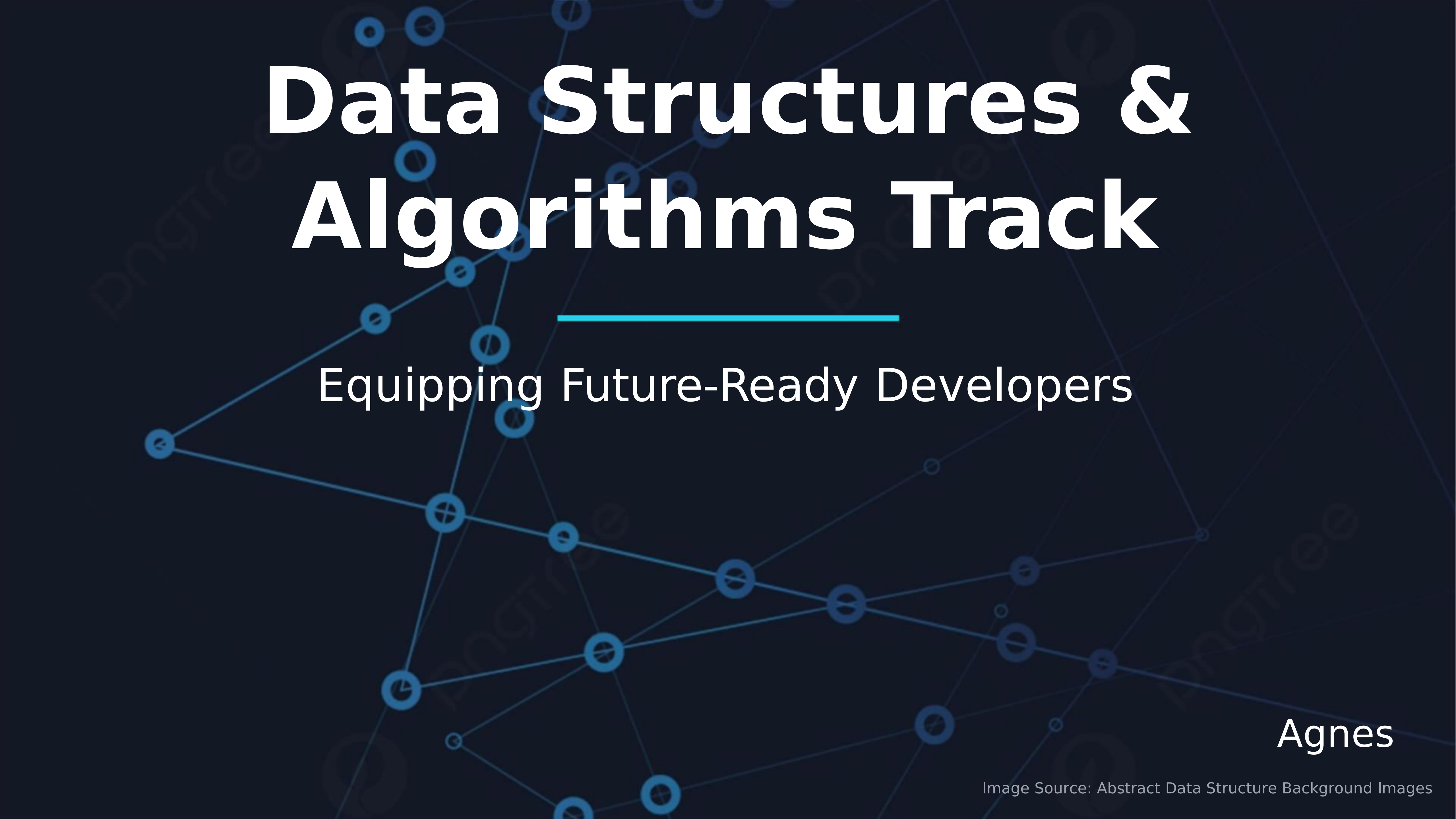


Data Structures & Algorithms Track

Equipping Future-Ready Developers

Agnes

Image Source: Abstract Data Structure Background Images

Program Overview: 3 Months Intensive

Months 1-2: Online Learning

Foundational & Core Concepts

- **Basics of Programming & DSA:** Comprehend core data structures and algorithms.
- **Algorithmic Techniques:** Deep dive into divide and conquer, greedy algorithms, and dynamic programming.
- **Problem-Solving Foundations:** Build a

