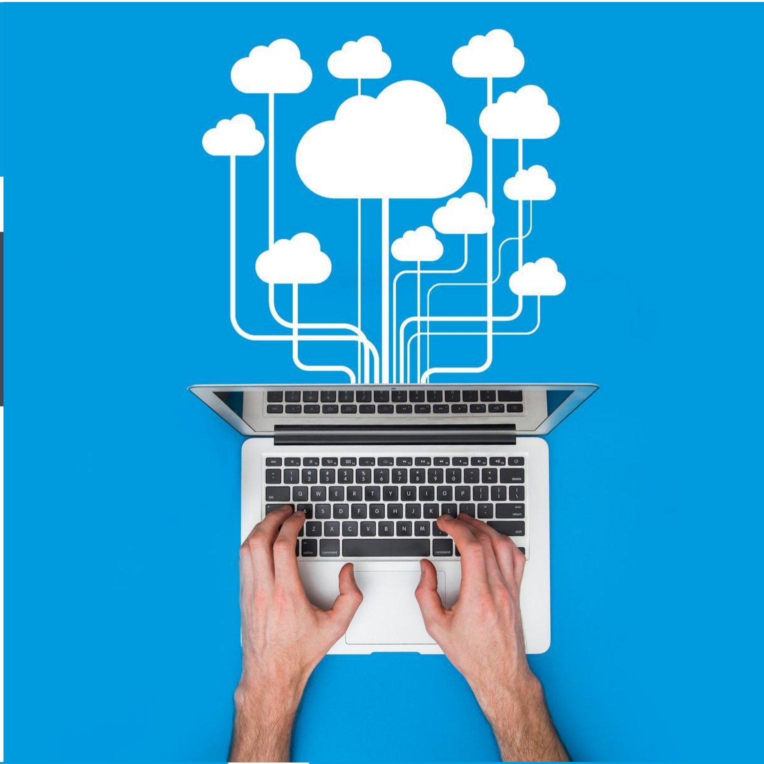
ACA'24

MASTERING ALGORITHMS

Mentors

1)Venkatesh 2)Kumar Shivam

3)Nipun Nohria 4)Yash Pratap Singh



Motivation(WHY ?)

"Algorithms are the building blocks of problem-solving"

- Algorithmic problem solving is important in many areas of modern life, from hard and soft sciences to art and entertainment, especially in Computer Science.
- Enhances critical thinking and Algorithmic Problem solving Skills.
- Helpful in Placements - Being experienced in competitive programming , gives an edge during coding rounds of job interviews.

Learning Outcomes

The project aims to:

- ➢ Introduce efficient Algorithmic Problem Solving methods.
- ➢ Enhance problem-solving skills through competitive programming challenges and math puzzles.
- ➢ Explore the applications of number theory, graph theory, and combinatorics to solve complex problems efficiently.
- ➢ Delve into frequently used algorithms and techniques in competitive programming and algorithmic puzzles for a deeper understanding.

