University of Asia Pacific (UAP)

Department of Computer Science and Engineering (CSE)

Course Outline

Program: Computer Science and Engineering (CSE)

Course Title: Algorithm

Course Code: CSE 207

Semester: Fall-2019

Level: **4**th Semester

Credit Hour: 3.0

Name & Designation of Teacher: Dr. M. Kaykobad, Professor, CSE, BUET
Tanjina Helaly, Assistant Professor

Office/Room: 7th Floor

Class Hours: Sunday3:30-5:00 pm & Monday 9:30 -11:00 am (Sec A)
Sunday 5:00 -6:30 pm & Thursday 2:00-3:30 pm (Sec B)

Consultation Hours: Wednesday 2:00 – 3:30 pm (Sec A)
Thursday 2:00 – 3:30 pm (Sec B)

e-mail: tanjina@uap-bd.edu

Mobile: +8801983792229

Rationale: Algorithm course is required for the student to be able to design, develop and analyze algorithm to solve real life problem. This is a required course and a pre-requisite to Theory of Computation (CSE 307), Mathematics for Computer Science (CSE 401), and Artificial Intelligence and Expert Systems (CSE 403) in the CSE program

Pre-requisite (if any)**:** CSE 103, CSE 205

Course Synopsis: **Introduction:** The role of algorithms in computing. **Complexity analysis**: Growth of function, asymptotic notations, orders, designing worst case and average-case. **Recurrence relations**: Substitution method, iteration method, master method. **Divide and Conquer**: Basic idea, control structure properties of D & C, applications of D & C. **Dynamic Programming**: Elements of Dynamic Programming, Comparison with D & C. Application of Dynamic programming in: Optimal binary search tree, 0/1 Knapsack problem. **Greedy Method**:

Elements of greedy method, basic control structure, Application of Greedy method in: Minimum cost spanning tree, Huffman code, Job sequencing with deadline. **Backtracking:** Basic idea behind backtracking, control structure. Application of backtracking in: graph coloring problem, n -queens problems. **Branch and Bound**: Basic idea and control structure of Branch and Bound. FIFO branch and Bound, LC Branch and Bound, the l5-puzzle problem. **Graph related algorithms**: Breadth First search, Depth First search, Topological sort, Dijkstra's shortest path algorithm, The Bellman-Ford algorithm for single source shortest path, The Floyd-War shall algorithm for all pair shortest path, Johnson's algorithm for sparse graph, Flow networks, the Ford-Fulkerson method. **Number theory algorithms**: Factorization problem, discrete logarithm problem, RSA, ElGamal, Diffie-Hellman. **String Matching**: Naïve string matching algorithm, the Rabin-Karp algorithm. **Computational Geometry**: Line segment properties, finding the convex hull. **NP-Completeness**: Polynomial time, polynomial time verification, NP-completeness and reducibility, NP- completeness proofs, NP complete problems. **Approximation Algorithms:** Introduction, the vertex-cover problem, the traveling-salesman problem, the subset-sum problem.

Course Objectives (CO):

The objectives of this course are to:

1. To **provide** a thorough understanding of a variety of algorithms with real-life applications and the resource requirements.
2. To **introduce** a number of important algorithm design techniques as well as basic algorithms that are interesting both from a theoretical and also practical point of view.
3. To l**earn** the design and implementation of algorithms, using languages like C, C++, Java, etc.
4. To **enable** students to analyze time and space complexities of algorithms.
5. To **emphasize** on efficient algorithm designing, solving practical problems through algorithmic techniques and data structures to be used in the implementations of algorithms.
6. To **expose** the students to a variety of techniques that have practical applications, while conducting detailed analysis of the requirements required by the algorithms.

Learning Outcomes (LO):

Upon completion of the course, the students will be able to:

1. **Describe** the objective of design and analysis of algorithms.
2. **Explain** terms related to important algorithm design techniques and basic algorithms.
3. **Understand** a practical problem, **apply** techniques and appropriate data structures to design and implement algorithms to solve the problem.
4. **Analyze** performance and resource requirements of various algorithms.

5. **Design and develop** algorithmic solutions to real-life problems.

Teaching-learning and Assessment Strategy: Lectures, assignments, quizzes, exams

Linkage of LO with Assessment Methods & their Weights:

LO	Assessment Method	(%)
1 – 3	Quiz	15
1 – 5	Class attendance	10
2 – 5	Assignment	5
1– 4	Midterm Exam	20
1 – 5	Final Exam	50

Minimum attendance: 70% class attendance is mandatory for a student in order to appear at the final examination.

