



Competitive Programming

Competitive Programming helps you to become a great programmer and crack coding interviews and competitions. In this program, you will learn how to use advanced data structures to write highly efficient code. You'll also master various problem-solving paradigms that help you to ace any coding competition.

Elementary Data Structures and Algorithms

Build strong foundations in data structures and algorithms by mastering linked lists, stacks, queues, simple sorting techniques and searching algorithms.



Introduction to Algorithms

- What is an Algorithm?
- Asymptotic Notations
- Time complexity
- Space complexity
- Best, Worst, Average case of an Algorithm
- Writing Pseudo-code

Searching & Sorting

- Linear Search
- Binary Search
- Jump Search
- Bubble Sort
- Selection Sort
- Insertion Sort

Divide and Conquer

- Recursion Refresher
- Introduction to Divide and Conquer
- Matrix Multiplication
- Maximum Subarray Problem
- Merge Sort

Arrays & Linked Lists

- Introduction to Data Structures
- Two Pointer method
- Sliding Window
- Prefix Sum
- Singly Linked List
- Double Linked List
- Circular Linked List

Stacks & Queues

- Stack ADT
- Implementing Stacks with Arrays
- Implementing Stacks with Linked Lists
- Queue ADT
- Implementing Queues with Arrays
- Implementing Queues with Linked Lists



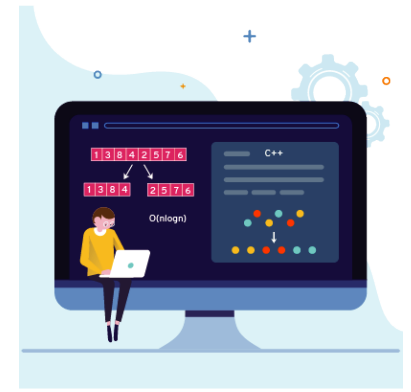
Course Project

LRU Cache

Build an LRU Cache to manage a cache of information stored on the computer.

Data Structures

Master Basic Trees and Graphs along with variants of Queues, Heaps, Hash tables in data structures, and advanced sorting techniques.



Variants of Queues

- Circular Queues
- Deque
- Priority Queues
 - Heaps

Sorting & Selection Algorithms

- Quick Sort
 - Lomuto's partition method
 - Hoare's partition method
- Heap Sort
- Counting Sort
- Bucket Sort
- Radix Sort
- Selection Algorithms
 - Randomised Selection Algorithm
 - Median of Medians

Hash Tables

- Direct Address Tables
- Hash Tables
- Hash Functions
- Chaining
- Open Addressing

Trees

- Introduction to Trees
- Tree Terminologies
- Binary Trees
- Binary Tree Traversals

Binary Search Trees

- Introduction to BST
- Operations on BST
- Self Balancing BST
 - AVL Trees
 - Red-Black Trees

Graphs & Graph Traversals

- Introduction to Graphs
- Graph Terminologies
- Graph Traversals
 - Breadth First Search
 - Depth First Search
- Representation of Graphs
- Topological Sort
 - Kahn's Algorithm
 - Topological Sort with DFS



Course Project

Secure Password
Verification

Build a secure password verification system for logging in to a website.

Problem Solving Paradigms for CP



Master different problem solving paradigms like Dynamic Programming, Greedy Algorithms and Complete Search that will help you to ace any coding competition.

Complete Search

- Iterative Complete Search
- Recursive Backtracking
 - N Queen Problem
 - Solving Sudoku
- Recursion Refresher

Greedy Algorithms

- Huffman Coding
- Activity Selection Problem
- Coin Change
- Load Balancing

Dynamic Programming

- Longest Common Subsequence
- Rod Cutting
- O-1 Knapsack Problem
- Edit Distance
- Matrix-chain Multiplication

Bit Manipulation

- Bitwise Operators
- Bit Masking
- Bit Hacks



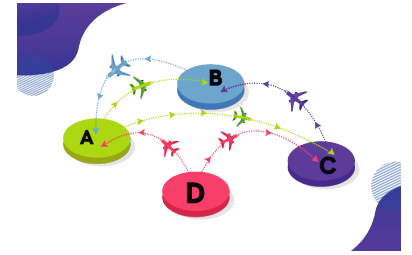
Course Project

Implement DLS
Method used in
Cricket

Implement a DLS method to calculate the target score for the team batting second in a cricket match interrupted by weather.

