**Data Structures and Algorithms**

**Course Overview**

This course provides an introduction to the design and analysis of algorithms for students. Topics covered include asymptotic complexity and algorithm design techniques such as incremental, divide-and-conquer, dynamic programming, greedy algorithms, and backtracking. Additionally, fundamental data structures such as stack, queue, heap, graph, tree and hashtables are covered, as well as their role in the design of efficient algorithms. Although the Python programming language is extensively used in examples, this class is not about Python. Rather, we will emphasize computational solutions to problems and the role of efficient algorithms as a tool for exploring and processing large data sets.

**Syllabus**

|  |  |
| --- | --- |
| Week | Topic |
| 1 | Introduction to Algorithms, Paradigms and Asymptotics |
| 2 | Data Structures Overview (Array, Hashtable, Heap) |
| 3 | Data Structures Overview (Stack and Queue) |
| 4 | Data Structures Overview (Linked List, Binary Tree, N-ary Tree) |
| 5 | Recursion, Divide and Conquer |
| 6 | Problem Solving Techniques: Binary Search, Two Pointers, Sliding Window |
| 7 | Graph Data Structure and Search Algorithms |
| 8 | Optimization: Intro to Dynamic Programming |
| 9 | Optimization: Intro to Backtracking and Memoization |
| 10 | Optimization: Intro to Greedy Algorithms |