

CS-510 – Advanced Analysis and Design of Algorithms

Instructor: Dr. Mehvish Irfan Poshni, **Semester:** Spring 2020, **Campus:** Quaid-e-Azam Campus

Course Description:

This is a graduate level core course for M. Phil. in Computer Science and is designed as a follow up course to the undergraduate course on analysis and design of algorithms. This course will primarily cover advanced techniques for design and analysis of algorithms. Some classical algorithms that are considered historically important will also be discussed in this context. Emphasis is laid on a rigorous treatment of program correctness as well as complexity of algorithms.

Number of lectures: 32 lectures (with 8 ramadan lectures expected to span 50 minutes each)

Goals and Objectives:

On the completion of this course students would be able to meet following objectives:

1. Demonstrate understanding of design and analysis techniques as well as specific algorithms covered in class.
2. Demonstrate the ability to apply the concepts learnt by being able to solve homework and exam problems.

Text:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", MIT Press, Third edition, 2009
2. Handouts

Reference Material:

- I. Jon Kleinberg, and Eva Tardos, *Algorithm Design*, Addison-Wesley Longman Publishing Co., 2005.
- II. Vijay Vazirani, *Approximation Algorithms*, Springer International Edition, 2003.
- III. Rajeev Motwani and Prabhakar Raghavan, *Randomized Algorithms*, Cambridge University Press, NY, 1995.
- IV. Sanjoy Dasgupta, Christos Papadimitriou, and Umesh Vazirani, *Algorithms*, e-book, 2006

Scheme of Study:

Week 01

1. Introduction, machine models, loop Invariants and proofs of correctness of algorithms (2.1, 2.2).
2. Review of asymptotic notations, Analysis (3.1, 3.2, 4.3)



Week 02

3. Recurrences, strong induction and substitution method (3.1, 3.2, 4.3).
4. Recursion tree method, Master method (4.4, 4.5).

Week 03

5. Introduction to random variables, probability distributions and expectations, hiring problem Appendices C.2, C.3, C.4, 5.4 (balls and bins).
6. Hiring assistant problem, Randomly permuting arrays (5.1, 5.2, 5.3)

Week 04

1. A Las Vegas algorithm: Randomized quicksort (7.3, 7.4)
2. A Monte Carlo algorithm: Randomized Mincut (Handout)

Week 05

1. Dynamic Programming I (15.1: Cut rod, 15.3 (reading exercise))
2. Dynamic Programming II (15.5: Optimal binary search trees / 15.2 Matrix chain multiplication)

Week 06

1. Dynamic Programming III (15.4: Longest common subsequence, 0/1 Knapsack)
2. Amortized Analysis (17.1, 17.2, 17.3)

Week 07

1. Application of amortized analysis: Fibonacci heaps (19.1,19.2)
2. Fibonacci heaps (19.3, 19.4)

Week 08

1. Ford Fulkerson (26.1, 26.2)
2. Ford Fulkerson (26.2 continued)

Midterm weekWeek 09

1. Application of Ford-Fulkerson on matching in bipartite graphs, a proof of Hall's theorem (26.3)
2. Introduction to Linear Programming (General to standard form), Linear Programming formulations (29.1, 29.2)



Week 10

3. Linear Programming duality
4. Complexity classes (34.1, 34.2, 34.3 motivation and definitions)

Week 11 (Ramadan week)

1. NP-completeness proofs (34.5 or handout)
2. NP-completeness proofs (Continued)

Week 12 (Ramadan week)

3. NP-completeness proofs (Continued)
4. Approximation algorithms I: Introduction, Vertex cover problem (35.1).

Week 13 (Ramadan week)

1. Approximation algorithms II (TSP 2-approximation) (35.2).
2. A randomized approximation algorithm (35.4).

Week 14 (Ramadan week)

1. Linear programming relaxations for approximation algorithms (35.4 continued)
2. Approximation algorithm for Set Cover (Handouts)

Week 15

1. Primal-dual method (Handout)
2. Primal-dual continued

Week 16

1. FPTAS for Knapsack problem (Handout).
2. Review

Grading Scheme/Criteria:

Assignments 25%

Midterms 35%

Final 40%

