CS345: Algorithms II

Background Assumed Knowledge of big-O, big-Omega, small-o; programming paradigms: divide-and-conquer, dynamic-programming, greedy-approach; elementary data-structures: array, linked-list, stack, queue, trees, binary-search-trees, balanced search trees, hash-tables; correctness proofs and complexity analysis of simple algorithms such as merge-sort, quick-sort etc.

Course Content This course will focus on fundamental algorithms from several domains including graph-theory, computational-geometry, strings, flow-networks, number-theory, polynomials and matrices.

Conduct of the Course This course will be conducted in flipped manner. A comprehensive set of notes will be provided which you should read before coming to the class. In the class we will discuss the doubts and related questions.

Topics

1. Graph Definitions and Elementary Algorithms: Shortest path by BFS, shortest path in edge-weighted case (Dijkasra's), depth-first search and computation of strongly connected components, emphasis on correctness proof of the algorithm and time/space analysis, example of amortized analysis. (4 lectures)
2. Matroids: Introduction to greedy paradigm, algorithm to compute a maximum weight maximal independent set. Application to MST. (3 lecture)
3. Graph Matching: Algorithm to compute maximum matching. characterization of maximum matching by augmenting paths, Edmond's Blossom algorithm to compute augmenting path. (3 lectures)
4. Flow-Networks: Maxflow-mincut theorem, Ford-Fulkerson Method to compute maximum flow, Edmond-Karp maximum-flow algorithm. (3 lectures)
5. Matrix Computations: Strassen's algorithm and introduction to divide and conquer paradigm, inverse of a triangular matrix, relation between the time complexities of basic matrix operations, LUP-decomposition. (3 lectures)
6. Shortest Path in Graphs: Floyd-Warshall algorithm and introduction to dynamic programming paradigm. More examples of dynamic programming. (3 lecture)
7. String Matching: Knuth-Morris-Pratt algorithm, Rabin-Karp algorithm, testing the membership of a regular language. (3 lectures)
8. Basic Number algorithms: Reciprocal and square algorithms to show that the time complexities of multiplication, squaring, reciprocal, and division are same. Extension to polynomials. (3 lectures)
9. Modulo Representation of integers/polynomials: Chinese Remainder Theorem, Conversion between base-representation and modulo-representation. Extension to polynomials. Application: Interpolation problem. (3 lectures)
10. Discrete Fourier Transform (DFT): In complex field, DFT in modulo ring. Fast Fourier Transform algorithm. Schonhage-Strassen Integer Multiplication algorithm. (4 lectures)
11. Linear Programming: Geometry of the feasibility region and Simplex algorithm. (2 lectures)
12. NP-completeness: Examples, proof of NP-hardness and NP-completeness. (2 lectures)

Text and References

1. My comprehensive notes will form the main text for the course.

In addition you may refer to

2. "Introduction to Algorithms" by Cormen, Leiserson, Rivest, Stein.
3. "The Design and Analysis of Computer Algorithms" by Aho, Hopcroft, Ullman.

4. "Algorithm Design" by Kleinberg and Tardos.

Tests Mid semester examination and four to five homeworks/quizzes. Quizzes will be held outside the lecture hours.

Grading The weightage of the mid-semester exam will be $30 - 35\%$, that of end semester exam will be $45 - 50\%$ and the weightage of homework/quizzes will be $15 - 25\%$.

Course Instructor Shashank K Mehta (skmehta@), 503 RM building, Computer Science Department.

Teaching Assistants Will be informed later.