Advanced Data Structures

Master Advanced Data Structures on Graphs, Trees and Union Find, and Algorithms on Minimum Spanning Trees, Pathfinding, and Flow networks.



Trees - Part A

- Segment Trees
- Fenwick Trees
- Interval Trees
- Range Trees
- Prefix Trees (Trie)
- Suffix Trees

Trees - Part B

- N-ary Tree
- 2-3 Tree
- B – Trees
- Binomial Heaps
- Fibonacci Heaps
- Van Emde Boas Tree

Minimum Spanning Trees

- Krushal's Algorithm
- Prim's Algorithm

Shortest Path Algorithms

- Dijkstra's Algorithm
- Bellman–Ford Algorithm
- Floyd–Warshall Algorithm
- Jhonson's Algorithm

Network Flow

- Flow Networks
- Minimum Cut
 - Soer Wagner Min-Cut Algorithm
 - Gomory-Hu Trees
- Maximum Flow
 - Ford-Fulkerson Method
 - Edmond-Karp's Algorithm
 - Dinic's Algorithm
 - Maximum Weighted Bipartite Matching

Disjoint-set (Union Find)

- Find and Union Operations
- Union by Rank
- Path Compression



Course Project

Quickest Route –
Google Maps

Suggest the quickest route with navigation from Source to Destination based on traffic data similar to Google Maps.

String Processing and Matching



Master various String matching and processing algorithms like KMP, Boyer Moore, Rabin–Karp Substring Pattern, Wildcard pattern matching.

Strings Processing

- Knuth–Morris–Prath's Algorithm
- Z's Algorithm
- Rabin–Karp Algorithm
- Aho Corasick Algorithm

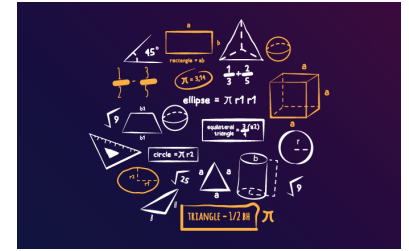
Pattern Matching

- Boyer–Moore Algorithm
- Wildcard Pattern Matching
- Manacher's Algorithm
- String Hashing

Course Project DNA Sequence Matching

Identify the patterns in DNA sequence obtained from different people to identify relationships among them.

Mathematics for Competitive Programming



Master concepts of Number Theory, Game Theory, Geometry, and Computational Geometry needed for Competitive Programming.

Mathematical Foundations

- Number Systems
- Matrix Operations and Properties
- Set Theory
- Counting and Probability
- Series Summation and Properties: AP, GP, HP

Number Theory

- Primality Testing
 - Sieve of Eratosthenes
 - Segmented Sieve
 - Fermat Method
- GCD and LCM
 - Basic and Extended Euclidean Algorithms
- Euler's Totient (Phi) Function
- Fibonacci Numbers
- Factorials
- Modulo Arithmetic
 - Modular Exponentiation
 - Modular Multiplicative Inverse
 - Chinese Remainder Theorem
 - Lucas Theorem

Game Theory

- MiniMax Algorithm
- Sprague-Grundy Theorem
- Nim Game

Computational Geometry

- Geometry Basics
- Closest Pair of Points
- Intersection Problems
- Convex Hull – Graham's Scan



Course Project

Chess Bot

Build a chess bot to play and compete with a user.

Selected Topics in CP

Master advanced topics like Approximation algorithms, Linear Programming, FFT which are useful in cracking top-level international coding competitions. Also, learn how to use libraries offered in C++, Java & Python for CP.



Advanced Topics

- Linear Programming
- Polynomials and FFT
- NP-completeness
- Approximation Algorithms

Useful Libraries for Competitive Programming

- C++ STL Library
- Java java.util Package
- Python Libraries

Contact us

 support@nxtwave.tech

 www.ccbp.in