Mapping of Course LO and Generic Skills:

Learning Outcome (LO) of the Course	Generic Skills* (Appendix-1)											
	1	2	3	4	5	6	7	8	9	10	11	12
Describe the objective of design and analysis of algorithms.	√			√								
Explain terms related to important algorithm design techniques and basic algorithms	√	√										
Understand a practical problem, **apply** techniques and appropriate data structures to design and implement algorithms to solve the problem	√	√	√	√	√							
Analyze performance and resource requirements of various algorithms		√		√								
Design and develop algorithmic solutions to real-life problems		√	√	√	√							

Lecture Schedule

Lecture #	Course Content	Assessment Strategy
1-3	Introduction to Algorithm. Analyzing Algorithms: Worst-Case and Best-Case	

	Analysis, Asymptotic notation.	
4-5	The Divide-and-Conquer Approach; Analyzing Divide-and-Conquer Algorithms (2.3.1, 2.3.2)	
6-8	The Maximum-Subarray Problem; Solving Recurrences (4.1, 4.4) Sorting algorithms Searching Algorithms	**CT1**
9-10	Direct-Address Tables, Hash Tables (11.1, 11.2) Hash Functions; Open Addressing (11.3, 11.4)	**Assignment1**
11-13	Greedy algorithm ; An Activity-Selection Problem; (16.1) Elements of the Greedy Strategy (16.2) Some Legacy Greedy Problems – Coin change, fractional knap sac.	CT2
14	review	
Mid Exam		
15-18	Dynamic Programming Basics, The Rod Cutting Problem (15.1) Longest Common Subsequence (15.4) Coin changing problem; Elements of Dynamic Programming (15.3) 0/1 Knapsack Problem,	**Assignment2**
19-20	Graph Theory. Applications of Graph Traversal Algorithms (DFS and BFS) (22.1, 22.2, 22.3,22.4)	CT3
21-23	Disjoint-Set Operations; Disjoint-Set Forests (21.1, 21.3) Growing a Minimum Spanning Tree (23.1) Kruskal's Algorithm (23.2) Single-Source Shortest Path Variants, Optimal Substructure of a Shortest Path, Negative-weight Edges, Cycles, Relaxation (24 (up-to 24.1 (exclusive))) The Bellman-Ford Algorithm (24.1) Dijkstra's Algorithm (24.3)	CT4
24	The Nave String-Matching Algorithm; The Rabin-Karp Algorithm(32.1, 32.2)	
25-27	Polynomial Time; Polynomial-Time Verification; NP-Completeness and Reducibility (34.1, 34.2, 34.3), NP-Hard, Reducibility	**Assignment3**
28	Review	
Final Exam		

Textbook: T.H. Cormen, C.E.Leiserson, R. L. Rivest, C. Stein: *Introduction to Algorithms*, Third Edition, 2009, PHI Learning Pvt. Ltd

Recommended References:

1. Doanld E. Knuth : *Fundamental Algorithms (vol-1) (The Art of Computer Programming)*, Third edition, 1997, Addison-Wesley Professional
2. Ellis Horowitz & Sartaj Sahni*: Fundamental of Computer Algorithms*, First Edition, 1983, Springer

Grading System: As per the approved grading scale of University of Asia Pacific (Appendix-2).

Student's responsibilities: Students must come to the class prepared for the course material covered in the previous class (es).
They must submit their assignments on time.
There will be no make-up quizzes.

Appendix-1: **Generic Skills**

No.	Generic Skills
1.	Engineering Knowledge
2.	Problem Analysis
3.	Design/Development of Solutions
4.	Investigation
5.	Modern Tool Usage
6.	The Engineer and Society
7.	Environment and Sustainability
8.	Ethics
9.	Communication
10.	Individual and Team Work
11.	Life Long Learning
12.	Project Management and Finance

Generic Skills (Detailed):

1. **Engineering Knowledge (T)** -Apply knowledge of mathematics, sciences, engineering fundamentals and manufacturing engineering to the solution of complex engineering problems;
2. **Problem Analysis (T)** – Identify, formulate, research relevant literature and analyze complex engineering problems, and reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences;
3. **Design/Development of Solutions (A)** –Design solutions, exhibiting innovativeness, for complex engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, economical, ethical, environmental and sustainability issues.
4. **Investigation (D)** Conduct investigation into complex problems, displaying creativeness, using research-based knowledge, and research methods including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions;
5. **Modern Tool Usage (A & D)** -Create, select and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modelling, to complex engineering activities, with an understanding of the limitations;
6. **The Engineer and Society (ESSE)** -Apply reasoning based on contextual knowledge to assess societal, health, safety, legal, cultural, contemporary issues, and the consequent responsibilities relevant to professional engineering practices.
7. **Environment and Sustainability (ESSE)** -Understand the impact of professional engineering solutions in societal, global, and environmental contexts and demonstrate knowledge of and need for sustainable development;
8. **Ethics (ESSE)** –Apply professional ethics with Islamic values and commit to responsibilities and norms of professional engineering code of practices.

9. **Communication (S) -**Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions;
10. **Individual and Team Work (S) -**Function effectively as an individual, and as a member or leader in diverse teams and in multi-disciplinary settings.
11. **Life Long Learning (S) -**Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
12. **Project Management and Finance (S) -**Demonstrate knowledge and understanding of engineering management and financial principles and apply these to one's own work, as a member and/or leader in a team, to manage projects in multidisciplinary settings, and identify opportunities of entrepreneurship.

Appendix-2: Grading Policy

Numeric Grade	Letter Grade	Grade Point
80% and above	A+	4.00
75% to less than 80%	A	3.75
70% to less than 75%	A-	3.50
65% to less than 70%	B+	3.25
60% to less than 65%	B	3.00
55% to less than 60%	B-	2.75
50% to less than 55%	C+	2.50
45% to less than 50%	C	2.25
40% to less than 45%	D	2.00
Less than 40%	F	0.00

Prepared by: **Checked by:** **Approved by:**
(Head of the Detp.)

--------------------------------- ----------------------------- -----------------------------