CS-302 Design and Analysis of Algorithms

Spring 2018

National University of Computer and Emerging Sciences – FAST, Lahore

Credit Hours: 3

Prerequisites: Data Structures

Instructors: Ms. Saira Karim (saira.karim@nu.edu.pk)

Office Hours: Monday and Wednesday 11:00 a.m. – 1:00 p.m.

Textbook:

- *Introduction to Algorithms* by Cormen, Leiserson, Rivest, and Stein, 3rd Ed., MIT Press, 2001.

Reference Books:

- Jon Kleinberg, Éva Tardos, Algorithm Design, Pearson/Addison-Wesley
- Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, Algorithms, McGraw-Hill Education
- *Algorithms in C++* by Robert Sedgewick, Addison-Wesley, 1992.
- *Data Structures and Algorithms* by Aho, Hopcroft, and Ullman.

Objectives:

The objective of this course is not to fill your brains with every algorithm that you would ever need. One of the aims of this course is to teach you to reason about algorithms and describe them. In addition, many known algorithms to solve known problems will be taught. At the end of the course, you should be able to choose an appropriate algorithm from a set of algorithms for a given problem.

Grading Policy:

Grades will be awarded on the basis of continuous assessment through quizzes, assignments, two midterm exams and a final exam. The distribution of marks is as under:

Quizzes	15%
Assignments	15%
Mid term I	15%
Mid term II	15%
Final exam	40%

In writing up your assignments and in answering questions in exams, be as clear, precise, and concise as you can. **Understandability will be an important factor in grading**.

Academic Integrity: All work MUST be done individually. Any copying of work from other person(s) or source(s) (e.g. the Internet) will automatically result in at least an F grade in the course. It does not matter whether the copying is done in an assignment, quiz, midterm exam, or final exam, it will be considered equally significant.

Note: in order to pass the course student needs at least 50% marks.

Outline and Distribution:

Lectures	**Description**	**Chapters of Text**
Week -1	The role of algorithms in computers, Asymptotic functions and notations (Bid-oh, big-omega, theeta) best and worst case time complexity	1, 2, 3
Week – 2	Poof of correctness using invariant conditions Merge Sort + Solving recurrences	2, 3
Week – 3	Quicksort (average case analysis), Heapsort(build heap analysis)	6,7
Week – 4	Lower bound for comparison based sorting, Sorting in linear time: Count Sort, Radix Sort, Bucket Sort	8
Week – 5	B Trees (structure, worst case height, search and insertion algorithm)	18
	Midterm – I	
Week – 6,7	Greedy Algorithms (Activity selection, fractional knapsack and huffman codes) proof of correctness	16
Week – 8,9	Dynamic Programming (0/1 knap sack, longest common subsequence, longest increasing/decreasing subsequence)	15
Week – 10	Graph Algorithms (Representation, BFS, DFS,)	22
	Midterm – II	
Week – 11	Applications of BFS and DFS (Finding shortest path, checking bipartite graph, Topological sort, strongly connected components)	22
Week – 12	Minimum Spanning Trees (MST)(Prim's Algorithm and Kruskal's Algorithm)	23
Week – 13	Shortest Path Algorithms (dijkstra's Algorithm, BellmanFord and Warshall Algorithm	24
Week - 14	Amortized Analysis (Aggregate method, accounting method, potential method)	17
	Final Exam	Comprehensive