Table Of Content

Topics

Week Distribution

Methodology

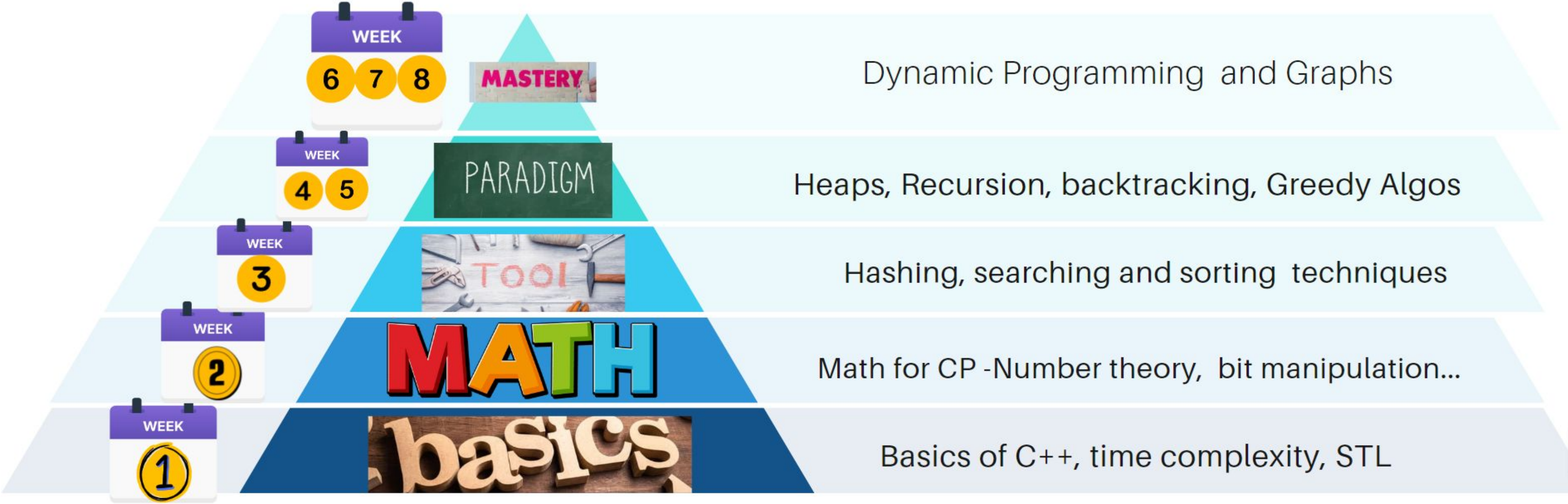
Evaluation Criteria

Our Team

Topics

- Week 1: Basics of C++ , time complexity, STL
- Week 2: Modular Arithmetic, Sieve of Eratosthenes , Theorems in number theory, bit manipulation, Constructions
- Week 3: Hashing, searching and sorting
- Week 4 and Week 5 : Heaps, Recursion, Greedy Algos
- Week 6 and Week 7: Dynamic Programming
- Week 7 and Week 8: Graph Algorithms

Topics and Timeline



Dynamic Programming and Graphs

Heaps, Recursion, backtracking, Greedy Algos

Hashing, searching and sorting techniques

Math for CP -Number theory, bit manipulation...

Basics of C++, time complexity, STL

Week 1

- Basic concepts of C++
- Time complexity in programming
- Introduction to vectors
- Introduction to Stacks and problems related to stacks .
- Introductions to Queues and related problems.

Week 2

- Number theory
- Primes and Modular arithmetic
- Bit manipulations
- Basic Combinatorics
- Game theory
- Probability

Week 3

- Brute force, Divide and conquer
- Sorting and Searching algorithms
- Hashing
- Sliding window
- Two pointer method

Week 4 and Week 5

- Introduction to Heaps
- Advanced concepts in heaps
- Recursion
- BackTracking
- Greedy Algorithms

Week 6

- Introduction to dynamic programming
- Recursion problems using dynamic programming
- Solving greedy problems using dynamic programming
- Travelling salesman using dp
- Longest common subsequence problem
- Knapsack Problem

Week 7 and Week 8

- Basics of graphs
- Adjacency lists
- BFS and DFS Traversal
- Theoretical discussion as well as code implementation
- Dijkstra's Algorithm
- Spanning Trees

Methodology

- Step 1: Theoretical description of the concept using examples
- Step 2: C++ code implementation of the problem
- Step 3: Reference material for students to refer to for additional clarity in the topic.
- Step 4: Topicwise codeforces/leetcode/hackerRank/geeksforgeeks weekly problems with deadline.
- Step 5: Doubt sessions and active whatsapp participation

An Algorithmic Puzzle every alternate day, typically have the flavor of Project Euler

Evaluation Criteria

1) Weekly assignment/practice problems
2) 2 major programming assignments – one before midterm and one before end term
3) Mid term and End term Report

Our Team

Yash Pratap Singh

Y22 CSE

7007519537

Nipun Nohria

Y22 CSE

8556827697

Kumar Shivam

Y22 CSE

6299712990

Venkatesh

Y22 CSE

9014764100

THANK YOU!