

Applied Data Structures and Algorithms using Python3

Jimut Bahan Pal

Semester - III, 2024

E-mail: pal.jimut@iitb.ac.in

Website: jimut123.github.io/courses/adsap2024

Credit Hours: 3

Course Description

This course, "Applied Data Structures and Algorithms using Python3," is designed to provide students with a comprehensive understanding of fundamental and advanced data structures and algorithms utilizing Python3 as the programming language.

Python3 is preferred for this course due to several key reasons:

- **Brevity and Readability:** Python's syntax is concise and clear, allowing students to express complex ideas in fewer lines of code than languages like C++ or Java. This helps quickly translate thoughts into working code.
- **Ease of Learning and Use:** Python is known for its gentle learning curve, making it accessible to students who might be new to data structures and algorithms.
- **Wide Usage in Industry:** Python is widely used in various domains, including deep learning research, data science, web development, and automation. Learning DSA in Python prepares students for a broader range of job opportunities.
- **Efficient Coding:** The extensive standard library and powerful data structures provided by Python, such as lists and dictionaries, enable efficient coding and problem-solving.
- **Interview Preparation:** Leading tech companies often allow coding interviews in Python, and its versatility makes it an excellent choice for tackling various algorithmic problems.
- **By focusing on Python3,** this course aims to hone students' skills in a language that is practical for academic purposes and highly valuable in the professional world.

Required Materials

- Course materials, i.e., codes, resources will be made available via Google Spaces.

- Pointers to additional resources week-wise will be made via course website or via Google Spaces.

Prerequisites

Only a willingness to learn and code.

Course Objectives

- Enhance understanding of key data structures and algorithms using Python3.
- Prepare students for technical interviews for SDE roles.
- Develop problem-solving skills through extensive coding practice.
- Able to see problems more clearly and apply intuition to newer ones.

Grading Policy

(Tentative, total - 100%) Relative grading will be used.

- 20%: Homeworks.
- 35%: Mid-Semester examination.
- 45%: End-semester examination.

Tentative Topics

The following topics provide a general overview of the content to be covered in this course. This list is tentative and not exhaustive, allowing for flexibility based on course progression and student interest.

- Arrays and Hashing - Understanding arrays and their operations, implementing common algorithms using arrays, hashing techniques, and hash tables. Focus on practical applications in problem-solving.
- Two Pointers - Introduction to the two-pointer technique, applying two-pointers for various problem types such as searching pairs, triplets, and subarrays. Utilizing the fast and slow pointer technique for cycle detection.
- Sliding Window - Understanding the sliding window technique and applying it to optimize solutions. Handling both fixed-size and dynamic-size window problems.
- Binary Search - Introduction to binary search and its variations, applying binary search to different data structures. Emphasis on solving both simple and complex binary search problems.

- Linked List - Understanding linked list structures and operations, implementing singly and doubly linked lists. Solving problems involving list manipulation and cycle detection.
- Trees - Introduction to tree data structures, understanding binary trees and binary search trees. Tree traversal techniques (in-order, pre-order, post-order) and advanced tree algorithms and applications.
- Heap/Priority Queue - Understanding heaps and priority queues, implementing heap operations. Using heaps for various problem-solving scenarios.
- Graphs - Introduction to graph theory, graph representation (adjacency list, adjacency matrix), graph traversal techniques (BFS, DFS). Emphasis on solving both basic and advanced graph problems.

This structure covers the essential topics with an emphasis on understanding key concepts. The practical application through platforms like LeetCode will be integral to developing problem-solving skills and preparing students for technical interviews and real-world scenarios.