Jahangirnagar University

Institute of Information Technology (IIT)

ICT 2201: Algorithm Analysis and Design

Course Outline

Course Teacher: Prof. Dr. Mohammad Abu Yousuf
Email: yousuf@juniv.edu, usuf672@yahoo.com
Class hours: 10:00-11:30(Monday, Wednesday) (Theory), 2:00-4:30(Monday)(Lab)
Credit: 3 credit hours.

Course Objectives

- Analyze the asymptotic performance of algorithms.
- Write rigorous correctness proofs for algorithms.
- Demonstrate a familiarity with major algorithms and data structures.
- Apply important algorithmic design paradigms and methods of analysis.
- Synthesize efficient algorithms in common engineering design situations.

Textbook:

1. Algorithms
   By Thomas H. Cormen, Leiserson, Rivest, and Stein

Reference Book:

1. Fundamentals of Computer Algorithms
   By Horowitz and Sahani
2. Data Structures and Algorithm Analysis in C++
   By Mark Allen Weiss
3. Algorithm Design
   By J. Kleinberg and E. Tardos

Course Outcomes:

On completion of the course students will be able to:

- Argue the correctness of algorithms using inductive proofs and invariants.
- Analyze worst-case running times of algorithms using asymptotic analysis.
- Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and-conquer algorithms. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.
- Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic-programming algorithms, and analyze them.
- Describe the greedy paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize greedy algorithms, and analyze them.

- Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate. Synthesize new graph algorithms and algorithms that employ graph computations as key components, and analyze them.

Marks distribution:

1. Home works/Presentation	10%
2. Class Tests	20%
3. Class Attendance	10%
4. Final	60%

Lecture wise course outline:

Lecture	Topic	Discussion points
1	Orientation with students, general discussion about algorithm	1. Introduction with the students 2. Basis of algorithm analysis 3. Defining algorithm
2	Overview of algorithm	1. Complexity 2. Time-space 3. Tradeoff
3	Sorting algorithm 1: Insertion sort	1. insertion sort algorithm 2. Analysis of insertion sort 3. Understanding the performance of insertion sort
4	Designing an algorithm : Divide and Conquer Method-1	1. Marge sort 2. Complexity of merge sort 3. Analysis the performance of merge sort
5	Designing an algorithm : Divide and Conquer Method-2	1. Quick sort 2. Complexity of quick sort 3. Analysis the performance of quick sort
6	Sorting algorithm 2: Heap sort	1. Properties of heap 2. Build a heap 3.Design a heap sort algorithm and analysis the performance of heap sort
7	Sorting algorithm 3:	1. Count sort 2. Radix sort 3. Bucket sort
8	Binary search tree 1	1. Understanding a BST 2. Find the minimum and maximum value of a BST 3. Find successor and predecessor of a BST
9	Binary search tree 2	1. Insert a data in a BST 2. Delete a data from a BST 3. Calculate the time complexity of various operation on a BST
10	AVL Tree, B-tree , Red-Black Trees	1. Insert into and delete from AVL tree 2. Searching, insertion and deletion in a B-tree 3. Insertion and deletion in a red-black tree
11	Designing an algorithm : Dynamic Programming (DP)- 1	1. What is DP? 2. Properties of DP 3. Matrix chain multiplication using DP
12	Designing an algorithm :	1. Computing a binomial coefficient by DP

	Dynamic Programming (DP)-2	2. Find longest common subsequence (LCS) 3. Performance of LCS
13	Designing an algorithm : Greedy approach-1	1. What is optimization problem? 2. Activity selection problem using greedy approach 3. Elements of greedy approach
14	Designing an algorithm : Greedy approach-2	1. 0/1 and fractional knapsack problem 2. Difference between DP and greedy approach. 3. Huffman coding
15	Designing an algorithm : Greedy approach-3	1. Understanding minimum spanning tree (MST) 2. Prim's algorithm to find MST 3. Kruskal's algorithm to find MST
16	Designing an algorithm : Back tracking algorithm-1	1. Understanding N-Queen problem 2. Solve N-Queen problem using back tracking method 3. Subset sum problem
17	Designing an algorithm : Back tracking algorithm-2	1. Hamiltonian cycle 2. Graph coloring problem 3. Graph coloring problem using intelligent back tracking
18	Graph algorithm and its application-1	1.Sequential representation of graphs 2.Warshall's algorithm 3.Linked representation of graph
19	Graph algorithm and its application-2	1. Operation on graphs: Searching in a graphs 2. Operation on graphs: Inserting in a graph, Deleting from a graph 3. Traversing a graph
20	Graph algorithm and its application-3	1. Traverse in a graph 2. Breadth-first search 3. Depth first search
21	Graph algorithm and its application-4	1. Single source shortest path 2. Dijkastra's algorithm to find SSSP 3. Bellman-Ford algorithm to find SSSP
22	Graph algorithm and its application-5	1. All pairs shortest path 2. Floyd-Warshall algorithm to find all pairs shortest path 3. Transitive closure of shortest path
23	Hashing	1. Hash tables 2. Hash functions, Collision 3. Open addressing and Perfect Hashing
24	P, NP problem	1. Polynomial time algorithm 2. Non polynomial time algorithm 3. Decision vs Optimization problem
25	NP Complete, NP Hard	1. Understanding NP Complete and NP hard problem. 2. NP complete problem algorithm 3. NP hard problem algorithm
26	Paper Presentation: 1	Papers will be selected two weeks before presentation
27	Paper Presentation: 2	Papers will be selected two weeks before presentation
28	Review class	-------

(Prof. Dr. Mohammad Abu Yousuf)
IIT, JU