→

Month 3: Offline Immersion

Project Application & Industry Immersion

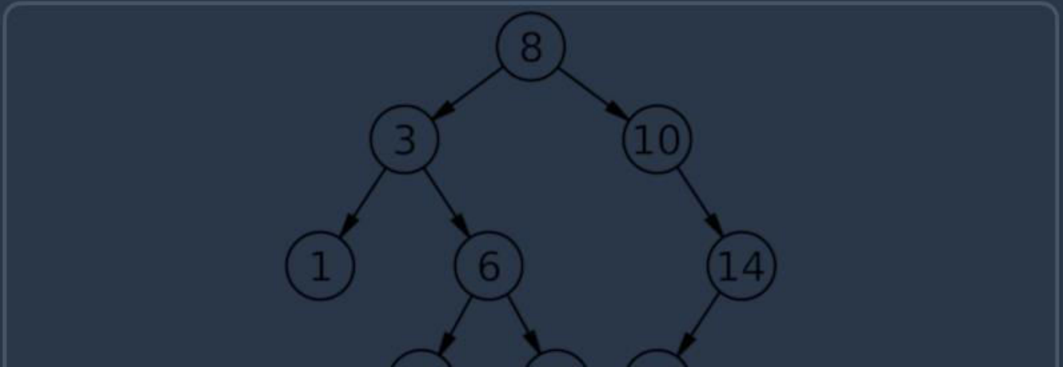
- **Hands-on Project Application:** Apply concepts through real-world projects and challenges.
- **Competitive Edge & SDE Interviews:** Enhance skills to gain a competitive edge for top company roles.

Phase 1: Foundational & Core Concepts (Months 1-2 Online)

Theoretical Understanding

Building a strong theoretical understanding of DSA fundamentals.

- **Core Principles:** Grasp programming basics and fundamental concepts in DSA.
- **Algorithmic Analysis:** Analyze algorithm efficiency and complexity using Big O notation.
- **Key Data Structures:** Deep dive into Arrays, Linked Lists, Trees, and Graphs.
- **Advanced Paradigms:** Understand Divide & Conquer, Greedy, and Dynamic Programming.



Practical Skills

Developing practical skills in implementation using chosen programming languages.

- **Hands-on Practice:** Apply theory through practical exercises and coding challenges.
- **Efficient Implementation:** Develop proficiency in Python, Java, or C++.
- **Problem-Solving Platforms:** Engage with problems on LeetCode, GeeksforGeeks, and Codeforces.
- **Targeted Problems:** Focus on searching, sorting, and string manipulation problems.

Month 1: Core Data Structures & Basic Algorithms

Week 1 Week 2 Week 3 Week 4

Week 1: DSA Fundamentals & Analysis

Foundational Concepts: Lay the groundwork for how data is organized and processed, covering the critical importance of DSA in interviews and development.

Algorithmic Performance: Master evaluating algorithm efficiency with Complexity Analysis and Big O notation to compare performance.

Week 2: Dynamic Data with Linked Lists

Core Concepts: Delve into Linked Lists, their advantages over static arrays, and essential operations like insertion, deletion, and traversal.

Varieties of Linked Lists: Explore Singly, Doubly, and Circular Linked Lists, covering their practical applications and implementation specifics.

