CS3301: APPLIED ALGORITHMS

Effective Term
Semester A 2022/23

Part I Course Overview

Course Title
Applied Algorithms

Subject Code
CS - Computer Science
Course Number
3301

Academic Unit
Computer Science (CS)

College/School
College of Engineering (EG)

Course Duration
One Semester

Credit Units
3

Level
B1, B2, B3, B4 - Bachelor's Degree

Medium of Instruction
English

Medium of Assessment
English

Prerequisites
Nil

Precursors
CS2303 Data Structures for Media or equivalent

Equivalent Courses
Nil

Exclusive Courses
Nil

Part II Course Details

Abstract

This course aims to introduce the algorithms in various domains, and techniques for designing efficient algorithms. It trains students the ability to analyse algorithms and the skills to design solutions to problems especially for media related problems..

Course Intended Learning Outcomes (CILOs)

	CILOs	Weighting (if app.)	DEC-A1	DEC-A2	DEC-A3
1	Prove the correctness and analyse the running time and performance of the major algorithms for those classic problems in various domains.		x	x	
2	Apply algorithmic paradigms and methods by using design techniques to solve problems.		x	x	
3	Investigate the complexities of various problems in different domains.			x	
4	Propose new solutions for problems through independent study.			x	x

A1: Attitude
Develop an attitude of discovery/innovation/creativity, as demonstrated by students possessing a strong sense of curiosity, asking questions actively, challenging assumptions or engaging in inquiry together with teachers.

A2: Ability
Develop the ability/skill needed to discover/innovate/create, as demonstrated by students possessing critical thinking skills to assess ideas, acquiring research skills, synthesizing knowledge across disciplines or applying academic knowledge to real-life problems.

A3: Accomplishments
Demonstrate accomplishment of discovery/innovation/creativity through producing /constructing creative works/new artefacts, effective solutions to real-life problems or new processes.

Teaching and Learning Activities (TLAs)

	TLAs	Brief Description	CILO No.	Hours/week (if applicable)
1	Lecture	General algorithmic design and analysis techniques will be taught in the lecture.	1, 2, 3, 4	3 hours/week
2	Tutorial	Exercises will be given in the tutorial and the lecturer (with the participation of students) will eventually give the answers.	1, 2, 3, 4	8 hours/semester
3	Assignment	Assignments contain problems that students should try to solve by adopting the best solutions.	1, 2, 3, 4	After class

4	Project	A programming project (for example a game project) needs to be completed by using algorithms learned in the course.	2, 4	After class

Assessment Tasks / Activities (ATs)

	ATs	CILO No.	Weighting (%)	Remarks (e.g. Parameter for GenAI use)
1	Assignments	1, 2, 3, 4	15	
2	Project	2, 4	15	

Continuous Assessment (%)
30

Examination (%)
70

Examination Duration (Hours)
2

Additional Information for ATs
For a student to pass the course, at least 30% of the maximum mark for the examination must be obtained.

Assessment Rubrics (AR)

Assessment Task
Assignment

Criterion
1.1Each question is given a score

Excellent (A+, A, A-)
75% or above

Good (B+, B, B-)
60% or above

Fair (C+, C, C-)
45% or above

Marginal (D)
40% or above

Failure (F)
Below 40%

Assessment Task
Project

Criterion
2.1 The project is evaluated by the innovation, design and use of algorithms

Excellent (A+, A, A-)
75% or above

Good (B+, B, B-)
60% or above

Fair (C+, C, C-)
45% or above

Marginal (D)
40% or above

Failure (F)
Below 40%

Assessment Task
Examination

Criterion
3.1Each question is given a score

Excellent (A+, A, A-)
75% or above

Good (B+, B, B-)
60% or above

Fair (C+, C, C-)
45% or above

Marginal (D)
40% or above

Failure (F)
Below 40%

Part III Other Information

Keyword Syllabus

Algorithm analysis. Algorithm design: divide-and-conquer approach, greedy approach. Graph algorithms: graph searching, topological sort, minimum spanning tree, shortest paths, backtracking and its applications in games, A* algorithm. String matching. Dynamic programming and longest common subsequence. Introduction to computational complexity.

Syllabus

- Algorithm analysis
  Introduction to computational complexity, Review on program correctness and complexities, and the mathematical tools for analysis.
- Graph algorithms
  Representation of graphs. Algorithms for graph searching. Topological sort. Minimum spanning trees. Greedy design approach. Shortest paths, transitive closure and their relations with matrix multiplication. Backtracking and applications in games. A* algorithm.

- String algorithms
  String matching. Longest common subsequence. Dynamic programming.

Reading List

Compulsory Readings

	Title
1	J. Kleinberg and E. Tardos (2013). Algorithm design. PEARSON EDUCATION. paperback

Additional Readings

	Title
1	T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein (2009). Introduction to Algorithms. MIT Press, 3rd edition.
2	Daniel, Sanchez-Crespo Dalmau (2003). Core Techniques and Algorithms in Game Programming