

Design and analysis of algorithms

2026 Fall

Title: Design and analysis of algorithms

Instructor: Banerjee Amitayu, Ph.D.

Aim of the course:

This is an introductory course with an emphasis on teaching techniques for the design and analysis of efficient algorithms providing a foundational base for coding in Python.

Content of the course:

1. **Basic linear and non linear data structures:** Arrays, linked lists, trees, graphs, Stacks, queues, hash tables etc.
2. **Asymptotic Notations:** Big Oh, Big Omega, Big Theta notations.
3. **Sorting Algorithms:** Bubble, selection, insertion, merge, and quick sort.
4. **Greedy Algorithms:** Definition, solution of coin change problem, Fractional Knapsack problem, Huffman encoding.
5. **Divide and Conquer Algorithms:** Definition, examples
6. **Dynamic programming Algorithms:** Definition, examples like solution to 0/1 Knapsack problem, and coin change problem.
7. **Heap data structure:** Definition, types of heaps, operations (heapify, insertion, deletion), applications (priority queues, heapsort algorithms)
8. **Graph traversal algorithms:** BFS (Breadth-First-Search) and DFS (Depth-First-Search), and their applications.
9. **Minimum spanning tree greedy algorithms:** Kruskal's and Prim's algorithms.
10. **Shortest Path algorithms:** Dijkstra algorithm (Greedy) and Floyd-Warshall algorithm (Dynamic programming).
11. **Network Flow:** Ford-Fulkerson Algorithm
12. **An overview of P, NP-complete, and NP-Hard problems.**

We will learn how to implement sorting algorithms, as well as BFS, DFS, Kruskal, Prim, Dijkstra, and Ford-Fulkerson algorithms in Python.

Grading criteria:

Oral exam and solving assignments.