Week 3: Ordered Collections - Stacks & Queues

Week 4: Recursion & Backtracking

Recursion: Focus on functions calling themselves, detailing base

Month 2: Intermediate to Advanced DSA

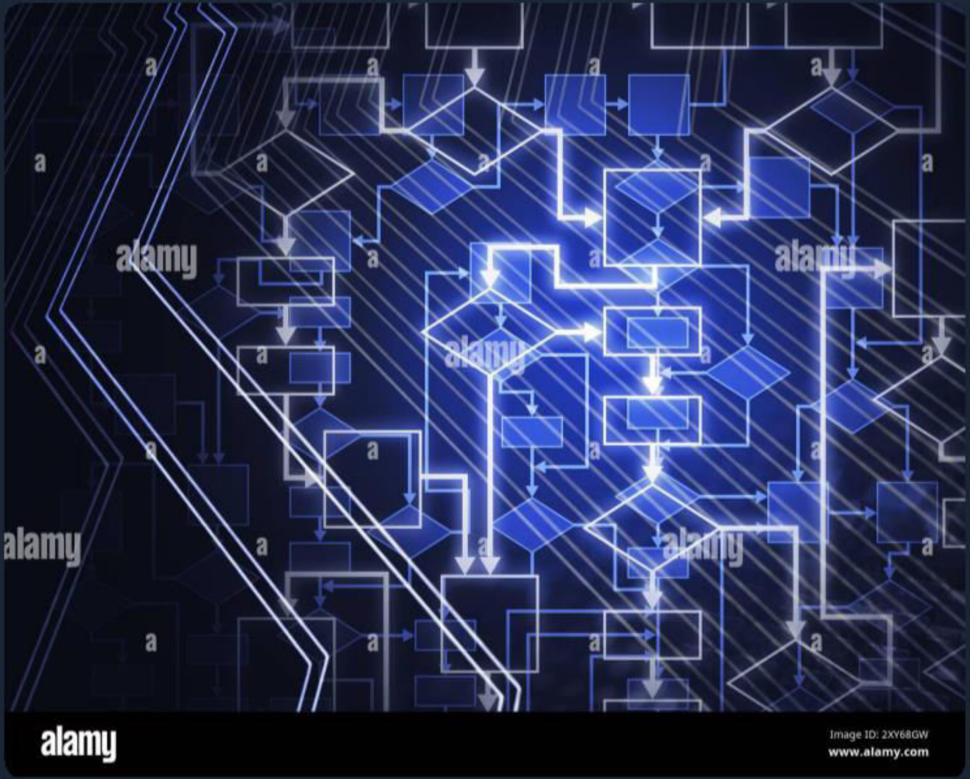
Week 5: Sorting Algorithms

Comparison-Based Sorting:

Dive deep into algorithms that sort data by comparing elements, analyzing QuickSort, MergeSort, and HeapSort for complexities and practical implementation.

Non-Comparison Sorting:

Explore efficient algorithms like Counting Sort, Radix Sort, and Bucket Sort, understanding their specific use cases and linear time potential.



Week 6: Hashing & Heaps

Hashing Fundamentals:

Master Hash Tables for efficient storage, covering hash

Phase 2: Project Application & Industry Immersion (Month 3 Offline)

Applied Learning: Mini-Project Development

Concept Integration: Apply all previously learned DSA concepts to build a tangible, real-world mini-project. This hands-on experience bridges the gap between theoretical knowledge and practical implementation.

Problem-Solving & Collaboration: Engage in collaborative problem-solving, designing robust solutions, and implementing efficient algorithms within a project context, fostering teamwork.

Competitive Edge: Advanced Problem Solving

Intensive Practice: Dive deep into advanced problem-solving techniques and competitive programming. This involves tackling complex algorithmic challenges from platforms like LeetCode and GeeksforGeeks.

Mastering Complexities: Focus on optimizing solutions for time and space complexity, exploring advanced topics like Dynamic Programming, complex Graph Algorithms, and Segment Trees.

Career Readiness: Technical Interview Mastery

Month 3: Advanced Algorithms & Project Implementation

Week 9: Mastering Graph Algorithms

Shortest Path Algorithms: A deep dive into Dijkstra's for non-negative weights and Bellman-Ford for graphs with negative weights, essential for network routing and logistics.

Minimum Spanning Trees: Learn Prim's and Kruskal's algorithms to find the most efficient way to connect a set of points, with applications in network design and clustering.

Week 10: Optimizing with DP & Greedy

Dynamic Programming: Master the art of solving complex problems by breaking them into overlapping subproblems, utilizing memoization and tabulation for optimal solutions.

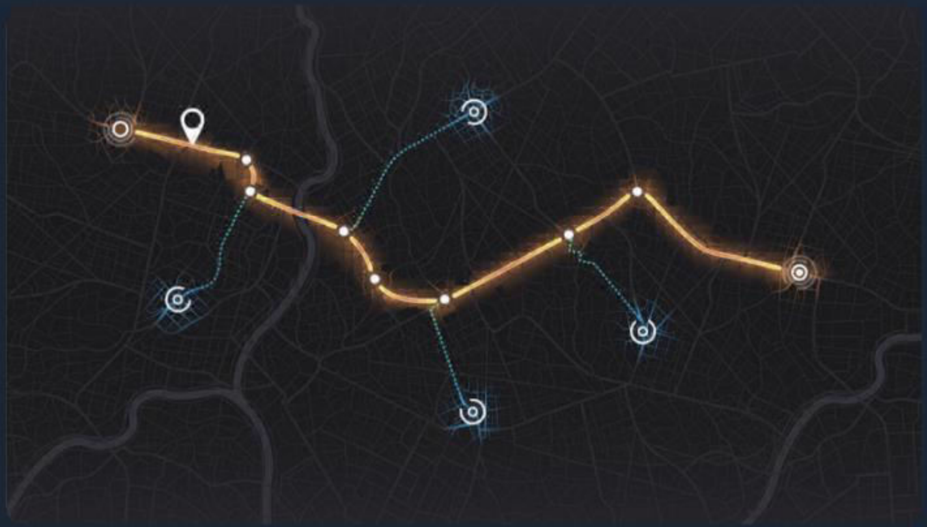
Greedy Algorithms: Understand the principle of making locally optimal choices at each stage, analyzing when this powerful strategy leads to a global optimum.

