Course Title: Data Structure & Algorithms – Level 1

Course Overview:

This comprehensive training course is designed for high school students aiming to excel in the USA Computing Olympiad (USACO) contests. The course covers fundamental and advanced data structures and algorithms, with a strong emphasis on practical problem-solving techniques relevant to competitive programming. Students will learn to implement efficient algorithms in Python, tackle complex programming challenges, and develop strategies to excel in USACO contests. Each week includes theoretical concepts, hands-on coding exercises, and practice problems similar to those found in USACO competitions.

Course Objectives:

- Develop a strong foundation in Python programming.
- Understand and implement key data structures and algorithms.
- Solve complex problems using efficient algorithms.
- Prepare for USACO contests with targeted practice and strategies.
- Improve problem-solving skills and coding proficiency under contest conditions.

Course Duration: 33 weeks (1 hours per session, 1 session per week). Each week there is one session for lecture and one session for workshop.

Prerequisites:

- Familiar with programming fundamentals and data structures.
- Windows 10/11 laptop or desktop.

Course Syllabus:

Week 1-3: Basics knowledge

- Overview of Algorithms and Data Structures
- Importance of Data Structures in Problem Solving
- Algorithmic Complexity (Big-O, Big-Theta, Big-Omega)
- Recursion and Recurrence Relations
- Mathematical Induction

- Basic Probability for Algorithms

Lab/Assignment: Analyze and compare the time complexity of simple algorithms.

Week 4-6: Sorting Algorithms

- Bubble Sort
- Selection Sort
- Insertion Sort
- Quick Sort
- Merge Sort

Lab/Assignment: Implement and analyze basic sorting algorithms.

Week 7-9: Arrays and Strings

- Introduction to Arrays and Strings
- Common Operations (Traversal, Insertion, Deletion)
- List, Tuple, Set and Dictionary operations
- String manipulations
- String Matching Algorithms (Naive, Rabin-Karp)

Lab/Assignment: Implement common array operations and solve string manipulation problems.

Week 10-12: Searching Algorithms

- Overview of searching algorithms
- Linear Search
- Binary Search
- Search Optimization Techniques
- Complete Search

Lab/Assignment: Implement linear, binary search algorithms. Practice complete search algorithm using USACO problems.

Week 13-15: UASCO Contest Preparation

- Problem Solving Paradigms
- Common Algorithmic Strategies
- Mock Contests

- Analysis of UASCO Problems

Lab/Assignment: Participate in mock contests and analyze solutions.

Week 16-18: Greedy Algorithms

- Overview of greedy algorithmic strategy
- Greedy choice property and optimal substructure
- Examples of greedy algorithms: minimum spanning tree, Huffman coding, interval scheduling
- Analysis of greedy algorithms

Lab/Assignment: Solve problems using greedy algorithms.

Week 19-21: Stacks and Queues

- Introduction to Stacks
- Stack Operations (Push, Pop, Peek)
- Applications of Stacks (Balanced Parentheses, Postfix Evaluation)
- Introduction to Queues (Simple, Circular, Deque)
- Queue Operations (Enqueue, Dequeue)
- Applications of Queues (Breadth-First Search)

Lab/Assignment: Implement stack operations and solve problems using stacks and queues.

Week 22-24: USACO Training – Chapter 1 – Part 1

- Review the complete search problems in section 1.3
- Review the greedy problems in section 1.4

Lab/Assignment: Solve the problems in section 1.3 and 1.4 for Chapter 1 of USACO Training.

Week 25-27: USACO Training – Chapter 1 – Part 2

- Review the More Searching Techniques in Section 1.5
- Review the Binary Number problems in Section 1.6

Lab/Assignment: Solve the problems in Section 1.5 and 1.6.

Week 28-30: USACO Contest Lab - 1

- Review bronze problems from previous years of USACO contest

Lab/Assignment: Implement stack operations and solve problems using stacks and queues.

Week 31-33: USACO Contest Lab - 2

- Review all bronze problems for Year 2024-2025 USACO Contest

Lab/Assignment: solve UACO bronze level problems using the algorithms we learned in this course.