Syllabus

for 26:198:685 Introduction to Algorithms and Data Structures

Fall 2018, Monday 1-4PM 1WP room 518

Last updated on 08/30/18 at 03:27:30 PM

Texts and References**:**

(Required) T. Corman, C. Leiserson, R. Rivest, and C. Stein *"Introduction to Algorithms", Third edition* MIT Press 2009.

This book is our main text.

Required work

There will be 6 or 7 homework sets during the semester. They will constitute 50% of your final grade. Also, **you will not pass this course if you fail to hand in two homework assignments or more regardless of your total score!**

We will have just a final exam. The final constitutes 50% of your grade

Topics

Our main purpose in this course is to cover a broad spectrum of topics on the theory of algorithms and data structures. Algorithms are the foundation of computer science and information technology, and every expert and researcher in IT has to have a working knowledge of them. In this course, we will cover the most fundamental algorithms and the essential data structures. We will also learn the basic mathematical tools necessary to analyze these algorithms.

Below is a list of topics to be covered in the course. A star (*) in front of the topic means it will be covered if there is enough time

Topic 1: Basics and definitions. (1)

The notion of an algorithm and computational resources used to perform it (primarily time and space, but also amount of parallel computational resources will be considered.

Simple and familiar algorithms such as sums and products of numbers, sums and products of polynomials, and sums and products of matrices as case studies

The notion of asymptotic analysis, and the concept of Big O

Use of counting techniques, recursion and summation methods to estimate the performance of algorithms. Case study: computing minimum and maximum elements, simple sorting methods such as insertion and selection sorting

Reading: Chapters 1,2,3

Topic 2: Divide and Conquer paradigm and recursion, randomized algorithms, and average case analysis (1)

Recurrence relations and recursive algorithms: Recursive methods of multiplying polynomials, Strassen's matrix multiplication, Mergesort algorithm, computing performance of algorithms through recurrence relations

The notion of average case analysis of algorithms, the notion of randomized algorithms, and the difference between the two paradigms. Case study the Quicksort algorithm.

Computing medians and other order statistics

Readings: Chapters 4,5,7,9

Topic 3: Other topics involving numbers, polynomials, and matrices (2)

*Matrix computations, Solving systems of equations, LUP factorization, determinants, permanents (***Read chapter 27)**

The Fast Fourier Transform, its inverse, and its analysis. Applications to polynomial computations. (**Read chapter 30)**

Topic 4: Number theoretic algorithms, Applications to random number generation, cryptography, RSA (2)

Euclid’s algorithm for gcd (greatest common divisor) of two number

modular arithmetic, and the notion of a group

Random number generation

Testing and generating prime numbers efficiently

Public Key cryptography, trapdoor functions, cryptographic hash functions

RSA public key system, and the assumption of difficulty of factoring integers

Reading chapter 31

Topic 5: Information storage and retrieval, Searching, and supporting data structures (3)

Elementary data structures such as arrays, linked list, stacks, and queues.

Hash tables, hash functions, and various hashing and addressing techniques and their (probabilistic) analysis

Binary trees, relation to height and number of nodes, priority queues, heaps, Heapsort

Representation of collections by balanced binary search trees, building, insertion, search, delete, merge. Implementation by red-black trees. Analysis of basic operations

k-d trees, and range searching with applications to machine learning

*B-trees and external search (searching data distributed across many fields.

Reading: Chapters 10, 11, 12, 13, 18 and notes

Topic 5A: Introduction to Block chain and cryptocurrencies (1)

The notion and properties of blockchain data structure

Application in implementation cryptocurrency, bitcoin

*An introduction to distributed computing.

Reading: Chapters 10, 11, 12, 13, 18 and notes

Topic 6: Algorithms and data structures for networks and graphs (2.5)

Definition of directed and undirected graphs, adjacency, and incidence matrix representation, depth-first and breadth-first search, topological sorting, strong and weak connectedness, and components,

The “greedy paradigm” minimum spanning trees and Prim's and Kruskal's algorithms.

Shortest paths and the dynamic programming paradigm. Dijkstra's shortest path algorithm, Floyd-Warshall algorithm, the idea of dynamic programming, application of dynamic programming to other problems (knapsack, TSP, matrix chain product)

*Maximum flow problem, Max-flow-min-cut theorem, Ford and Fulkerson Augmenting path algorithm, Complexity analysis

Reading: Chapters 22, 23,24,25, 26*

Topic 7: NP-completeness and Computational intractability (1.5)

The precise definition of an algorithm: boolean algebra, Circuits

Decision problems, optimization problems, and functional problems: Why focusing on decision algorithms is useful

The first NP-complete problem: Satisfiability problem (SAT), and refinements: 3-SAT

NP-completeness of CLIQUE, HAMILTONIAN-CYLCE, TSP, COLORING, SUBSET-SUM, KNAPSACK, INTEGER PROGRAMMING

Reading: Chapter 34