Week 11: Project & Elite Problem Solving

Advanced Problem Solving: Synthesize all learned DSA concepts to tackle complex challenges from platforms like LeetCode and Codeforces, honing algorithmic intuition.

Mini Project Implementation: Apply knowledge to a real-world scenario, building a tangible portfolio piece that demonstrates design and implementation skills.

The Capstone: Route Optimization System



System Overview: Finding Optimal Paths

- **Core Functionality:** This capstone project challenges participants to engineer a robust route optimization system. Its primary goal is to efficiently compute optimal paths within complex network environments, demonstrating mastery of advanced algorithms and data structures.

- **Practical Application:** The system provides a reliable solution for navigating intricate graphs, such as road networks or logistical supply chains. It leverages sophisticated algorithms to determine the most efficient routes, offering immediate, actionable insights for various real-world scenarios.

Core Pathfinding Algorithms

Graph Representations: The system utilizes efficient graph representations, primarily **Adjacency Lists**, optimized for sparse graphs common in route networks.

Dijkstra's Algorithm: Implemented for scenarios with non-negative edge weights (e.g., travel time). It efficiently computes shortest

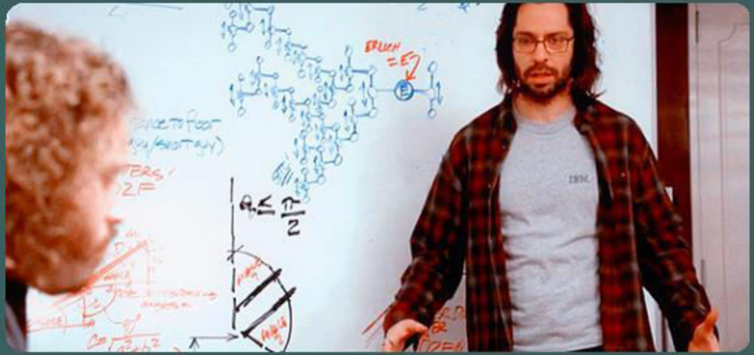
Efficient Data Structures

Priority Queues: A **Binary Heap-based Priority Queue** ensures the next unvisited node with the smallest distance is retrieved in logarithmic time, optimizing Dijkstra's.

Adjacency Lists: Chosen for superior space efficiency in typical road networks, which are often sparse, allowing for faster neighbor

Career Launchpad & Beyond

Week 12: Capstone Showcase & Competitive Programming Simulation



Mini Project Showcase: Present your capstone project, demonstrating applied knowledge of DSA. Showcase design choices, efficient implementations, and practical
problem-solving to peers and mentors.

Competitive Programming Simulation: Engage in an intensive, timed session simulating real-world contests. Hone your problem-solving speed and accuracy under
pressure for top-tier company interviews.

Career Workshops

Resume & Portfolio Building: Craft compelling resumes and impactful portfolios that highlight your DSA projects and technical skills to stand out to recruiters.

LinkedIn Optimization: Master strategies for optimizing your LinkedIn profile and leveraging the platform for job searching and professional branding.

Mock Interviews

Whiteboarding & Coding: Participate in realistic mock interviews focusing on whiteboard and live coding challenges to practice articulating your thought process.

Feedback & Refinement: Receive constructive feedback on your problem-solving, communication, and technical explanations to improve for real SDE interviews.

Networking & Graduation

Industry Connections: Connect with professionals, alumni, and potential employers. Build valuable relationships that can aid in your career launch and future growth.

Graduation Ceremony: Celebrate your significant achievements and the successful completion of the intensive DSA track. A formal recognition of your dedication.

Join the Sapiens AI Journey!

Master DSA for Efficient, Optimized Code

Technical Excellence: Develop a profound understanding of DSA to write clean, high-performing, and scalable code.

Problem-Solving Prowess: Gain analytical skills to identify bottlenecks and approach challenges with robust design patterns.

Solve Complex Problems & Excel in Competitive Programming

Algorithmic Fluency: Train your mind to deconstruct intricate problems and formulate optimal solutions under pressure.

Problem Variety: Build resilience and adaptability by tackling a diverse range of challenges from various domains.

Kickstart Your Career in Tech

Industry Readiness: Build the indispensable foundational