

FULL CURRICULUM • 2026

Competitive Programming

Think faster, solve harder, win contests.

Every module, every lesson — the complete syllabus, end to end.

9

MODULES

126

LESSONS

5

MONTHS

₹18,500

FULL PROGRAMME

What this course covers

1

Level 1 • Beginner

Introduction to Competitive Programming • Standard Template Library (STL) • Bit Manipulation and Modulo Arithmetic

1 month • 3 modules • 42 lessons • ₹4,500

2

Level 2 • Intermediate

Graphs • Segment Trees • Backtracking

2 months • 3 modules • 47 lessons • ₹6,000

3

Level 3 • Advanced

Greedy Algorithms • Dynamic Programming • Advanced Concepts

2 months • 3 modules • 37 lessons • ₹8,000

The whole journey

This is the complete syllabus for Competitive Programming — 9 modules and 126 lessons, in the order you will learn them. The level markers show how far each stage carries you, but the curriculum runs as one continuous path from the first lesson to the last.

1

LEVEL 1

Beginner

Introduction to Competitive Programming · Standard Template Library (STL) · Bit Manipulation and Modulo Arithmetic

1 month · 3 modules · 42 lessons · ₹4,500

1. Introduction to Competitive Programming

13 lessons

- Introduction to the Course
- What is Competitive Programming
- Competitive Programming Competitions
- Websites for Competitive Programming
- Terminologies in Competitive Programming
- Choosing a Programming Language
- Taking Input & Output in Competitive Programming Contests
- Cout vs Printf
- Endl vs Back n
- Fast cin cout Methods _ Conclusion
- Time Complexity in Competitive Programming
- Calculating Time Complexity of Code
- Importance of English in Competitive Programming

2. Standard Template Library (STL)

17 lessons

- Introduction to STL in C++
- Standard Array
- Standard Vector
- Standard Set
- Standard Multiset
- Standard Map
- Standard Multimap
- Standard Pair
- Standard Forward List
- Standard List
- Standard Unordered Set
- Standard Unordered Multiset
- Unordered Map and Unordered Multimap
- Standard Stack
- Standard Queue
- Standard Priority Queue
- Standard Deque



3. Bit Manipulation and Modulo Arithmetic

12 lessons

- Importance of Bit Manipulation Techniques
- Basic Binary Operations
- Bit Operations in C++
- Check ith bit is SET or NOT SET
- Toggle ith bit
- Check odd or even using bits
- Check if Number is Power of 2
- Need for Modulo $10^9 + 7$
- Modulo Arithmetic
- [A3.1] Problem XOR ORED
- [A3.2] Solution - XOR-ORED
- [A3.3] Solution with Code - XOR-ORED

2

LEVEL 2

Intermediate

Graphs · Segment Trees · Backtracking

2 months · 3 modules · 47 lessons · ₹6,000



4. Graphs

17 lessons

- Introduction to Graphs
- Weights in Graphs
- Why Study Graphs
- Edge List
- Adjacency Matrix
- Adjacency List
- DFS Traversal
- Implementing DFS Traversal
- BFS Traversal
- Implementing BFS Traversal
- Minimum Spanning Tree
- Concept of Kruskal's Algorithm
- Concept of Prim's Algorithm
- Concept of Dijkstra Algorithm
- [A4.1] Problem - CHFPLN
- [A4.2] Solution - CHFPLN
- [A4.3] Solution with Code - CHFPLN



5. Segment Trees

19 lessons

- Sum of Updated Range Problem
- Solving Sum of Range Problem with Segment Trees
- Segment Trees - Update Queries
- Segment Trees - Calculate Sum
- Implementing Segment Tree - Part 1
- Implementing Segment Tree - Part 2
- Problem - Range Minimum Query
- Implementing Range Minimum Query
- Implementing Segment Tree - Tricks
- Lazy Propagation in Segment Trees - Part 1
- Lazy Propagation in Segment Trees - Part 2
- Problem - Xenia and Bit Operations
- Solution - Xenia and Bit Operations
- Solution with Code - Xenia and Bit Operations
- Problem - Circular RMQ
- Solution with Code - Circular RMQ - Part 1
- Solution with Code - Circular RMQ - Part 2



6. Backtracking

11 lessons

- Introduction to Backtracking
- N Queens Problem Explained
- N Queens Solved
- Knights Tour Problem Explained
- Knight's Tour Solved
- Rat in Maze Problem Explained
- Rat in Maze Solved
- Problem Richie Rich
- Solving Richie Rich
- Problem False Number
- Solving False Number

3

LEVEL 3

Advanced

Greedy Algorithms · Dynamic Programming · Advanced Concepts

2 months · 3 modules · 37 lessons · ₹8,000



7. Greedy Algorithms

12 lessons

- Introduction to Greedy Algorithms
- Understanding Greedy with Fractional Knapsack Problem
- Solving Fractional Knapsack Problem
- Tasks and Deadlines Problem
- Solving Tasks and Deadlines Problem
- Optimal File Merging Problem
- Solving Optimal File Merge Patterns Problem
- Huffman Coding
- [A7.1] Problem - Dragons
- [A7.2] Solution with Code - Dragons
- [A7.3] Problem - Little Elephant and Bits
- [A7.4] Solution with Code - Little Elephant and Bits



8. Dynamic Programming

14 lessons

- Introduction to Dynamic Programming
- Top down vs Bottom Up
- Fibonacci Problem Top Down
- Implementing Fibonacci with DP
- 2D Grid Traversal Problem
- Implementing 2D Grid Traversal Problem
- Understanding Memoization
- Fibonacci with Tabulation
- Implementing Fibonacci with Tabulation
- 2D Grid Traversal Problem with Tabulation
- Implementing 2D Grid Traversal Problem with Tabulation
- Understanding Tabulation
- [A8.1] Problem - Subset Sum Problem
- [A8.2] Solving - Subset Sum Problem

9. Advanced Concepts

11 lessons

- Binary Lifting (Kth Ancestor of a Tree Node)
- LCA (Lowest Common Ancestor) Problem
- Fenwick Trees - Introduction
- Fenwick Trees - Sum of Range (Range Find Query)
- Fenwick Trees - Update (Point Update Query)
- Problem - Distance between nodes in Trees
- Solution - Distance between nodes in Trees
- Implementing Fenwick Trees
- Implementing Binary Uplifting
- Implementing LCA
- Solution with Code - Distance between nodes in Trees

Certificates & what comes next

You receive a certificate at the end of every level you finish. Join for a single level or the full programme — the choice is yours. Learners who stand out may be invited to work on live client projects as interns; these places are merit-based, limited, and depend on projects being available.