



# FAST School of Computing

## CS2009 – Design and Analysis of Algorithms

Spring 2025

**Instructor Name:** Muhammad Aasim Qureshi

**Email:** [aasim.qureshi@lhr.nu.edu.pk](mailto:aasim.qureshi@lhr.nu.edu.pk)

**Office Location:** New Building – Office Number **76**

**Office Hours:** **Wednesday/Friday** 11:00 A.M. to 01:00 PM

**TAs**

BSE 6B ---- Fahad Iqbal

BSE 6A ---- Husnain Ali

BSE 6C ---- Mahad

### Course Information:

**Program:** BS(SE)

**Credit Hours:** 3

**Type:** Core    **Pre-requisite:** Data Structures

**Class:** BSE 6A

**Venue:** CS-03

**Days:** Tuesday/Thursday @ 11:30 am

**Class:** BSE 6B

**Venue:** F-209

**Days:** Monday/Wednesday @ 01:00 pm

**Class:** BSE 6C

**Venue:** F-203

**Days:** Tuesday/Thursday @ 02:30 am

### Course Learning Objective:

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.

### Course Learning Outcomes (CLOs):

1. **Implement** the algorithms, compare the implementations empirically, and apply fundamental algorithms knowledge to solve practical problems related to the program. **(C3)**
2. **Analyze** the time and space complexity of different algorithms by using standard asymptotic notations for recursive and non-recursive algorithms. **(C4)**
3. **Evaluate** the correctness of algorithms by using theorem proving or executing test cases. **(C5)**
4. **Design** algorithms using different algorithms design techniques i.e., Brute Force, Divide and Conquer, Dynamic Programming, Greedy Algorithms and apply them to solve problems in the domain of the program. **(C6)**

### Course Textbooks:

Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, 4<sup>th</sup> Ed., 2022, MIT Press

### Additional references and books related to the course:

1. Algorithms, Sanjoy Dasgupta, Christos H. Papadimitriou, Umesh Vazirani, 2011, McGraw-Hill Education
2. Algorithm Design, Jon Kleinberg, Éva Tardos, 2013-2014, Pearson
3. Algorithms in C++, Parts 1-4 (Fundamentals, Data Structure, Sorting, Searching), Part 5 (Graph Algorithms), Sedgewick, Robert Wyk, Christopher J Van, 1998 - 2001, Addison-Wesley Professional
4. Data structures and Algorithms, Aho Hopcroft and Ullman, 1985, Addison-Wesley

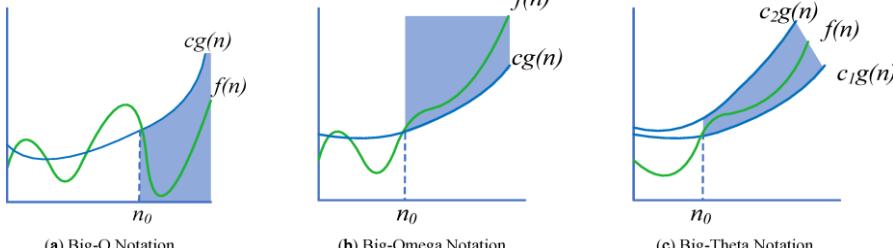
### (Tentative) Grading Criteria:

Assignments <b>(8-10%)</b>	Class Participation <b>(5-10%)</b>	Homework <b>(2-5%)</b>	Quizzes <b>(10-15%)</b>	Midterms <b>(30%)</b>	Final Exam <b>(40-45%)</b>
-------------------------------	---------------------------------------	---------------------------	----------------------------	--------------------------	-------------------------------

- Grading scheme for this course is **Absolute** under application of CS department's grading policies.
- Minimum requirement to pass this course is to obtain at least **50%** absolute marks.

## Course Policies:

- Quizzes may be announced or surprised. No Late Submissions or Makeup Quizzes.
- Students bear all the responsibility for protecting their assignments. In case of cheating, both parties will be considered equally responsible.
- **Plagiarism** in any work (Quizzes, Assignment, Midterms, and Final Exam) from any source, Internet or other Student(s)-class fellow or senior, will result in **F** grade or deduction of absolute marks.
- 80% attendance is required for appearing in the Final exams.

Week	Lecture Details	Book Chapters
1	<ul style="list-style-type: none"> <li>- The role of algorithms in computers,</li> <li>- Asymptotic functions and Notations, best and worst-case time complexity</li> </ul>  <p>(a) Big-O Notation      (b) Big-Omega Notation      (c) Big-Theta Notation</p>	1, 2, 3
2-3-4	<b>Divide and Conquer</b> <ul style="list-style-type: none"> <li>- Maximum subarray sum,</li> <li>- Counting inversions,</li> <li>- Merge sort,</li> <li>- Quicksort</li> <li>- <b>Solving recurrences</b></li> </ul>	2, 3, 6
5	<ul style="list-style-type: none"> <li>- Lower bound for comparison-based sorting,</li> <li>- Sorting in linear time: Count Sort</li> </ul>	8
6-7-8	<b>Dynamic Programming</b> <ul style="list-style-type: none"> <li>- Maximum subarray,</li> <li>- Rod cutting,</li> <li>- Longest common subsequence,</li> <li>- Binary knapsack</li> </ul>	15
9-10	<b>Greedy Algorithms</b> <ul style="list-style-type: none"> <li>- Activity selection,</li> <li>- Fractional knapsack</li> <li>- Huffman codes</li> </ul>	16
11-12	<b>Introduction to graphs</b> <ul style="list-style-type: none"> <li>- Revision of BFS, DFS Applications</li> <li>- (Cycle Detection, Shortest paths, Connected components)</li> </ul>	20
13	<b>Applications of DFS</b> <ul style="list-style-type: none"> <li>- Topological Sort</li> <li>- Strongly Connected Components</li> </ul>	20
14	<b>Minimum Spanning Trees</b> <ul style="list-style-type: none"> <li>- Prim's Algorithm</li> <li>- Kruskal's Algorithm</li> </ul>	21
15-16	<b>Shortest Path Algorithms</b> <ul style="list-style-type: none"> <li>- Single Source - Dijkstra's Algorithm, Bellman Ford Algorithm</li> <li>- All Pairs - Floyd Warshall Algorithm</li> </ul>	22-23