Course Outline

Course Contents

Week 1: Introduction to Data Structures and Python Basics

An overview of the course objectives, grading, and structure. Introduction to fundamental programming concepts in Python, including variables, loops, functions, and data types necessary for implementing data structures.

Week 2: Algorithm Complexity, Searching, and Sorting

An introduction to the concept of algorithm complexity and Big O notation. Overview of basic searching (linear and binary search) and sorting algorithms (selection, insertion, and bubble sort) to understand their performance and applications.

Week 3: Arrays, Sets, and Maps

Introduction to arrays as a foundational data structure. Understanding the operations on arrays, their limitations, and applications. Exploration of sets and maps (dictionaries in Python), their properties, and practical uses.

Week 4: Stacks

Introduction to the stack data structure, its operations (push, pop, peek), and common applications. Discuss real-world scenarios where stacks are useful, such as function calls, undo operations, and parsing expressions.

Week 5: Queues

Discussion on the queue data structure and its variations (FIFO queues, circular queues). Implementation of queues in Python and understanding their applications in areas like scheduling and buffering.

Week 6: Linked Lists

Understanding singly linked lists and their differences from arrays. Implementation of linked lists, including operations like insertion, deletion, and traversal. Discuss scenarios where linked lists are more efficient than arrays.

Week 7: Doubly Linked Lists and Applications

Extension of linked lists to doubly linked lists, which allow traversal in both directions. Discuss the additional operations possible with doubly linked lists and their applications, such as navigation in browser history and playlist management.

Week 8: Binary Trees

Introduction to tree data structures, focusing on binary trees. Understanding the concepts of nodes, edges, height, and depth. Discuss tree traversal algorithms (in-order, pre-order, and post-order) and their applications.

Week 9: Binary Trees (Continued)

Continuation of binary trees with a focus on more advanced topics, such as binary search trees (BSTs), their properties, and operations (insertion, deletion, search). Discussion on balancing trees and the significance of balanced trees in efficient data storage.

Week 10: Graphs

Introduction to graphs as a generalization of trees. Understanding graph representations (adjacency list, adjacency matrix), types (directed, undirected, weighted), and basic traversal algorithms (BFS, DFS). Applications of graphs in real-world scenarios like social networks and route planning.

Week 11: Advanced Topics

Exploring advanced data structures and algorithms such as heaps, hash tables, and priority queues. Understanding the significance and applications of these structures in various computer science problems.

Week 12: Recap

A comprehensive review of all topics covered throughout the course. Focus on reinforcing core concepts, discussing real-world applications, and preparing for the final assessment. Opportunity for students to clarify any questions and participate in a Q&A session.